Master’s Thesis

Constructing Assurance Case using Description and Discussion Information from an Issue Tracking System

Khana Chindamaikul

February 5, 2015

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology
A Master’s Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

Khana Chindamaikul

Thesis Committee:
Professor Hajimu Iida (Supervisor)
Professor Keiichi Yasumoto (Supervisor)
Associate Professor Kohei Ichikawa (Supervisor)
Assistant Professor Toshinori Takai (Supervisor)

Associate Professor Arnon Rungsawang (Supervisor)
Constructing Assurance Case using Description and Discussion Information from an Issue Tracking System*

Khana Chindamaikul

Abstract

We address the problem of constructing assurance cases such as high cost and time of constructing and maintaining assurance cases. We have presented several approaches for retrieving relevant documents and constructing assurance cases from existing documents, artifacts, or products in software development processes. Those proposed approaches allow us to reduce the cost of constructing assurance cases by reducing the time needed to select the relevant documents and construct assurance cases. We perform an experiment with a data set from an open-source software development project, in order to evaluate the effectiveness of our approaches. The experimental results suggest that the proposed approaches can be effective in terms of reducing the time and the cost for constructing assurance cases with acceptable confidence level, indicated by some assurance case quality metrics which are proposed by our research work.

Keywords:


Acknowledgements

With utmost sincerity, I would like to express my deep gratitude to the following people for their guidance and support, without whose help this work would never have been successful:

At first, I would like to express my very great appreciation to my supervisor, Assistant Professor Toshinori Takai for his valuable supports, constructive suggestions and enthusiastic encouragement during the planning and development of this research work. Without his guidance and supports, I would not successfully accomplish the master course. His willingness to give his time so generously has been very much appreciated.

I would also like to extend my thanks to Professor Hajimu Iida and Associate Professor Kohei Ichikawa for their generous support of the research environment and constructive comments. Without them, my research works would be not possible.

I would like to offer my special thanks to my thesis committees for reviewing my thesis and for those helpful comments and suggestions that guided me to improve the quality of my thesis.

Thanks to Associate Professor Arnon Rungsawang and Associate Professor Daniel Port for their useful critiques and support of this research work. Those comments are really helpful to improve my research work.

Special thanks to Professor Kenichi Matsumoto and Dr. Pattara Leelaprute for the great opportunity that they gave to me for an internship at the Nara Institute of Science and Technology. I gained invaluable experiences during my internship and these experiences inspire me to study abroad.
Thanks to generous support of Mobile Communication Fund Docomo Scholarship for Privately Financed International Students. It is a very substantial and great support. This scholarship can cover all of my daily expense and relieve me from worrying so much about the cost of tuition fee.

Special thanks to JSPS KAKENHI Grant No.25730044 for research expense. This grant can afford me to continue my research work and gave me the great opportunity to go to international conference.

I wish to thank all members of the Software Design and Analysis Laboratory and my Thai friends for their guidance of living in Japan and their friendship. Without them, my life in the university would be more difficult and more troublesome.

Finally and most importantly, I would like to express my deep gratitude to my family. Their unconditional love and support make me have the wonderful day. They are the source of my strength and sustenance for doing master thesis.
Contents

Abstract i

Acknowledgements ii

1 Introduction 1
  1.1. Thesis Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
  1.2. Organization of Thesis . . . . . . . . . . . . . . . . . . . . . . . 5

2 Assurance Case 6
  2.1. Roles of Assurance Case . . . . . . . . . . . . . . . . . . . . . . . 6
  2.2. Goal Structuring Notation . . . . . . . . . . . . . . . . . . . . . . 7
  2.3. Problem of Assurance Case . . . . . . . . . . . . . . . . . . . . . 9
  2.4. The Existing Approach for Constructing Assurance Case . . . . . 9
  2.5. The Existing Approach for Measuring Quality of Assurance Case . 10

3 Background Techniques 11
  3.1. Document Retrieval (DR) . . . . . . . . . . . . . . . . . . . . . . . 11
  3.2. Topic Modeling (TM) . . . . . . . . . . . . . . . . . . . . . . . . . 12
  3.3. Formal Concept Analysis (FCA) . . . . . . . . . . . . . . . . . . . 13
  3.4. Design Rationale . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
  3.5. Safety Contract . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

4 Proposed Approaches 18
  4.1. Using DR for Constructing Assurance Cases . . . . . . . . . . . . 18
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>Using TM for Constructing Assurance Cases</td>
<td>19</td>
</tr>
<tr>
<td>4.3</td>
<td>Using DR and TM for Constructing Assurance Cases</td>
<td>20</td>
</tr>
<tr>
<td>4.4</td>
<td>Using DR and FCA for Constructing Assurance Cases</td>
<td>21</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Selecting properties</td>
<td>23</td>
</tr>
<tr>
<td>4.5</td>
<td>Using Design Rationale for Constructing Assurance Cases</td>
<td>24</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Extension of the Safety Contract</td>
<td>24</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Conceptual Mapping of the Design Rationale to Safety Contracts</td>
<td>25</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Construct Assurance Cases from Assurance-Contracts</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>Proposed Assurance Case Indicator</td>
<td>27</td>
</tr>
<tr>
<td>5.1</td>
<td>Claim Coverage</td>
<td>27</td>
</tr>
<tr>
<td>5.2</td>
<td>Argument Coverage</td>
<td>28</td>
</tr>
<tr>
<td>5.3</td>
<td>Defect Density</td>
<td>31</td>
</tr>
<tr>
<td>5.4</td>
<td>Combination of Assurance Cases Indicators</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>Experiments</td>
<td>32</td>
</tr>
<tr>
<td>6.1</td>
<td>Purpose</td>
<td>32</td>
</tr>
<tr>
<td>6.2</td>
<td>Studied Target</td>
<td>33</td>
</tr>
<tr>
<td>6.3</td>
<td>Evaluation Criteria and Measures</td>
<td>33</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Accuracy Measures</td>
<td>33</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Quality of Assurance Cases</td>
<td>35</td>
</tr>
<tr>
<td>6.4</td>
<td>Constructing Assurance Cases</td>
<td>36</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Performance Comparison of Retrieved Information of each Approach</td>
<td>36</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Quality Comparison of Assurance Cases from each Approach</td>
<td>39</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Construction of Assurance Cases from Design Rationale Document</td>
<td>45</td>
</tr>
<tr>
<td>7</td>
<td>Conclusion and Future Work</td>
<td>57</td>
</tr>
<tr>
<td>8</td>
<td>Contribution</td>
<td>59</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Overview of our research work ........................................ 4
2.1 A structure of an Assurance Case (modified from [1]) .......... 7
2.2 Basic elements of Goal Structuring Notation (GSN) .......... 7
2.3 An assurance case expressed in GSN notation ................. 8
3.1 An Overview of Document Retrieval .............................. 11
3.2 An example of a Concept Lattice ................................. 14
3.3 The structure of Design Rationale (modified from [2]) ........ 15
3.4 Types of Line in Design Rationale Document .................. 15
3.5 An example of Design Rationale ................................. 16
4.1 Using DR for Constructing Assurance Cases .................... 19
4.2 Using TM for Constructing Assurance Cases .................... 19
4.3 Using DR and TM for Constructing Assurance Cases ........... 20
4.4 Using DR and FCA for Constructing Assurance Cases ......... 21
4.5 The context types of Goal Structuring Notation ............... 26
5.1 Comparison between ACs ........................................... 28
5.2 Master Assurance Case $A_M$ for $A_1$ and $A_2$ ............... 29
6.1 Searching documents in concept lattice ......................... 35
6.2 A precision and recall of each approach ....................... 41
6.3 Parts of an obtained assurance case ............................. 43
6.4 The classification of issues in Issue Tracker System ........... 46
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5</td>
<td>The example results of design rationale documents (a)</td>
<td>49</td>
</tr>
<tr>
<td>6.6</td>
<td>The example results of design rationale documents (b)</td>
<td>50</td>
</tr>
<tr>
<td>6.7</td>
<td>The example results of design rationale documents (c)</td>
<td>51</td>
</tr>
<tr>
<td>6.8</td>
<td>A part of assurance case for authentication</td>
<td>54</td>
</tr>
<tr>
<td>6.9</td>
<td>A part of assurance case for the moodle forum</td>
<td>55</td>
</tr>
</tbody>
</table>
## List of Tables

1.1 The approach procedure and its details ........................................ 4

3.1 A set of top $n$ words for each topic ....................................... 12
3.2 A set of top $k$ topics for each document ................................. 12
3.3 An Example of a Formal Context ............................................ 13
3.4 An example of safety contract for the metal cutter system .......... 17

4.1 An extension of safety contract for the metal cutter system .......... 24
4.2 Conceptual mapping between design rationale and assurance-contracts 25
4.3 Conceptual mapping between safety contracts and GSN elements .... 26

6.1 Performance of Retrieved Information of each Approach .............. 38
6.2 Accuracy of each approach .................................................... 41
6.3 Browsing cost of each relevant documents ................................. 42
6.4 Size of the obtained assurance cases ....................................... 44
6.5 Claim coverage of each an assurance case .................................. 44
6.6 Argument coverage of each an assurance case .............................. 44
6.7 The example of bug report in moodle tracker system ............... 47
6.8 The example of discussion issue in moodle tracker system .......... 47
6.9 An example results of safety contracts (a) for the authentication ... 52
6.10 An example results of safety contracts (b) for the Moodle forum .... 52
6.11 An example results of safety contracts (c) for the language interface 53
Chapter 1

Introduction

Nowadays, many systems tend to be huge. It is difficult to achieve complete safety or security for such large systems, including large software systems. A traditional software testing and evaluation approaches cannot achieve the necessary level of justified confidence, due to factors such as the size, complexity, and continuing evolution of the product, along with unexpected events and other external influences [3], [4]. Even increased strictness of criteria or more testing will not sufficiently increase the confidence or prevent accidents.

Instead, an assurance case is expected to provide a required level of justified confidence for the systems. An assurance case is defined as

“A documented body of evidence that provides a convincing and valid argument that a specified set of critical claims about a system’s properties are adequately justified for a given application in a given environment” [5].

In industry, this document is mainly used in safety critical systems, i.e. its failure could result in loss of life, significant property damage, or damage to the environment [6]. For example, some safety related standards in automotive, aviation, railway and nuclear power plants system domains require an assurance case [7], [8]. An assurance case for the safety property is called a safety case.
An assurance case can often be a large size of documents, and thus the cost for constructing and maintaining go far beyond a reasonable level. In a survey paper of system safety and system assurance [9], the future research directions of this area include:

“We also have to recognise that safety cases are costly to develop, so we should seek more efficient means of construction”.

An assurance case is usually constructed manually. This is not only extremely time-consuming, but also cannot avoid defects of an assurance case itself, i.e. on statements of claims, strategies for arguments and evaluation of evidence.

One of major proportions of the costs of developing assurance cases is to prepare information. Since assurance cases should be supported by appropriate evidence, a lot of coordinating effort from huge information sources is needed, e.g. design analysis, risk assessment, tests and verification efforts, or requirements from stakeholders. Generally, assurance cases are constructed by assurance engineers, who are expected to be independent from developers of the system or development engineers. This implies that the tasks for understanding existing documents and artifacts will take a lot of time and efforts for them.

Our work aims to enable third parties to construct an assurance case like independent verification and validation (IV&V) activities. More precisely, the goal is to reduce the time needed to understand and search the relevant documents to cut the cost of constructing an assurance case. We have presented five approaches (Described in Section 4) of retrieving relevant documents from existing documents, artifacts, or products in software development processes. In the proposed approaches, we apply the combinations of several techniques, e.g. Document Retrieval (DR) [30], Topic Modeling (TM) [31], and Formal Concept Analysis (FCA) [35].

We also proposed the new approach for constructing assurance cases from design rationale documents, which are created from the relevant documents. The approach allows the assurance people to speed up the creation of assurance cases and the obtained assurance case is expected to capture important information.
that is likely to be unnoticeable. Since quantitative metrics for evaluating the quality of assurance cases have not been established yet, we proposed a new metrics for assurance cases, i.e. claim coverage, argument coverage and defect density, for measuring the quality of assurance cases. For applying the proposed indicators, an assurance case under evaluation is assumed to be represented in Goal Structuring Notation (GSN). GSN is one of the major graphical representations of assurance cases. We call the proposed metrics assurance case indicators and use them in the experiment part of this research work.

We perform an experiment in order to evaluate the effectiveness of our approaches. As the target of the experiment, we focus on issues in issue tracking systems, which can be used as materials for constructing an assurance case. An issue tracking system [10], [11] is a software application that keeps tracking, managing and maintaining the reported software issues in software development projects. An issue tracking system often also contains knowledge base information, the solution for a problem and so on. The reason we choose this data source is that it can be regarded as huge information sources which gather a set of artifacts in a software development projects, and it is easier to access and obtain data than other real data source in industry. We compare the effectiveness of the proposed five approaches mentioned above, and the experimental results suggest that the approach which combines Document Retrieval and Formal Concept Analysis technique provides the best accuracy for constructing assurance cases. Furthermore, a concept lattice of this approach is shown as being effective for searching the relevant information. More precisely, it provides the descriptive labels offering users additional guidance for exploring necessary information; they can then shorten the search path by only exploring relevant nodes and ignoring the other ones, thus reducing the searching time. Additionally, the results also suggest that constructing assurance cases by using this approach achieved the same quality level as manual approach with less time and efforts.
1.1. Thesis Overview

This thesis proposed the novel approach for retrieving relevant documents and constructing assurance cases. Figure 1.1 shows the overview of our whole research work. In order to construct assurance cases from issues in issue tracking system, there are four steps for the translation as shown in Table 1.1.

Table 1.1: The approach procedure and its details

<table>
<thead>
<tr>
<th>Trans.</th>
<th>Description</th>
<th>Degree of automation</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>Retrieving relevant documents</td>
<td>Automatic</td>
<td>4.1-4.4</td>
</tr>
<tr>
<td>(B)</td>
<td>Constructing Design Rationale</td>
<td>Manual</td>
<td>3.4</td>
</tr>
<tr>
<td>(C)</td>
<td>Constructing Assurance-Contracts</td>
<td>Semi-auto</td>
<td>4.5.2</td>
</tr>
<tr>
<td>(D)</td>
<td>Constructing Assurance Cases</td>
<td>Semi-auto</td>
<td>4.5.3</td>
</tr>
<tr>
<td>(E)</td>
<td>Measuring quality of Assurance</td>
<td>Quantitative</td>
<td>5</td>
</tr>
</tbody>
</table>

Note that (E) is a quantitative evaluation method of assurance cases quality.
1.2. Organization of Thesis

The remaining thesis is organized as follows. Chapter 2 provides the knowledge of assurance cases, including roles of assurance cases, graphical representation of assurance cases, the problems of assurance cases and the existing approaches for dealing with the problem. Chapter 3 presents a background of the techniques we used in our research work. Chapter 4 describes the overview and the details of our proposed approaches for constructing assurance cases. Chapter 5 shows the proposed assurance case indicator in order to measure the quality of assurance cases. Chapter 6 presents the experiments. Chapter 7 draws the conclusion and discussion about the future work. Finally, Chapter 8 presents the contribution of our research work.
Chapter 2

Assurance Case

2.1. Roles of Assurance Case

An Assurance Case (AC) is defined as “a documented body of evidence that provides a convincing and valid argument that a specified set of critical claims about a system’s properties are adequately justified for a given application in a given environment” [5]. Assurance cases are mainly used for critical systems such as automotive, aviation, railway and nuclear power plants systems. Some safety and security related standards in industry require assurance cases [7], [8].

An assurance case consists of 3 kinds of information: a claim, an argument and evidence. A claim is a proposition about an attribute or a property of the system. Typically, a claim is a risk-related requirement of the system to be assured, e.g. safety, security, or dependability. Evidence is data supporting a claim, which is either facts, assumptions or other assurance cases. An argument is a description showing how the evidence supports or justifies the claim, which can be deterministic, probabilistic or qualitative. Figure 2.1 illustrates an image of an assurance case.
2.2. Goal Structuring Notation

*Goal Structuring Notation (GSN)* [1] is one of the most widely used graphical representation of the structured argument for assurance cases. This representation is designed to enable assurance cases in term of easy understanding and can be manipulated by machines.

GSN is a tree whose node types are shown in Figure 2.2.

- *Goals* are claims of assurance cases which are written in rectangles.
- *Strategies* are arguments of assurance cases which are written in parallelograms.
• *Solutions* are evidences of connected goal which are written in ovals.

• *Contexts* are definitions of terms appeared in the goals or strategies or references represent the domain/scope, which are written in a box with rounded corners.

• *Undeveloped elements* indicates a state that the goal to which this symbol is attached requires further development. Undeveloped is written in diamond box.

The root node is a goal. A goal is decomposed through a strategy. Every leaf is either evidence, undeveloped for supporting a goal, or a context. A context is attached to a goal or strategy to describe an additional information. An example of assurance cases in GSN is shown in Figure 2.3.

![Figure 2.3: An assurance case expressed in GSN notation](image_url)
2.3. Problem of Assurance Case

The cost for achieving certification of safety critical computer systems is estimated at 25-75% of the development costs [12]. As a part of certification, constructing and maintaining an assurance case for the safety-critical system is time-consuming and its cost go far beyond reasonable level. Moreover, currently there is no satisfying approach exists to support automatic generation. In the next subsection, the existing approaches are discussed.

We consider the difficulty of the construction of assurance cases by reducing to the following three problems.

1. There is no systematic approach to enable third parties (cf. IV&V) who do not relate to a system, to construct, check or maintain assurance cases easily.

2. There is no effective way to find the related information from various and huge sources of information in a software development projects for constructing an assurance case. Note that only a small fraction of artifacts is relevant or necessary to the system assurance.

3. There is no explicit way to measure the quality of an assurance case.

In our research work, we solved these three important problems by proposing the approach for assisting the construction of assurance cases (Described in Section 4), and presenting the assurance cases indicator for measuring the quality of assurance cases (Described in Section 5).

2.4. The Existing Approach for Constructing Assurance Case

There are some guidelines for constructing assurance cases, especially safety cases, in industry(e.g. [13–15]). The early systematic approach is to use patterns of assurance cases [16]. Some researchers proposed translation methods from existing
products, e.g. design document and risk analysis results, to assurance cases [17,18] but those methods requires some specific structures as source documents. Strunk and Knight [19] used problem frames to create and structure the implementation goals and contexts of assurance cases written in a goal structuring notation. Sljivo [20] proposed the approach to generate safety case argument-fragments from safety contracts. But, the safety contracts are not prepared in all the system/software development projects. In this thesis, we give an basic idea to extend the method by giving translation from the information in issue tracking systems to safety contracts. D. Jackson et al. [21] showed a concrete illustration of an approach to constructing a dependability case for the control software of a medical device. Denney et al. [22] presented a tool that uses model based transformation and extended GSN to support the automated construction of safety cases. We proposed some approaches for retrieving information from a document repository for constructing assurance cases [23–25].

2.5. The Existing Approach for Measuring Quality of Assurance Case

There are some proposals to give methods for evaluating confidence of assurance cases. Kelly [26] proposed several aspects to check for review activities of assurance cases. Bloomfield and Littlewood [27] proposed a method to measure confidence based on conditional probability to evaluate so-called “multi-legged” arguments as a way to increase confidence from single leg cases. Denney et al. [22] proposed a quantitative measure mainly focusing on the coverage of arguments. Denny et al. [28] also proposed another method to evaluate confidence of assurance cases based on Bayesian Networks. Goodenough et al. [29] introduced a notation, called a confidence map, which is expected to be used with GSN to show the status of the current confidence. We proposed some metrics of assurance cases (we call them assurance cases indicators) for evaluating and comparing the quality of assurance cases [25].
Chapter 3

Background Techniques

In this section we present background information on the techniques which are used in our research work.

3.1. Document Retrieval (DR)

*Document Retrieval* is a technique to elicit documents related to a given query from a large set of documents [30]. Each document is treated as an unstructured text. A query can be a sentence or a set of several words. Figure 3.1 shows the overview of Document Retrieval technique, where the input of Document Retrieval is a query and a large set of documents and the output of Document Retrieval is a relating documents with ranks to the given query. The rank represents the degree of relatedness for each document. In the following, we call the result a *ranked results*.

![Figure 3.1: An Overview of Document Retrieval](image-url)
3.2. Topic Modeling (TM)

*Topic Modeling (TM)* is a technique for automatically extracting semantic topics from a collection of text documents [31]. A *topic* in topic modeling is represented as a list of words that occur in statistically meaningful ways or frequently occur together. The underlying idea is based on the assumption that each document can be represented by a small number of topics, where each topic is assumed to be dominated by a small fraction of all possible words.

Some algorithms for Topic Modeling such as Latent Dirichlet Allocation (LDA) [32], Pachinko Allocation Model (PAM) [33] have been proposed. We use MALLET [34] as a tool for topic modeling, which is based on LDA. Table 3.1 shows a set of words for each topic, e.g. topic number 13 relates to safety and security.

Table 3.1: A set of top $n$ words for each topic

<table>
<thead>
<tr>
<th>Topic No.</th>
<th>A set of words for each topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>modules, current, feature, imo, part</td>
</tr>
<tr>
<td>12</td>
<td>http, moodle, org, check, sourceforge</td>
</tr>
<tr>
<td>13</td>
<td>login, authen, password, security, secure</td>
</tr>
<tr>
<td>14</td>
<td>user, interface, typo, bug, function</td>
</tr>
</tbody>
</table>

Table 3.2 shows a set of topics for each document, e.g. document number 69 relates to topic 35, 13 and 21 (we shows only top 3 topics). Note that a proportion of each topic shows how much topic relates to a document.

Table 3.2: A set of top $k$ topics for each document

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>BugID8069</td>
<td>35</td>
<td>8.15%</td>
<td>13</td>
<td>5.01%</td>
<td>21</td>
<td>4.81%</td>
</tr>
<tr>
<td>70</td>
<td>BugID8070</td>
<td>86</td>
<td>18.33%</td>
<td>31</td>
<td>11.02%</td>
<td>17</td>
<td>9.21%</td>
</tr>
<tr>
<td>71</td>
<td>BugID8071</td>
<td>30</td>
<td>6.23%</td>
<td>13</td>
<td>4.51%</td>
<td>61</td>
<td>2.37%</td>
</tr>
<tr>
<td>72</td>
<td>BugID8072</td>
<td>43</td>
<td>16.23%</td>
<td>42</td>
<td>8.15%</td>
<td>12</td>
<td>3.71%</td>
</tr>
</tbody>
</table>
3.3. Formal Concept Analysis (FCA)

*Formal Concept Analysis* is a way to analyse relations between concepts [35]. An input of FCA is a *formal context* and an output is a *concept lattice*. A formal context consists of a set of *objects*, a set of *properties*, and a relation between objects and properties.

Table 3.3 shows an example of a formal context; the first row is a set of properties; the first column is a set of objects; ‘X’ represents a relationship between objects and properties. A concept lattice is a lattice whose nodes are the set of *concepts* where a concept is a maximal collection of objects that share common properties. Edges in a lattice represent subset relation with respect to objects (or equivalently properties but in the opposite direction).

<table>
<thead>
<tr>
<th></th>
<th>duplic</th>
<th>student</th>
<th>ldap</th>
<th>server</th>
<th>manual</th>
<th>error</th>
<th>secure</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDL-8031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDL-8065</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MDL-8081</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDL-8086</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDL-8153</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDL-8442</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.2 is an example of a concept lattice; the circle shapes are the concept nodes, the gray rectangles represent the properties and the white rectangles represent the objects.

3.4. Design Rationale

*Design Rationale* is a documentation of the reasons behind a decision for a given decision problem. This document includes its justification, other considered alternatives, and argumentation leading to the decision [2]. The additional infor-
Figure 3.2: An example of a Concept Lattice

Information in Design Rationale provides insight into the system and also gives the richer view of decision making.

Since, activities of system assurance can boil down to giving confident decision making [37]. Therefore, information in Design Rationale would be valuable for constructing assurance cases. For the larger the system, the more decisions and changes will be made during its lifetime. Hence, it is difficult to capture all the information, including unexpected changes of the system which affects the conclusion or structure of arguments in the assurance cases. Then, capturing the information in Design Rationale would be some assistance in term of pointing the important information for constructing assurance cases and could help to make change management of assurance cases easier.

Figure 3.3 shows the structure of Design Rationale, where a decision is a statement representing a decision problem to be solved, alternatives are candidates of solutions for the decision or choices considered to satisfy the decision, arguments are reasons for supporting or against the alternatives, and premises are evidence that used to support each argument. Note that there are 2 types of argument. The first one is pro-argument which is used for supporting and marked with the symbol ‘+’. The second one is con-argument which is used for against
the alternatives and marked with the symbol ‘-’.

Figure 3.3: The structure of Design Rationale (modified from [2])

There are six types of line for connecting to premises show in Figure 3.4. The green line represents a link for a normal premise; the red line with red spot represents a link for a negated premise; the red dashed line represents a link for an exception premise; the green dashed line with green spot represents a link for a negated exception premise; the green dotted line represents a link for an assumption premise; the red dotted line with red spot represents a link for a negated assumption premise.

Figure 3.4: Types of Line in Design Rationale Document
The example of decision which is recorded by Design Rationale is shown in the Figure 3.5. The decision is “How to send username and initial password to users” and it has two alternatives to solve the problem posed in the decision. The first one is “Sending both username and initial password in the single email”. However, there are not only advantages but also disadvantages for this choice such as the security problems. The second one is “Sending username and initial password via different emails”. This choice also has some advantages and disadvantages which are expressed by pro and con-arguments.

Figure 3.5: An example of Design Rationale
3.5. Safety Contract

Meyer [38] introduced the notion of contracts in the context of programming language in 1992. Recently, the idea is applied to safety-critical system developments [20], [39], [40]. A safety contract is a set of information about a component of a system. More precisely, a safety contract [20] is a triple \((A, G, E)\), where \(A\) is an assumption, \(G\) is a guarantee that is provided by the component if assumptions are met, and \(E\) is evidence showing the backing of the implication. Intuitively saying, if the assumption satisfies then the guarantee will also hold with the evidence.

Table 3.4: An example of safety contract for the metal cutter system

<table>
<thead>
<tr>
<th>Assumption</th>
<th>The detection sensor module of metal cutter detects a human or animal is approaching within ([0, 3) meters].</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guarantee</td>
<td>The motor module of metal cutter will stop within (0.3) seconds.</td>
</tr>
<tr>
<td>Evidence</td>
<td>software architecture design specification</td>
</tr>
</tbody>
</table>

Table 3.4 shows an example of safety contract for the metal cutter system; this contract guarantees that the motor of metal cutter will stop within \(0.3\) seconds if the detection sensor detects the object within 3 meters.
Chapter 4

Proposed Approaches

In this section, we present the detail of our proposed approaches to find issues which can be used for constructing assurance cases, from an issue tracking system. The first 4 approaches are mainly constructing assurance cases which provide confidence for the system to against those well-known problems. The fifth proposed approach, we expect that the constructed assurance cases could mention about unnoticeable or subtle problems that are important for system assurance.

4.1. Using DR for Constructing Assurance Cases

We utilize Document Retrieval for retrieving the related documents from a document repository. We feel that this technique could be of some assistance in term of reducing the time and effort for searching relevant information and constructing assurance cases. Figure 4.1 shows a flow of this approach. The procedure has 4 steps as follows:

(a) **Formulating a query**: a user selects words or sentences that relate to the claim of an assurance case.

(b) **Searching documents**: a set of documents that relates to a query is retrieved by document retrieval technique, we call it the *ranked results*. 
(c) **Screening results**: the user selects the relevant documents from the ranked results.

(d) Constructing an assurance case from the relevant documents and information obtained in the previous steps.

### 4.2. Using TM for Constructing Assurance Cases

We utilize Topic Modeling to analyze large volumes of unlabeled documents. Topic Modeling provides the set of topics and the set of words for each document. For this reason, we expect that this technique could help user to find the relevant information by reading the topic of each document. Figure 4.2 shows a flow of this approach. The procedure has 3 steps as follows:

![Diagram](image-url)
(a) **Inferring documents**: the set of documents are inferred using a topic modeling tool. Then, documents that are related to the same topic are grouped into the same group. A TM tool can also provide a set of words for each topic. So, user can use some of these words for naming the group. The group that has a name similar to a query is called *group of interest*.

(b) **Screening results**: user selects documents from a small size of documents in *group of interest*. Since each group is characterized by a finite set of words given by TM, it is expected that user can select relevant documents without careful reading inside of the given documents.

(c) Constructing an assurance case from the relevant documents and information obtained in the previous steps.

### 4.3. Using DR and TM for Constructing Assurance Cases

We combine document retrieval technique with topic modeling technique to retrieve relevant documents for constructing assurance case. The flow of this approach is shown in Figure 4.3.

![Figure 4.3: Using DR and TM for Constructing Assurance Cases](image)

The procedure has 5 steps as follows:
(a) **Formulating a query**: a user selects words or sentences that relate to the claim of an assurance case.

(b) **Searching documents**: a set of documents that relates to a query is retrieved by document retrieval technique. The output of this step is called ranked results.

(c) **Inferring documents**: a set of documents in ranked results are inferred using a topic modeling tool. The detail of this step is described in section 4.2.

(d) **Screening results**: user selects relevant documents from a small size of documents in *group of interest*.

(e) Constructing an assurance case from the relevant documents and information obtained in the previous steps.

### 4.4. Using DR and FCA for Constructing Assurance Cases

We combine *document retrieval* and *formal concept analysis* technique to retrieve relevant documents for constructing assurance case. The flow of this approach is shown in Figure 4.4.

![Figure 4.4: Using DR and FCA for Constructing Assurance Cases](image-url)

Figure 4.4: Using DR and FCA for Constructing Assurance Cases
(a) **Formulating a query**: a user selects words or sentences that relate to the claim for an assurance case.

(b) **Searching documents**: a set of documents that relates to a query is retrieved by document retrieval technique (*ranked results*).

(c) **Selecting properties**: the *tf-idf weighting* of words from the top *n* documents in the *ranked results* are computed. This weighting is intended to reflect how important a word is to each document in a collection. All the words from the top *n* documents are ranked by their *tf-idf weighting* and the top *k* words are selected as properties of a formal context.

(d) **Formulating a formal context**: a *formal context* is generated from a set of top *n* documents (objects) in the ranked results and a set of top *k* words (properties) from the step (c).

(e) **Applying formal concept analysis**: the *formal context* from the previous step is used as an input. Then FCA builds a set of concepts (*which is a concept lattice*). A *concept lattice* is very helpful for users in reducing their search effort by providing a label for each concept node.

(f) **Screening results**: The user selects the relevant documents from the resulting concept lattice by reading the annotated description of each concept node and by seeing the relation among the concept nodes.

(g) Constructing an assurance case from the relevant documents and information obtained in the previous steps.
4.4.1 Selecting properties

In the following, we present how the \textit{tf-idf} technique is adopted and used to select terms (properties) to be used in FCA. The idea is based on the study of Myat [41].

In this technique, a document is regarded as a set of unordered terms (words). We denote a set of documents in the ranked results as $D = \{d_1, d_2 \ldots d_n\}$. A set of distinct terms which occur in $D$ is denoted as $T_D = \{t_1, t_2 \ldots t_m\}$.

\textit{Term Frequency (tf)} is defined as $tf_{ij} = f_{ij}/\max\{f_{ij}\}$, where the $f_{ij}$ is the frequency of term $t_i$ in document $d_j$, $\max\{f_{ij}\}$ is the maximum frequency computed over all terms in $d_j$ $(1 \leq i \leq m, 1 \leq j \leq n)$. Note that, Term Frequency indicates that the more frequent terms in a document are more important.

\textit{Inverse Document Frequency (idf)} is defined as $idf_i = \log_2 (N/df_i)$, where $N$ is a total number of documents $D$ and the $df_i$ is a number of documents containing term $t_i$ $(1 \leq i \leq m)$. Note that, Inverse Document Frequency indicates that terms which appear in many different documents are less indicative of whole topic.

In order to rank every term $t_i \in T_D$, we apply the following equation to determine the ranking of terms with $(1 \leq i \leq m$ and $1 \leq j \leq n)$:

\[
\text{tf-idf weighting } \left( w_{ij} \right) = tf_{ij} \cdot idf_i = \frac{f_{ij}}{\max\{f_{ij}\}} \log_2 \left( \frac{N}{df_i} \right).
\]

Using this equation we can rank all the unique terms in $D$. So the terms occurring frequently in the document but rarely in the rest of the collection is given high weight (high rank).

In this paper, we only select top 25 terms as the properties and select top 100 documents as objects of a formal context, where the numbers of terms and documents are based on the study by Poshyvanyk et al [42].
4.5. Using Design Rationale for Constructing Assurance Cases

We utilize Design Rationale to capture information which is necessary for constructing assurance cases and propose a procedure to construct assurance cases from Design Rationale. There are two steps for construction; the first step is to translate the design rationale document into safety contracts, we proposed the conceptual mapping of the design rationale document to safety contract; the second step is to construct assurance cases from safety contracts, we use the approach for generating assurance cases which is proposed by Sljivo [20].

4.5.1 Extension of the Safety Contract

We also proposed the extension of the traditional safety contract with the definition and disagreement elements, called Assurance-Contract. This Assurance-Contract allows for capturing the additional information which is important for system assurance case, such as the context in which the assumption and the guarantee holds, or the disagreement and exception of the system or components.

The Assurance-Contract \((\mathcal{D}_F, \mathcal{A}, \mathcal{G}, \mathcal{D}_S)\) is composed of assumptions \(\mathcal{A}\) and guarantees \(\mathcal{G}\), where the system or component holds the guarantees \(\mathcal{G}\) if its assumptions \(\mathcal{A}\) on its definition and context \(\mathcal{D}_F\) are satisfied along with its disagreement or exception \(\mathcal{D}_S\). Table 4.1 shows an example of the extended safety contract for the metal cutter system.

<table>
<thead>
<tr>
<th>Definition</th>
<th>SW version 4.5, a height of detected object &gt; 20cm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumption</td>
<td>The detection sensor module of metal cutter detects a human or animal is approaching within [0, 3 meters].</td>
</tr>
<tr>
<td>Guarantee</td>
<td>The motor of metal cutter will stop within 0.3 sec.</td>
</tr>
<tr>
<td>Disagreement</td>
<td>The cost of accurate sensor is very high</td>
</tr>
<tr>
<td>Evidence</td>
<td>Software architecture design specification</td>
</tr>
</tbody>
</table>

Table 4.1: An extension of safety contract for the metal cutter system
4.5.2 Conceptual Mapping of the Design Rationale to Safety Contracts

In order to build safety contracts from the design rationale document, it is necessary for mapping the design rationale elements to safety contracts elements. The conceptual mapping between the design rationale and safety contracts is depicted in Table 4.3. Note that we use the extension of safety contracts for this translation, and the elements of design rationale are described in Section 3.4.

Table 4.2: Conceptual mapping between design rationale and assurance-contracts

<table>
<thead>
<tr>
<th>Design Rationale Elements</th>
<th>Assurance-Contracts Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>- All types of premise</td>
<td>Definition</td>
</tr>
<tr>
<td>- Decision and its alternative</td>
<td></td>
</tr>
<tr>
<td>- Assumption premise</td>
<td>Assumption</td>
</tr>
<tr>
<td>- Negated assumption premise</td>
<td></td>
</tr>
<tr>
<td>- Normal premise</td>
<td>Guarantee</td>
</tr>
<tr>
<td>- Negated premise</td>
<td></td>
</tr>
<tr>
<td>- Exception premise</td>
<td>Disagreement</td>
</tr>
<tr>
<td>- Negated exception premise</td>
<td></td>
</tr>
</tbody>
</table>

4.5.3 Construct Assurance Cases from Assurance-Contracts

We construct an assurance case from a set of assurance-contracts by extending Sljive’s approach [20]. This is based on semi-automatic generation of argument-fragments from assumption/guarantee safety contracts by using rules and patterns for automatic generation.

From the view of an assurance case which is expressed in GSN (as mentioned in Section 2.2), a context is attached to a goal or strategy to describe an additional information or an environment. We consider a context of GSN that can be classified into the following 3 type (see Figure 4.5):
• Limitation Context ($C_L$): to describe the exception or the limitation of the attached claim of an assurance case.

• Definition Context ($C_D$): to describe the definition or environment of the attached claim of an assurance case.

• Strategy Context ($C_S$): to describe a backing of the attached strategy.

Figure 4.5: The context types of Goal Structuring Notation

At the same time, as described in Section 4.5.1. Our extension of the Safety Contract captures an assumption, a guarantee, a disagreement and evidence. Following Table 4.3 we present how we use the conceptual mapping between our extension of safety contracts and GSN elements for the translation.

Table 4.3: Conceptual mapping between safety contracts and GSN elements

<table>
<thead>
<tr>
<th>Assurance-Contracts Elements</th>
<th>GSN elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Guarantee</td>
<td>Goals</td>
</tr>
<tr>
<td>- Assumed safety requirement</td>
<td></td>
</tr>
<tr>
<td>- Evidence</td>
<td>Solutions</td>
</tr>
<tr>
<td>- Assumption</td>
<td>$C_S$</td>
</tr>
<tr>
<td>- Definition</td>
<td>$C_D$</td>
</tr>
<tr>
<td>- Disagreement</td>
<td>$C_L$</td>
</tr>
</tbody>
</table>
Chapter 5

Proposed Assurance Case Indicator

This chapter gives the proposed measurements of quality of assurance cases. First, we give three ways to measure confidence of assurance cases. The proposed metric consists of a combination of them.

5.1. Claim Coverage

This metric is used for evaluating the coverage of assurance cases claims. For this metric, we assume a reference-issues-list. For example, if the claim is related to the safety, then a reference-issues-list can be “the reference hazard list” containing a list of hazards which are already known and agreed by stakeholders. Using the notion of a reference-issues-list, the claims coverage is defined as follow:

\[
\text{Claim coverage} = \frac{\# \text{ issues which are covered by ACs}}{\# \text{ issues in reference-issues-list}}
\]

The claim coverage represents how much assurance cases is mentioning about a reference-issues-list. Note that the 100% of this metric does not mean that the system completely satisfy the top goal of assurance cases.
5.2. Argument Coverage

This metric is used for evaluating the coverage of arguments and evidence whereas the previous one can be regarded as the coverage of claims. For example, if the claims Prob. $a$ and Prob. $b$ in Figure 5.1 are elements of the given reference issue list and the others are not, then the values of both GSNs with respect to the claim coverage measure are the same.

In fact, the right-hand side GSN in Figure 5.1 has richer arguments than the left-hand side, and thus the evaluator of this GSN could have more confidence.

In this research, we propose the relative measurement to evaluate results of the experiment. The relative measurement is an approach to compare assurance cases with the master assurance case.

Since in the experiment, the assurance case construction part has been executed by a single expert group, it can be assumed that there is no conflict among the obtained assurance cases. That leads us to construct a “master” assurance case which contains all the information of assurance cases from all approaches. In other words, each approach’s assurance case can be obtained by trimming some
branches from the master assurance case. For example, if we assume two assurance cases $A_1$ and $A_2$ in the Figure 5.2 are obtained, then we can construct the master assurance case of them as $A_M$.

Moreover, it is natural to expect that a goal in a higher position (i.e. nearer to the root) has more significant role than a lower ones, we adopt the following measurement. For example, assurance case $A_1$ in the Figure 5.2 lacks the sub-tree of $G_6$ compared to assurance case $A_2$ whereas assurance case $A_2$ lacks the sub-trees of $G_4$ and $G_5$ compared to assurance case $A_1$. In this case, it can be expected that the lack of the sub-tree of $G_6$ has a larger impact than the lack of the sub-tree of $G_4$ and $G_5$.

Let $A$ be an assurance case given in a GSN, which is under evaluation. $A_M$ be the master assurance case, which is written in a GSN. According to the definition of master assurance case, there exists the injective function $\iota$ from $\text{goal}(A)$ to $\text{goal}(A_M)$ where $\text{goal}(\cdot)$ represents the set of all the goals appeared in $A$, mapping a goal $g$ to $\iota(g)$ where $\iota(g)$ has the same statement of $g$. Before defining the argument coverage for assurance cases, we give the argument coverage for goals. Let $[\cdot]$ be a function taking a goal of the assurance case to be evaluated and $\iota(g)$ also has the same position in a tree of $g$. 

---

\[1\] $\iota(g)$ also has the same position in a tree of $g$. 

---

29
returns a real number from 0 to 1 defined as follows: let \( g \) be a goal in \( \mathcal{A} \).

1. if \( \iota(g) \) is undeveloped, then \( \|g\| = 1 \),

2. if both \( \iota(g) \) and \( g \) are directly supported by evidence, then \( \|g\| = 1 \),

3. if \( \iota(g) \) is directly supported by evidence but \( g \) is not, i.e. \( g \) is undeveloped, then \( \|g\| = 0 \),

4. if \( \iota(g) \) is supported by sub-goals \( g_{M,1}, \ldots, g_{M,n} \) with \( n \geq 1 \), then let \( e_i \) for \( 1 \leq i \leq n \) be a real number defined as follows:
   - if there exists \( g_i \in \text{goal}(\mathcal{A}) \) such that \( f(g_i) = g_{M,i} \), then \( e_i = \|g_i\| \), and
   - otherwise, \( e_i = 0 \).

Then the value of \( g \) is defined as \( \|g\| = \sum_{1 \leq i \leq n} e_i / n \).

The argument coverage \( \mathcal{A} \) for assurance cases is defined as \( \|[A]\| = \|\text{root}(A)\| \) where \text{root} represent the root node.

For example, in assurance case \( \mathcal{A}_1 \) of Figure 5.2, \( G_2, G_3, G_4, \) and \( G_5 \) are all evaluated as 1 and then \( G_1 \) is 0.5 since it lacks the argument about \( G_6 \) in compared with assurance case \( \mathcal{A}_M \). So, the argument coverage of assurance case \( \mathcal{A}_1 \) is 0.5. On the other hand, we can calculate the argument coverage of assurance case \( \mathcal{A}_2 \) as 0.66 because it lacks the argument about \( G_4 \) and \( G_5 \) in compared with assurance case \( \mathcal{A}_M \). Note, if we compare only a number of the goals of assurance case \( \mathcal{A}_1 \) and assurance case \( \mathcal{A}_2 \), they are the same.
5.3. Defect Density

We assert that quality of assurance cases is decreased as a number of assurance cases defects are presented. The defect density is used to measure how many defects occur in assurance cases. Kelly et al. [26] proposed the review aspects for assurance cases.

We should take into account that increasing size and complexity of assurance cases will also raise defects. A defect density is defined as a number of defects divided by a size of assurance cases which is indicated by a number of elements (e.g. claim, evidence, argument), shown in equation below.

\[
\text{Defect density} = \frac{\text{#Defects in AC}}{\text{#Elements in AC}}
\]

5.4. Combination of Assurance Cases Indicators

We propose the new indicator for evaluating the quality of assurance cases as triple

\[(c, a, d),\]

where \(c\) is claim coverage, \(a\) is an argument coverage, and \(d\) is a defect density.
Chapter 6

Experiments

6.1. Purpose

We have conducted three studies in our experiment. The purposes of those studies are as follows:

- The first study (performance comparison of retrieved information): to compare the accuracy of retrieving information of each approach and suggest the approach that can provide useful means for constructing assurance cases.

- The second study (quality comparison of assurance cases): to compare the quality of assurance cases from each approach and suggest the most effective approach in term of the cost of construction and the quality of assurance cases.

- The third study (construction of assurance cases from design rationale document): to illustrate the construction of assurance cases from design rationale documents. We expect that the obtained assurance cases can capture important information that is likely to be unnoticeable.
6.2. Studied Target

The target system is a course management system for educational institution, which is called Moodle. Moodle [43] is a learning system designed to provide educators, administrators and learners with a single robust, secure and integrated system to create personalised learning environments. Moodle is built by the Moodle project which is coordinated by an Australian company of 30 developers which is financially supported by a network of 60 Moodle Partner service companies worldwide.

Moodle Tracker System [44] is an issue tracker system which records and manages all issues related to Moodle and related systems. The target issues are issue numbers from 8,000 to 8,500. The reason we choose this range is that it has sufficient number of security issues according to our preliminary analysis.

6.3. Evaluation Criteria and Measures

6.3.1 Accuracy Measures

We compare the accuracy of each approach by measuring the cost of reading and the cost of browsing, which is proposed by J. M. Cigarrn [45].

Cost of Reading [45]

This measures is used for evaluating the reading cost while finding all relevant documents. In order to measure this cost we use the precision and the recall method.

\[
\text{Precision} = \frac{|\text{relevant docs} \cap \text{retrieved docs}|}{|\text{retrieved docs}|}
\]

\[
\text{Recall} = \frac{|\text{relevant docs} \cap \text{retrieved docs}|}{|\text{total relevant docs}|}
\]
Cost of Browsing [45]

This measures is used for evaluating the browsing cost of each relevant document. We assume that for the manual approach, a user has to scan each document until the relevant document is found (i.e. linear searching). Therefore, the search visits only \(N/2\) documents [46] on average, the browsing cost is only 50%. Note that \(N\) is the total number of documents.

We also impose the same worst case for our proposed approach, i.e. assume that a user has to start searching at the top concept node, and while visiting a concept in the concept lattice a user has to read all the documents relating to this concept. In reality, users may start searching at the node whose label is relating to the user’s interest and may not read all documents in the concept.

We use the lattice browsing complexity (\(LBC\)) [45] in order to measure this cost. \(LBC\) is defined as follow. Note that \(C_{\text{view}}\) is a concept which is visited by a user.

\[
LBC(C) = \frac{|C_{\text{view}}|}{|C|} \times 100\%
\]

Consider the example from Figure 6.1. We assume the worst case for this example which is user wants to find the document number MDL-8031. This document is related to the security and it is in the deepest position level in the graph. So, we start searching at the top concept node and then go through the concept nodes that their label relate to security words.

For this example, \(C_{\text{view}} = 4\), which is the minimal number of concept a user has to visit. \(C = 10\), which is the total number of concept node. Therefore, \(LBC(C) = (4/10) \times 100\% = 40\%\), which means that a user need to search at most 40% of the total concepts in this graph.
6.3.2 Quality of Assurance Cases

We compare assurance case results from each approach by our assurance case indicator which is proposed in Section 5. The proposed indicator consist of a triple \((c, a, d)\) where \(c\) is claim coverage, \(a\) is an argument coverage, and \(d\) is a defect density. However, in this experiment, the assurance cases construction part has been executed by only single expert group. It implies that the defect density of each assurance case is supposed to has the same value.
6.4. Constructing Assurance Cases

6.4.1 Performance Comparison of Retrieved Information of each Approach

In this section, we show the preliminary study to compare the accuracy of the retrieved information of each approach. We first describe the design of study. Then we give some results of this study. Finally, we explain the summary that would lead to the next study.

Studied Design

We present four different way of retrieving information for constructing assurance cases. The first approach consists in reading all issues and constructing an assurance case. The second approach is using document retrieval to prepare information (Section 4.1). The third approach is using topic modeling to prepare information (Section 4.2). The fourth approach is using both document retrieval and topic modeling (Section 4.3). The details of those approaches are as follows:

(1) Approach A: we read all of the issues and found that there were only 14 issues that related to safety or security (relevant issues). Those 14 issues will be used for constructing an assurance case. We also used the knowledge from this approach as base knowledge of this study.

(2) Approach B (using DR): we used “safety security” as a query. The issues that relate to this query are retrieved by the document retrieval engine. The output of document retrieval is called ranked results. We read all issues in ranked results and found relevant issues for constructing an assurance case.

(3) Approach C (using TM): we used the set of selected issues as an input of Mallet. The number of topics we used as inputs of Mallet is in range from 76 to 115. One of the observations that we had while trying to use less than 76 topics is that words “safety security” did not appear in a set of words for each topic. While when we tried to use more than 115 topics, words “safety security”
security” appeared more than once in a set of words for topics. For example, word “safety” appears in a set of words for topic 1 and it also appears in a set of words for topic 2. From 76 to 115 topics, there was only 1 topic which has words “safety security” in a set of words. To simplify the evaluation we considered the case when words “safety security” appeared only once. We call the topic which has the words “safety” and “security” in a set of words the \textit{safety topic}. After that, we grouped the issues which have top 2 topics related to the \textit{safety topic} into the same group. This group was called the \textit{safety group}. Then, we analyzed the \textit{safety group} by measuring the precision and recall while we changed the number of topics from 76 to 115. We selected 76 topics as the best input topic number of Mallet for this approach because it had the best precision and recall. Finally, we constructed an assurance case from those relevant issues in safety group.

(4) Approach D (using DR and TM): we used “safety security” as a query. Then, we utilized the document retrieval technique for searching issues in issue tracking system. A list of issues that relate to the query is retrieved, we called the ranked results. After that, we used all issues in the ranked results as input of MALLET. We used the number of topics ranging from 5 to 26 as an input of topic modeling. The reason why we choose this range is same as the reason of the third approach. We selected 5 topics as the best input topic number of mallet for this approach. Finally, we constructed an assurance case from those relevant issues in safety group.

**Studied Results**

We compare the approaches by measuring an effort, a time, coverage and a cost. The results are summarized in Table 6.1. We assume as the worst case that user read all issues in the \textit{safety group} for the third approach and the fourth approach. Note that, the average time for reading each issue is about 4.40 minutes. An effort is measured as the number of issues that a user has to read during constructing assurance cases. Coverage is measured by recall.
Table 6.1: Performance of Retrieved Information of each Approach

<table>
<thead>
<tr>
<th>Approach</th>
<th>Effort</th>
<th>Time (min.)</th>
<th>Coverage</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach A</td>
<td>501</td>
<td>2204.4</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Approach B (DR)</td>
<td>42</td>
<td>184.8</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Approach C (TM)</td>
<td>14</td>
<td>61.6</td>
<td>Low</td>
<td>Very Low</td>
</tr>
<tr>
<td>Approach D (DR+TM)</td>
<td>9</td>
<td>39.6</td>
<td>Medium</td>
<td>Very Low</td>
</tr>
</tbody>
</table>

- Approach A: user takes a lot of time for understanding whole documents.
- Approach B (DR): user still takes a lot of time for understanding the documents if the number of retrieved documents are big.
- Approach C (TM): there are only few documents that can be information for constructing assurance cases.
- Approach D (DR+TM): a size of documents for considering is small and coverage is acceptable. So it is possible to reduce the effort, time and cost for constructing assurance cases.

Summary

Although the study presented here is a preliminary phase, the result suggests that the proposed approach which combines document retrieval and topic modeling technique can provide useful means for constructing assurance cases. The result also shows that the approach can be effective because it is possible to reduce an effort, a time and a cost of construction. However, some relevant issues may miss by using this approach. So, trade-off between time and coverage of assurance cases is required for consideration.
6.4.2 Quality Comparison of Assurance Cases from each Approach

In our previous study, we presented four approaches (A) read all documents, (B) using DR, (C) using TM, (D) using DR and TM to find informative issues in issue tracking system for constructing assurance cases. We conducted preliminary experiment to compare those four approaches by measuring only accuracy of the retrieved information. It means the previous study did not measure the quality of assurance cases directly. The results of this previous study suggested that the (D) approach can be the most effective in term of providing useful means for constructing assurance cases.

In this study, we start from (D) as the base approach and propose a new approach for constructing assurance cases. Moreover, we propose a metrics to evaluate the quality of an assurance case directly. We first describe the design and environments of this study. Then we discuss the evaluation criteria and show the experimental results. Finally, we draw the summary.

Setting Environments

Issues in Moodle tracker system from issue number 8,000 to 8,500 are used as information for constructing assurance cases.

We compare 3 approaches for preparing information to construct assurance cases. The first one is a manual approach, the second approach is the best one from the previous study, and the third approach is the new proposed approach. The details of these approaches are as follows:

1. **Approach A (Manual)**: Reading all documents and determining the relevant documents that relate to security properties.

2. **Approach D (DR and TM)**: Combining the document retrieval with the topic modeling for retrieving relevant documents. The details of this approach is shown in Section 4.3.
(3) **Approach E (DR and FCA)**: Combining the document retrieval with the FCA for retrieving relevant documents. The details of this approach is shown in Section 4.4.

Note that assurance cases of each approach are constructed from their relevant documents by the same expert group.

**Evaluation Criteria**

In this study, we compare the accuracy of each approach by measuring the cost of reading and the cost of browsing, which is explained in Section 6.3.1. We also compare the quality of assurance case from each approach by our assurance case indicator which is proposed in Section 5.

**Experimental Results**

We chose to retrieve security-related documents. Three set of queries are formulated. We used the same set of queries for the Approach D and the Approach E. Note that the total number of relevant documents is calculated by a number of the union of relevant documents from all approaches and from the expert group.

**Cost of reading**: We compute the precision and recall of each approach for each query. The results in Table 6.2 and Figure 6.2 show that the approach E provides better precision and recall than the approach D for all queries (better accuracy). Note that the quality of the result can be affected by the quality of a query, e.g. the quality of the query3 is lower than the query1, thus the precision and recall of the query3 are lower than others.

The results also show that the approach E brings the improvement of searching over the approach A (improvement of precision). This means a user can reduce the overall time for searching. However, some relevant documents may miss by using the approach E. Therefore, trade-off analysis between the time and the coverage is required for consideration. Also remark that the approach A misses some relevant documents too. According to the reviewing of this experiment, this is caused by a decrease of concentration.
### Table 6.2: Accuracy of each approach

<table>
<thead>
<tr>
<th>Approach</th>
<th>Query</th>
<th>#Retrieval</th>
<th>#Relevant</th>
<th>Total relevant</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach A: Read all documents</td>
<td>-</td>
<td>500</td>
<td>11</td>
<td>14</td>
<td>2.20%</td>
<td>78.57%</td>
</tr>
<tr>
<td>Approach D: DR and Topic Modeling</td>
<td>Query1</td>
<td>15</td>
<td>6</td>
<td>14</td>
<td>40.00%</td>
<td>42.86%</td>
</tr>
<tr>
<td></td>
<td>Query2</td>
<td>16</td>
<td>8</td>
<td>14</td>
<td>50.00%</td>
<td>57.14%</td>
</tr>
<tr>
<td></td>
<td>Query3</td>
<td>17</td>
<td>2</td>
<td>14</td>
<td>11.76%</td>
<td>14.29%</td>
</tr>
<tr>
<td>Approach E: DR and FCA</td>
<td>Query1</td>
<td>11</td>
<td>9</td>
<td>14</td>
<td>81.82%</td>
<td>64.29%</td>
</tr>
<tr>
<td></td>
<td>Query2</td>
<td>11</td>
<td>9</td>
<td>14</td>
<td>81.82%</td>
<td>64.29%</td>
</tr>
<tr>
<td></td>
<td>Query3</td>
<td>12</td>
<td>4</td>
<td>14</td>
<td>33.33%</td>
<td>28.57%</td>
</tr>
</tbody>
</table>

Figure 6.2: A precision and recall of each approach
**Cost of browsing:** the browsing cost of each relevant documents is shown in Table 6.3. We compute the LBC measures for concept lattices of approach E and compare those with the cost of browsing of other approaches.

<table>
<thead>
<tr>
<th>Document ID</th>
<th>C</th>
<th>$C_{\text{view}}$</th>
<th>Approach A (Linear)</th>
<th>Approach D (Linear)</th>
<th>Approach E (LBC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDL-8031</td>
<td>28</td>
<td>2</td>
<td>50%</td>
<td>50%</td>
<td>7.14%</td>
</tr>
<tr>
<td>MDL-8070</td>
<td>28</td>
<td>3</td>
<td>50%</td>
<td>50%</td>
<td>10.71%</td>
</tr>
<tr>
<td>MDL-8097</td>
<td>28</td>
<td>3</td>
<td>50%</td>
<td>50%</td>
<td>10.71%</td>
</tr>
<tr>
<td>MDL-8131</td>
<td>28</td>
<td>1</td>
<td>50%</td>
<td>50%</td>
<td>3.57%</td>
</tr>
<tr>
<td>MDL-8148</td>
<td>28</td>
<td>2</td>
<td>50%</td>
<td>50%</td>
<td>7.14%</td>
</tr>
</tbody>
</table>

We searched for all 14 relevant documents in the obtained concept lattices. We present only the example top 5 relevant documents, in order by the document ID number.

The results show that the browsing cost of approach E is lower than other approaches for all relevant documents. This indicates that concept lattices of approach E are easier to browse than the list of documents of approach A and D.
Obtained Assurance Cases:

Since the obtained assurance cases are too large for presenting here, some parts are shown in Figure 6.3. The top goal of this assurance case shows that system is acceptably secure. This goal is decomposed through a strategy which argues by three major sub-properties of security especially for Moodle system, as Figure 6.3(a). An example of a leaf part is shown in Figure 6.3(b).
Table 6.4: Size of the obtained assurance cases

<table>
<thead>
<tr>
<th></th>
<th>#Goal</th>
<th>#Strategy</th>
<th>#Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC of approach A</td>
<td>21</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>AC of approach D</td>
<td>22</td>
<td>13</td>
<td>6</td>
</tr>
<tr>
<td>AC of approach E</td>
<td>24</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>Master AC</td>
<td>26</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

The size of assurance cases are shown in Table 6.4. Next, we measure quality of assurance cases by the assurance cases indicator.

**Claim coverage**: for this measurement, we use a list of “common types of security vulnerability”, provided by Moodle system (see moodle documents\(^1\)) as a reference-issues-list. The result of the claim coverage is shown in Table 6.5.

Table 6.5: Claim coverage of each an assurance case

<table>
<thead>
<tr>
<th></th>
<th>Approach A</th>
<th>Approach D</th>
<th>Approach E</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Vulnerability which AC cover</td>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>#Reference vulnerability</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Claim coverage</td>
<td>0.60</td>
<td>0.53</td>
<td>0.60</td>
</tr>
</tbody>
</table>

According to Table 6.5, the claim coverage of assurance cases from each approach are not so different. This indicates that quality of the assurance case from approach E is comparable to approach A and D. Moreover, the approach E takes less time than the approach A and D.

**Argument coverage**: the results of the argument coverage is shown in Table 6.6.

Table 6.6: Argument coverage of each an assurance case

<table>
<thead>
<tr>
<th></th>
<th>Approach A</th>
<th>Approach D</th>
<th>Approach E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing elements</td>
<td>G11,G25,G30</td>
<td>G3,G11</td>
<td>G11</td>
</tr>
<tr>
<td>Argument coverage</td>
<td>0.801</td>
<td>0.819</td>
<td>0.986</td>
</tr>
</tbody>
</table>

\(^1\)http://docs.moodle.org/dev/Security#Common_types_of_security_vulnerability
According to Table 6.6, the argument coverage of the assurance case from the approach E is higher than other approaches. This result indicates that the quality of the assurance case from approach E is comparable to other approaches.

The experimental results shown in this section demonstrate that the new proposed approach E provides convenient way to prepare information for constructing assurance cases with accurate information. Furthermore, the results indicate that quality of the assurance case from the proposed approach E is acceptable in comparison with assurance cases obtained by hand.

Summary

In this study, we proposed a new approach which combines document retrieval and formal concept analysis to prepare information for constructing assurance cases. We performed an experiment to evaluate the proposed approach. The results show that the proposed approach provides acceptable accuracy for constructing assurance cases. Furthermore, a concept lattice of this approach is shown as being effective for searching the relevant information. More precisely, it provides the descriptive labels offering users additional guidance for exploring necessary information. The experimental results indicate that an assurance case obtained by the new proposed approach can have as the same quality level as the manual approach.

6.4.3 Construction of Assurance Cases from Design Rationale Document

In this section, we conducted the case study to show the construction of assurance cases from design rationale document. We first describe the studied target and studied design. Then we show the result of the construction. Finally, we draw the summary that would lead to the future work.
Studied Target

From the previous studies, i.e. Section 6.4.1, 6.4.2, we found that the issues in moodle tracker system that can be classified into the following 2 types, i.e. a bug report and a discussion issue, as shown in Figure 6.4.

- Bug report: the structure of this issue type is simple. It can be retrieved easily by document retrieval techniques.

- Discussion issue: this issue type contains important information for system assurance, but its conclusion and its underlying reason are generally not clear. Thus, we need additional techniques to capture the reason behind it.

Figure 6.4: The classification of issues in Issue Tracker System

Note that the first study (Section 6.4.1) and the second study (Section 6.4.2) are mainly working with bug report; the constructed assurance cases of those two studies are only about well-known problems. In this section, we try to deal with discussion issues, we expect that the constructed assurance cases can mention about subtle problems that are important for system assurance.
Table 6.7: The example of bug report in moodle tracker system

<table>
<thead>
<tr>
<th>Bug No.</th>
<th>MDL-8054</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>The activity “Main Glossary” allows students to modify the terms added to the glossary.</td>
</tr>
<tr>
<td>Description</td>
<td>By creating the activity ”Glossary” when the option ”Main Glossary” is selected, only teachers should be allowed to insert terms and definitions. However, this option is ignored, allowing the students to insert and modify.</td>
</tr>
<tr>
<td>Comment</td>
<td>There are no comments yet on this issue.</td>
</tr>
</tbody>
</table>

The example of bug report in moodle tracker system is shown in Table 6.7. There is no discussion in comments and its conclusion is clear. The issue is fixed in version 1.9.

Table 6.8: The example of discussion issue in moodle tracker system

<table>
<thead>
<tr>
<th>Bug No.</th>
<th>MDL-8036</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Moodle Forums - Why not Focus on Course Content and Use PHPBB, VBulletin, etc.</td>
</tr>
<tr>
<td>Description</td>
<td>Why not focus on Course Content delivery and just build integration to more extensive forums like PHPBB or VBulletin, or others?</td>
</tr>
</tbody>
</table>
| Comment  | -Bill: one other potential benefit is that a separate forum would tend to help improve performance and increase scalability  
-Petr: there are many reasons to have our own forums that are well integrated with Moodle etc. |

The example of discussion issue in moodle tracker system is shown in Table 6.8. There are some discussion in comments, including its advantage and disadvantage.
Studied Design

In this study, we try to construct assurance cases that can mention about unnoticeable or subtle problems. We use a discussion issue as information for constructing this kind of assurance cases. There are 2 types of discussion issue; the first one is a collaboration issue and the second one is a contradiction issue (see Figure 6.4). Those issues are defined as follow:

- **Collaboration Issue**: in the “title”, “description” or “comments”,
  - there is at least 1 what, where, when, why or how question proposed, and
  - there is at least 1 solution, idea, reason or any other related information presented in some subsequence comments

- **Contradiction Issue**: in the “title”, “description” or “comments”,
  - there is at least 1 solution or idea presented, and
  - there is at least 1 comment arguing with the solution or idea in some subsequence comments

There are 3 students in total participated in this study. Those are hired to find discussion issues by reading issues from issue number 8,000 to 8,300 and to construct the design rationale documents. Then, the expert group review the results of design rationale documents and construct safety contracts from those rationale documents. Finally, expert group construct assurance cases from those safety contracts.

Construction of Design Rationale from Discussion Issue

Some of design rationale documents are shown in Figure 6.5, Figure 6.6 and Figure 6.7.
Figure 6.5: The example results of design rationale documents (a)
Figure 6.6: The example results of design rationale documents (b)
Figure 6.7: The example results of design rationale documents (c)
Construction of Safety Contracts from Design Rationale

We construct safety contracts from each design rationale document from the previous step. In order to build safety contracts, we use conceptual mapping of the design rationale to safety contracts which is proposed in Section 4.5.2. Some of safety contracts are shown in Table 6.9, Table 6.10 and Table 6.11.

Table 6.9: An example results of safety contracts (a) for the authentication

<table>
<thead>
<tr>
<th>Definition</th>
<th>Moodle version 1.9 STABLE and above</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumption</td>
<td>Enforce password complexity for user authentication</td>
</tr>
<tr>
<td>Guarantee</td>
<td>The improvement of the security level for the user authentication</td>
</tr>
<tr>
<td>Disagreement</td>
<td>None</td>
</tr>
<tr>
<td>Evidence</td>
<td>Moodle issue number MDL-8031</td>
</tr>
</tbody>
</table>

The safety contract in Table 6.9 is constructed from the design rationale document which is shown in Figure 6.5, where the assumption is if Moodle enforces password complexity for user authentication, then there is the improvement of the security level for user authentication.

Table 6.10: An example results of safety contracts (b) for the Moodle forum

<table>
<thead>
<tr>
<th>Definition</th>
<th>Moodle version 1.8, 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumption</td>
<td>Implement forum by integrating with the existing forum such as PHPBB or VBulletin</td>
</tr>
<tr>
<td>Guarantee</td>
<td>There are some improvements of scalability and reliability, if the forum goes down for some reason, the core Moodle educational content delivery system would not be compromised</td>
</tr>
<tr>
<td>Disagreement</td>
<td>May not well integrated with the core Moodle</td>
</tr>
<tr>
<td>Evidence</td>
<td>Moodle issue number MDL-8036</td>
</tr>
</tbody>
</table>

The safety contract in Table 6.10 is constructed from the design rationale document which is shown in Figure 6.6, where the assumption is if Moodle im-
plement forum by integrating with the existing forum, then Moodle can improve the scalability and reliability. However, it may not well integrate with the core Moodle.

Table 6.11: An example results of safety contracts (c) for the language interface

<table>
<thead>
<tr>
<th>Definition</th>
<th>Moodle version 1.9 STABLE and above</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumption</td>
<td>Have language editing interface by default created local version of the language files</td>
</tr>
<tr>
<td>Guarantee</td>
<td>There are some improvements of usability. Moodle has lots of administrators that are not technically trained. This can support the non-technical user</td>
</tr>
<tr>
<td>Disagreement</td>
<td>It is expensive and it take some time to implement the new GUI (interface), and</td>
</tr>
<tr>
<td>Evidence</td>
<td>Moodle issue number MDL-8012</td>
</tr>
</tbody>
</table>

The safety contract in Table 6.11 is constructed from the design rationale document which is shown in Figure 6.7, where the assumption is if Moodle has language editing interface, then Moodle can improve the usability for non-technical user.

Construction of Assurance Cases from Safety Contracts

We construct assurance cases from each safety contracts from the previous step. In order to construct assurance cases, we use the rules and patterns which are proposed by Sljivo [20] for automatic generation. Note that each assurance case in this study is a piece of the whole assurance cases. The example results of assurance cases are shown in Figure 6.8, Figure 6.9.
Figure 6.8: A part of assurance case for authentication
Figure 6.9: A part of assurance case for the moodle forum
Summary

In this study, we conducted the case study to show the construction of assurance cases from design rationale document. The obtained assurance cases can capture important information that is likely to be unnoticeable. Moreover, this study also showed that the approach allows system assurance people to speed up the creation of assurance cases. However, the assurance case from each safety contract is a piece of the whole assurance cases. Thus, more efforts and researches need to tackle and combine them together, and this would lead to the next future work.
Chapter 7

Conclusion and Future Work

In this thesis, the authors address the problem of constructing assurance cases such as high cost and time of constructing and maintaining assurance cases. We have presented several approaches of retrieving relevant documents from existing documents, artifacts, or products in software development processes, which can be used as materials for constructing assurance cases (Described in Section 4). Those approaches allow us to reduce the cost of constructing assurance cases by reducing the time needed to understand and select the relevant documents from all the whole documents.

We performed an experiment to evaluate those proposed approaches of retrieving relevant documents. The results showed that the proposed approach which combines document retrieval and formal concept analysis provides the best accuracy for constructing assurance cases. Furthermore, a concept lattice of this approach is shown as being effective for searching the relevant information. More precisely, it provides the descriptive labels offering users additional guidance for exploring necessary information.

We also proposed the new approach for constructing assurance cases from design rationale document. The approach allows the system assurance people to speed up the creation of assurance cases and the obtained assurance case is expected to capture important information that is likely to be unnoticeable.

Additionally, we proposed a new indicator for evaluating the quality of assur-
ance cases using three evaluation metrics, i.e. claim coverage, argument coverage and defect density. The results suggest that constructing assurance cases by using our approaches achieved the same quality level as manual approach with less time and efforts.

In conclusion, our approaches can reduce a time and a cost of constructing assurance cases, and the quality of assurance cases is also acceptable. We plan to move this research in several directions. First, we plan to improve the quality of labels of the lattice concept nodes. Second, we plan to automatically suggest an appropriate query for a user such as query reformulation. Third, we plan to use the supplemental notation of GSN which is proposed by Takai [47] for supporting the translation of our extended safety contracts to assurance cases. This supplemental notation of GSN aiming for dealing with changes of assurance cases, thus it is more suitable for the translation than the general GSN.
Chapter 8

Contribution

The followings are the summary of our contributions.

- Our proposed approaches (Section 4.3, 4.4) are effective to use for software of any size and becomes more valuable while the size and complexity of software is increasing, in which users can reduce the time and the cost of construction (Table 6.2, 6.3, 6.5, 6.6).

- This research shows the new approach (Section 4.5) for the construction of assurance case from design rationale documents. The approach allows assurance case people to speed up the creation and to capture important information that is likely to be unnoticeable.

- Our proposed approaches (Section 4.3, 4.4) enable third parties (cf. IV&V), who do not relate to a system, to construct an assurance case easily by using common keywords for searching relevant information (Table 6.2, 6.3).

- This research shows the new indicator (Section 5) for evaluating the quality of an assurance case which is very important to confidence of the system.
List of Publications

Refereed


References


Workshop on assurance cases for security - the metrics challenge, DSN 2007,
2007.

[27] Robin E. Bloomfield and Bev Littlewood. Multi-legged arguments: The
impact of diversity upon confidence in dependability arguments. In DSN,

[28] Ewen Denney, Ganesh Pai, and Ibrahim Habli. Towards measurement of
confidence in safety cases. In Proceedings of the 2011 International Sym-
posium on Empirical Software Engineering and Measurement, ESEM ’11,

International Conference on Software Engineering, ICSE ’13, pages 1161–


April 2012.

[32] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allo-

[33] Li Wei and Andrew McCallum. Pachinko allocation: DAG-structured mix-
ture models of topic correlations. ICML ’06: Proceedings of the 23rd inter-


