INFORMATION

SCIENCE

TECHNICAL

REPORT

NAIST-IS-TR2014002

ISSN 0919-9527

Automatic Approach to Prepare Information for Constructing an Assurance Case

Khana Chindamaikul, Toshinori Takai,

Daniel Port, and Hajimu Iida

December 2014

NAIST

〒630-0192

奈良県生駒市高山町 8916-5

奈良先端科学技術大学院大学

情報科学研究科

Graduate School of Information Science

Nara Institute of Science and Technology

8916-5 Takayama, Ikoma, Nara 630-0192, Japan

Automatic Approach to Prepare Information for Constructing an Assurance Case

Khana Chindamaikul, Toshinori Takai, Daniel Port**, and Hajimu Iida

Nara Institute of Science and Technology

Abstract. We address the problem of constructing an assurance case and propose an approach to extract information from an issue tracking system that can be used to construct an assurance case. This approach takes advantage of information searching and data mining techniques to identify relevant information, which can also be used as materials for constructing an assurance case. This paper gives an overview of an approach and reports the result of the experiment and gives an evaluation. Experimental results suggest that this approach can be effective in reducing the time and cost for constructing an assurance case with acceptable quality.

Keywords: Assurance Case, Document Retrieval, Topic Modeling, Formal Concept Analysis

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. 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 [1]. 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 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" [2]. This approach is mainly used in critical systems such as automotive, aviation, railway and nuclear power plants systems. Some safety and security related standards in industry require an assurance case, e.g. [3],[4].

Assurance cases can often be a large size of documents, and thus the cost for constructing and maintaining go far beyond a reasonable level. Moreover, assurance cases is usually constructed manually. Such hand-made approaches cannot avoid defects on statements of claims, strategies for arguments and evaluation of evidence.

Our work aims to reduce the cost of constructing an assurance case by reducing the time needed to understand and select from all the documents. Instead

** Department of Information Technology Management, University of Hawaii at Manoa

of having to read all the possible documents, our work automatically selects the relevant documents from existing documents, artifacts, or products in software development processes. Among them, we focus on issues in issue tracking systems, which can be used as materials for constructing an assurance case. In order to evaluate the effectiveness of our approaches, we introduce an evaluation metric. This metric is an indicator for evaluating the quality of an assurance case. Note that confidence of the system given by an assurance case is indicated by the quality of an assurance case.

We present a concrete illustration of our approach. We have performed an experiment of the approach with students. We also measured the quality of obtained assurance cases by the proposed metric. The results showed that the approach provides better results than a manual approach when considering precision and recall per effort. This suggests that our approach can reduce the time and the cost to construct assurance cases. Moreover, the quality of assurance cases obtained by our approaches is comparable to a manual one.

This paper is organized as follow. Section 2, we begin by introducing some background information. Section 3, we briefly describe about related work and our previous investigation. Section 4, we describe a construction and indicator. Section 5, we present an experiment with students and discuss about the results. Section 6, we draw our conclusion and future work. Section 7, we describe the contribution of our work.

2 Background knowledge

An assurance case (AC) 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. Generally, a claim of an assurance case is a risk-related requirement of the system to be assured, e.g. safety, security, or dependability. Evidence is data supporting the claim holds, which can be either facts, assumptions or other ACs. An argument is a description showing how the evidence supports or justifies the claim, which can be deterministic, probabilistic or qualitative. Fig. 1 illustrates an example of structure of an assurance case.

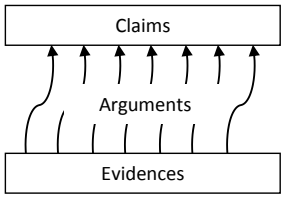

Fig. 1: A structure of an assurance case (modified from [5])

Goal Structuring Notation (GSN) [5] is one of the most widely used graphical representation of the structured argument for an assurance. This kind of representation has been designed to enable an assurance case in term of easy

understanding and can be manipulated by machine. There are 5 types of node in *GSN*; *goals* are claims of ACs, which are modeled as rectangles; *strategies* are arguments of ACs, which are written in parallelograms; *solutions* are evidences of goal, which are express in ovals; *contexts* are a definition or reference, which are expressed in a box with rounded corners; *undeveloped* are goals which are under construction, are represented by diamond box. A goal is decomposed through a strategy. Every leaf is either evidence or undeveloped for supporting a goal. And a context is attached to a goal or strategy to describe the definition. An example of ACs in GSN is shown in Fig.6.

Document retrieval (DR) is a technique to elicit several documents related to a given query from a large set of documents [6]. Each document is treated as an unstructured text. A query can be a sentence or a set of several words. We utilize a document retrieval method in Moodle tracker system[1] as a text search engine for finding related issues to user queries.

Topic modeling (TM) is a technique for automatically extracting semantic topics from a collection of text documents [7]. 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 LDA[8], PAM[9] have been proposed. We use *Mallet*[2] as a tool for TM, which is based on LDA. Fig. 2(a) shows a set of topics for each document, e.g. document 69 relates to topic 35 and 13. Fig. 2(b) shows a set of words for each topic, e.g. topic 13 relates to security. Note that a proportion of each topic shows how much topic relates to a document.

#doc	name	topic	proportion	topic	proportion
69	BugID8069.txt	35	0.081590739	13	0.071975355
70	BugID8070.txt	86	0.183348467	31	0.061046309
71	BugID8071.txt	30	0.06233451	13	0.06233451

(a)

Topic	A set of words for each topic
11	modules current feature imo part
12	moodle http org check sourceforge
13	login auth password security case

(b)

Fig. 2: Results of Topic Modeling

Formal concept analysis (FCA) is a way for representing relations between concepts [10]. An input of FCA is called 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. A concept lattice is a lattice whose nodes are the set of *concepts* where a concept is a maximal collection of objects that have common properties. Edges in concept lattice represent subset relation with respect to objects (or equivalently properties).

[1] https://tracker.moodle.org/
[2] http://mallet.cs.umass.edu/topics.php

3 Related Work

There are some guidelines for constructing assurance cases, especially safety case, in industry(e.g.[11–13]). The early systematic approach is to use patterns of ACs[14]. Some researchers proposed translation methods from existing products, e.g. design document and risk analysis results, to assurance cases[15, 16] but those methods requires specific structure for source documents. Strunk and Knight[17] used problem frames to create and structure the implementation goals and contexts of assurance cases written in a goal structuring notation. D. Jackson *et al.* [18] showed a concrete illustration of an approach to constructing a dependability case for the control software of a medical device.

There are some proposals to give methods for evaluating confidence of assurance cases. Kelly [19] proposed several aspects to check for review activities of assurance cases. Bloomfield and Littlewood [20] 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.* [21] proposed a quantitative measure mainly focusing on the coverage of arguments. Denny *et al.* [22] also proposed another method to evaluate confidence of assurance cases based on Bayesian Networks. Goodenough *et al.* [23] introduced a notation, called a *confidence map*, which is expected to be used with GSN to show the status of the current confidence.

The procedure of the extended approach we propose later, which uses the FCA technique, is inspired by the study of Cho and Richards[24] where formal concept analysis is used to improve information retrieval from the Web.

In our previous work [25], we presented four approaches[3](a), (b), (c) and (d) to find informative issues in issue tracking system for constructing an assurance case and 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 ACs 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 an assurance case.

In this paper, we start from (d) as the base approach and propose an extended approach. We also conduct a larger experiment with several participants. Moreover, we propose a metrics to evaluate the quality of an assurance case directly.

4 Proposed Construction and Indicator

4.1 Proposed Construction of ACs

Base approach [25]: First, a DR technique is used to find possibly related issues to the given queries. Then, issues each of which has the same topic are grouped into a same group by TM. After that, user selects relevant issues that relate to an assurance case. Finally, user constructs an assurance case from the relevant issues.

[3] (a) read all issues, (b) using TM, (c) using DR, (d) using DR and TM

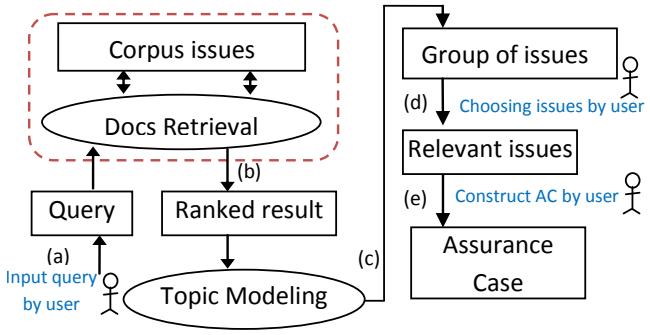

Fig. 3: Base approach: preparation by using DR and TM

Our base approach to prepare information for constructing an AC is shown in Fig. 3. There are 5 steps in this approach.

(a) Formulating a query: user selects words or sentence that relate to the claim for an AC.
(b) Searching document: a set of issues that relates to a query is retrieved (we call the list of ranked result) by a DR engine.
(c) Applying TM: issues from the ranked result are inferred using a topic modeling tool. Then, issues 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 (e.g. Fig. 2(b)). 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*.
(d) Screening the result: user selects issues from a small size of issues 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 issues without careful reading inside of the given issues.
(e) Construct an AC from the retrieved relevant issues.

Extended approach : this approach uses FCA with a base approach. There are 7 steps in this approach. Fig. 4 shows a flow of this approach.

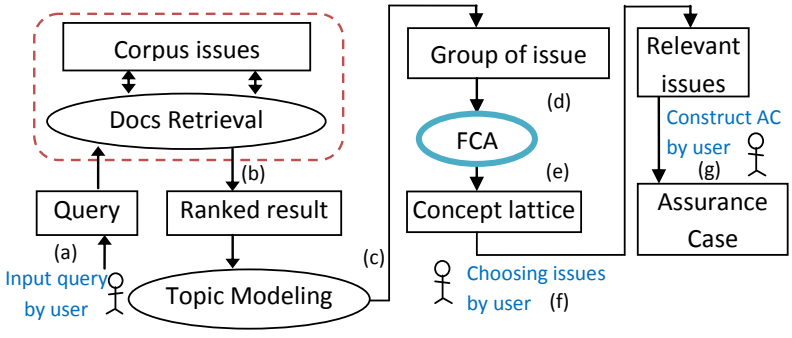

Fig. 4: Extended approach: preparation by using DR, TM and FCA

(a) to (c) are the same as the previous approach.

(d) Formulating a formal context: all issues in a *group of interest* are selected as objects of formal context. And the top 2 topics from a set of topics for each issue in a group of interest are selected as properties of formal context.
(e) Applying formal concept analysis: *formal context* from the previous step is used as an input of FCA. Then FCA builds the set of concepts for a given context (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 the result: In this step, user can easily select the issue from the resulting concept lattice by reading the annotated description of each concept node and seeing a relation among concept nodes.
(g) Construct an AC from the retrieved relevant issues.

Using information from the previous approaches, user can express each of them in arguments. In our approach, users are assumed to use GSN [5] to represent ACs.

4.2 Indicator

The confidence of the system is indicated by the quality of an AC. We propose the new indicator for evaluating the quality of an assurance case from each approach, which is consisting of triple (c, a, d) where c is claim coverage, a is an argument coverage, and d is a defect density.

Claim coverage: this metric is used for evaluating the coverage of an AC claims about the type of known issues in the system. For this metric, we assume a *reference issues list*. For example, if the claim is related to safety, then the types of accidents list can be a reference issues list and we can call "the type of reference accident list". Using the notion of reference issues lists, the claims coverage is defined as follow:

$$Claim\ \ coverage = \frac{\#Types\ of\ reference\ issues\ which\ are\ covered\ by\ AC}{\#Types\ of\ issues\ in\ reference\ issues\ list}$$

The claim coverage represents how much an AC is mentioning about types of reference issue list. This list is provided by the developer of a system or agreement among related stakeholders, and typically appears in the document or website of a system. In this paper, we focus on the reference issue list which relates to security, we call reference vulnerability list. Note that the 100% of this metric does not mean the system completely satisfy the top goal of the AC.

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 Fig. 5 are elements

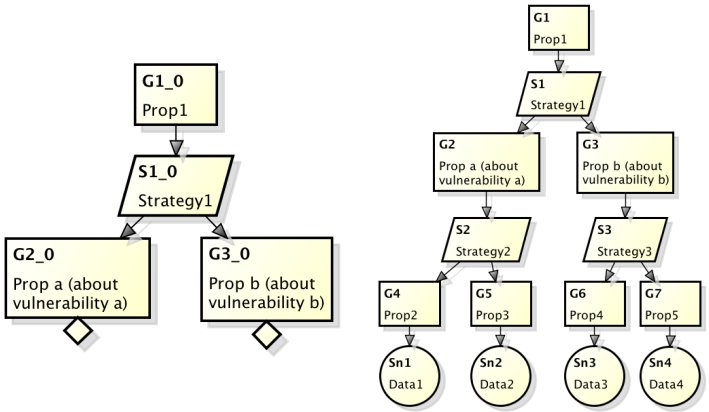

Fig. 5: Comparison between ACs

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 Fig.5 has richer arguments than the left-hand side, and thus the evaluator of this GSN could have more confidence. In this evaluation, we propose the relative measurement to evaluate results of the experiment. The relative measurement is an approach to compare ACs with the master assurance case.

Since in the experiment, the AC construction part has been executed by a single expert group, it can be assumed that there is no conflict among the obtained ACs. That leads us to construct a "master" assurance case which contains all the information of ACs from all groups. In other words, each group's AC can be obtained by trimming some branches from the master AC. For example, if we assume two assurance cases $\mathcal{A}_1$ and $\mathcal{A}_2$ in the Fig. 6 are obtained, then we can construct the master assurance case of them as $\mathcal{A}_M$.

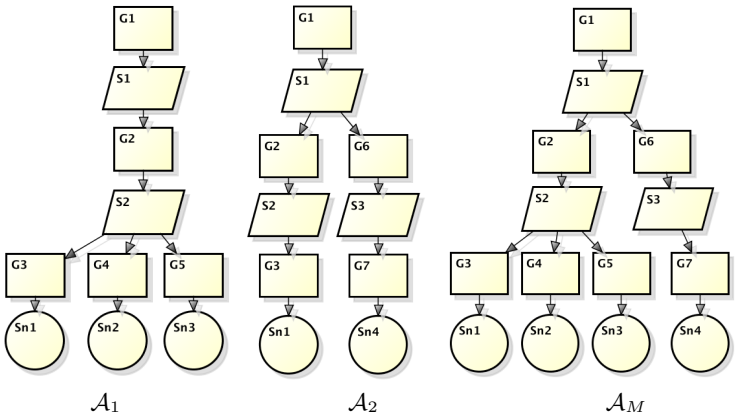

Fig. 6: Master Assurance Case $\mathcal{A}_M$ for $\mathcal{A}_1$ and $\mathcal{A}_2$

Moreover, since it can be expected that a goal in a higher position (i.e. nearer to the root) has a more significant role than any lower one, we adopt the following measurement. For example, AC $\mathcal{A}_1$ in the Fig. 6 lacks the sub-tree of G6 in compared with AC $\mathcal{A}_2$ whereas AC $\mathcal{A}_2$ lacks the sub-trees of G4 and G5 in compared with AC $\mathcal{A}_1$. In this case, it can be expected that the lack of the sub-tree of G6 has larger impact than those of G4 and G5.

Let $\mathcal{A}$ be an AC given in a GSN, which is under evaluation. $\mathcal{A}_M$ be the master AC, which is also written in a GSN. According to the definition of master ACs, there exists the injective function ι from $\mathsf{goal}(\mathcal{A})$ to $\mathsf{goal}(\mathcal{A}_M)$ where $\mathsf{goal}(\cdot)$ represents the set of all the goals appeared in $\mathcal{A}$, mapping a goal g to $\iota(g)$ where $\iota(g)$ has the same statement of g[4]. Before defining argument coverage for ACs, we give argument coverage for goals. Let $[\![\cdot]\!]$ be a function taking a goal of the AC to be evaluated and 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 undevelopped, 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 $g_i \in \mathsf{goal}(\mathcal{A})$ 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 ACs is defined as $[\![A]\!] = [\![\mathsf{root}(A)]\!]$ where root represent the root node.

For example, in $\mathcal{A}_1$, G2, G3, G4, and G5 are all evaluated as 1 and then G1 is 0.5 since it lacks the argument about G6 in compared with $\mathcal{A}_M$. So, the argument coverage of $\mathcal{A}_1$ is 0.5. On the other hand, you can calculate the argument coverage of $\mathcal{A}_2$ as 0.66 because it lacks the argument about G4 and G5 in compared with $\mathcal{A}_M$. Note that if you compare $\mathcal{A}_1$ and $\mathcal{A}_2$ only from the view of numbers of the goals, those are the same.

Defect density: we assert that quality of ACs is decreased as a number of ACs defects are presented. The defect density is used to measure how many defects occur in an AC. Kelly et al.[19] proposed the review aspects for ACs.

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

$$Defect\ \ density = \frac{\#Defects\ \ in\ \ AC}{\#Elements\ \ in\ \ AC}$$

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

5 Experiment and Results

5.1 Setting environments

We decide to construct an AC for each properties of the system such as security, maintainability, usability and reliability. Among them, here we only describe security property. The results of other properties are discussed at the end of Section 5.4.

For the security property, we specified that queries were to be restricted to the narrow area which relate to security. The target system of experiment is a course management system for educational institutes, which is called Moodle. Issues in *Moodle tracker system*[5](from issue no. 8000 to 8500) is used as information for constructing an AC. The experiment was executed by the following groups.

Groups: There are 8 people in total participated in this experiment. We divide those people into 3 groups; group I has 2 people, group A has 3 people and group B has 3 people. We also provide the background knowledge about the Moodle system and properties (e.g. security, maintainability, usability and reliability) to those groups in order to minimize the threats to validity of our results and against unfair comparison.

- Group I: This group is familiar with the Moodle system and has been recruited to constructing an AC via manual approach (read all issues).
- Group A: This group is used to prepare information for constructing an AC via a base approach (using DR and TM).
- Group B: This group is used to prepare information for constructing an AC via an extended approach (using DR, TM and FCA).

Tasks for each group our experiment was conducted along the following steps:

Group I: the members of this group read all issues from issue number 8000 to number 8500, and determined the relevant issues which relate to the given properties. The reason we choose this range is that security is the rarest one among the other properties and a number of security issues in this range is bigger than other range. Note that constructing meaningful arguments in ACs is required plenty of issues.

Group A: each person in Group A created 3 queries and applied Base approach. Finally, Group A searched the relevant issues in the group of interest.

Group B: each member of Group B was assigned to one person of Group A exclusively, and used the same queries as the assigned person created, applied Extended approach to those issues, and searched the relevant issues in the concept lattice. The reason Group B used same queries as Group A is that we want to avoid unfair comparison between those approaches. Quality of the result can be affected by the quality of query.

[5] Moodle tracker system is a bug tracking system which records and manages all issues related to Moodle and related systems.

5.2 Results of approach performance

We show the performance of each approach by measuring a *precision per effort* (*PPE*) and *recall per effort (RPE)* from each group. Note that a total number of relevant issues are calculated by the union of a number of relevant issues from Group I, A and B. Precision and recall are defined as follows.

$$Precision = \frac{|\{relevant\ docs\} \bigcap \{retrieved\ docs\}|}{|\{retrieved\ docs\}|}$$

$$Recall = \frac{|\{relevant\ docs\} \bigcap \{retrieved\ docs\}|}{|\{total\ relevant\ docs\}|}$$

Table 1: Experimental results from group I

Group I							
	#Retrieval	#Selection	#Correct	Total relevant	Effort	PPE(%)	RPE(%)
Tester.1	500	12	11	14	500	0.004	0.157

Table 2: Experimental results from group A

Group A							
	#Retrieval	#Selection	#Correct	Total relevant	Effort	PPE(%)	RPE(%)
Tester.2	15	8	6	14	15	2.667	2.857
Tester.3	16	10	8	14	16	3.125	3.571
Tester.4	17	6	2	14	17	0.692	0.840

Table 3: Experimental results from group B

Group B							
	#Retrieval	#Selection	#Correct	Total relevant	Effort	PPE(%)	RPE(%)
Tester.5	15	9	7	14	15	3.111	3.333
Tester.6	14	6	5	14	14	2.551	2.551
Tester.7	15	4	1	14	15	0.444	0.476

Table. 1, 2 and 3 present the results of section 5.1. The 1^{st} columns of each table are participator. The 2^{nd} columns are a number of retrieval issues from each approach. The 3^{rd} columns are a number of selected issues by user. The 4^{th} columns are a number of relevant issues from selected issues. The 6^{th} columns are a number of issues which are read by user.

As we can see in the 6^{th} column of each table, approaches for Group A and B reduce a size of documents for consideration from 500 to around 15 issues.Then that means user can save time for finding relevant issues. We also compare a precision per effort and a recall per effort of each group as shown in Fig. 7.

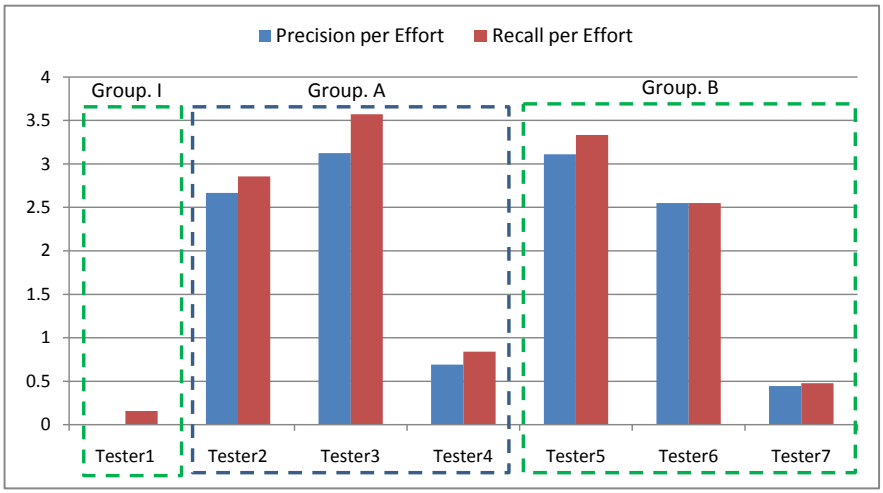

Fig. 7: A precision per effort (PPE) and a recall per effort (RPE) of each group

As we can see in Fig. 7, PPE and RPE of group A and B are higher than Group I. This mean user in Group A and B could find more relevant information than Group I while making same effort. However, PPE and RPE of group A and B are not significantly different. The reasons are discussed in Section 5.5.

These results also show that our approaches increase precision per effort and recall per effort of finding relevant information compare to a manual approach.

5.3 Obtained ACs

Since the obtained ACs are too large for presenting, some parts are shown in Fig.8. The top goal of this AC 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, see Fig. 8(a). An example of a leaf part is shown in Fig.8(b). Table 4 shows the size of obtained AC.

Table 4: Size of the obtained ACs

	AC of group I	AC of group A	AC of group B	Master AC
#Goal	21	22	18	26
#Strategy	14	13	12	16
#Solutions	8	6	4	8

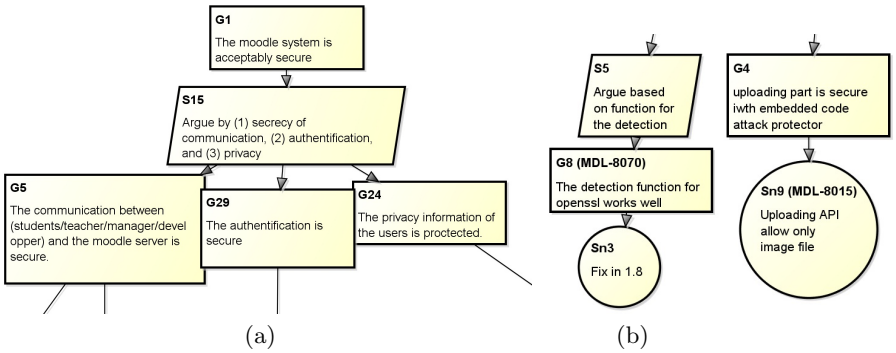

Fig. 8: Obtained an assurance case

5.4 Results of an assurance case quality

Claim coverage: this measurement is based on the reference vulnerability list, which is provided by the Moodle system (see moodle docs[6]). This list does not mean a complete list of vulnerability for Web application in general. Rather, that can be regarded as an agreement among developers.

The results of an evaluation by measuring claim coverage for each an assurance case are shown in Table 5.

Table 5: Claim coverage of each an assurance case

	AC of group I	AC of group A	AC of group B
#Vulnerability which an AC cover	9	8	8
#Reference vulnerability	15	15	15
Claim coverage	0.6	0.533	0.533

In Table 5, the claim coverage of each an AC is not so different. This indicates that coverage of ACs from our approaches are comparable to the manual one. Moreover, our approaches take less time than a manual one.

Argument coverage: The results of an evaluation by measuring argument coverage for each an assurance case are shown in Table 6.

As we can see in Table 6, there is no significant difference of argument coverage between each an AC. This result indicates that quality of ACs from our approaches are comparable to the manual approach.

[6] http://docs.moodle.org/dev/Security#Common_types_of_security_vulnerability

Table 6: Argument coverage of each an assurance case

	AC. from group I	AC. from group A	AC. from group B
Missing elements	G11,G25,G30	G3,G11	G3,G14,G18,G25
Argument coverage	0.801	0.819	0.778

Defect density: Since, the AC construction part has been executed by only single expert group. So, the defect density of each an assurance case is supposed to has the same value.

This experiment results has demonstrated that our Base and Extended approaches provide a convenient ways to prepare information for constructing an AC with accurate information. Furthermore, the evaluation results indicate that quality of assurance cases from our approaches is comparable to a manual one.

As we already said, we also experimented on other properties such as availability, usability, maintainability. However, for those properties, only few correct issues could be obtained by both Base and Extended approaches. We will discuss about this in Section 5.5.

5.5 Discussion

This experiment, FCA technique did not make any contribution for improving a base approach. Several reasons can be considered. (1) The labels of the concept nodes were not so meaningful. In this experiment, we use the outputs of TM as the labels of FCA directly. Although an output of TM contains keywords representing one topic, it was rare that a user can capture the characteristics of each topic from the set of the keywords. (2). Some students did not read labels because they did not understand clearly about how to utilize labels.

Experiments on non-security properties did not have good results. We can consider the following reason. (1) Students were not familiar with those properties, so they could not give appropriate query. For the security property, students could easily imagine keywords like password, login, etc. Since those keywords are directly used in the discussions and details of the issues. On the other hand, for other properties, developer had some discussion in more specific words; those words are rarely used for general sub-properties, countermeasures, etc. Note that the quality of a query is crucial in our approach.

6 Conclusion and Future works

The main purpose of this research is to reduce time of searching and understanding relevant documents, and also to reduce cost of constructing an assurance.

We proposed two approaches to achieve our goal and we have performed experiment of our approach with student testers. The base approach, which combines document retrieval and topic modeling, provides better results than a manual approach when considering precision and recall per effort. This means

we can reduce the time and the cost to construct ACs. However, there is no significant difference in results between the base approach and the extended approach which combines base approach and FCA.

We also proposed a new indicator for evaluating the quality of ACs from each approach using three evaluation metrics, i.e. claim coverage, argument coverage and defect density. The results suggest that constructing an assurance case by using our approaches achieved the same quality level as manual approach. Moreover, ACs from our approaches can cover major areas of security.

In conclusion, our approaches reduce time and cost of constructing ACs, and quality of ACs is also acceptable. We plan to move this research in several directions. First, we plan to compare our approach with other different strategies such as approach which combines document retrieval and FCA. Second, we plan to improve quality of labels of the concept nodes by using queries as labels. Third, we plan to automatically suggest an appropriate query for user such as query reformulation. Finally, we plan to evaluate the impact of FCA.

7 Contribution

The contributions of this paper can be enumerated as follows.

- Our approach is effective to use for software of any size and becomes more valuable while the size and complexity of software is increasing, in which we can reduce time and cost of construction.
- Our approach enables 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.
- This paper showed the new indicator for evaluating the quality of an assurance case which is very important to confidence of system.

References

1. Software Engineering Institute, Carnegie Mellon University, `http://www.sei.cmu.edu/dependability/tools/assurancecase/`
2. Bishop, P. G., Bloomfield, R. E.: A Methodology for Safety Case Development. In: F. Redmill, T. Anderson (Eds.), Industrial Perspectives of Safety-critical Systems: Proceedings of the Sixth Safety-critical Systems Symposium, Birmingham (1998) . London, UK: Springer. ISBN 3540761896
3. International Standard-ISO 26262-Road vehicles-Functional safety, International Organization for Standardization, (2011)
4. IEC 62278, "Railway application - The specification and demonstration ofdependability, reliability, availability, maintainability and safety (RAMS)", March, 2002
5. Weaver, R.A., Kelly, T.P.: The Goal Structuring Notation - A Safety Argument Notation. Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases, (July 2004)
6. Samuel, D.: Overview of the Full-Text Document Retrieval Benchmark. In the Benchmark Handbook for Database and Transaction Processing Systems (1993)

7. David M. Blei.: Probabilistic topic models. Commun. ACM 55, 77-84. , (2012)
8. David M. Blei, Andrew, Y. Ng., Michael, I.J.: Latent dirichlet allocation. J. Mach. Learn. Res. 3 (2003)
9. Wei, L., Andrew, M.: Pachinko allocation: DAG-structured mixture models of topic correlations. In Proceedings of the 23rd international conference on Machine learning, (2006)
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations (1st ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA. (1997)
11. GSN contributors. GSN community standard version 1.0, (2011)
12. Adelard (Firm). ASCAD /az-kad/: Adelard Safety Case Development Manual. Adelard, (1998)
13. Railtrack Plc. Engineering Safety Management: Yellow Book. Fundamentals and guidance. Number no. 3, v. 1-2. Railtrack PLC, 2000.
14. Kelly, T.P., McDermid, J.A.: Safety case construction and reuse using patterns. In Peter Daniel, editor, Safe Comp 97, pages 55-69. Springer London, (1997)
15. Nurlida Basir, Ewen Denney, and Bernd Fischer. Deriving safety cases for hierarchical structure in model-based development. In Erwin Schoitsch, editor, SAFECOMP, volume 6351 of Lecture Notes in Computer Science, pages 68-81. Springer, (2010)
16. Ewen Denney and Ganesh Pai. A lightweight methodology for safety case assembly. In F. Ortmeier and P. Daniel, editors, Proceedings of the 31st International Conference on Computer Safety, Reliability and Security (SAFECOMP 2012), volume 7612 of LNCS, pages 1-12. Springer-Verlag, Sep. (2012)
17. Elisabeth A. Strunk, John C. Knight.: The essential synthesis of problem frames and assurance cases. Expert Systems, 25(1):9-27, (2008)
18. Joseph P. Near, Aleksandar Milicevic, Eunsuk Kang, and Daniel Jackson. A lightweight code analysis and its role in evaluation of a dependability case. In Proceedings of the 33rd International Conference on Software Engineering, ICSE '11, pages 31-40, New York, NY, USA, 2011. ACM.
19. Kelly, T.P.:Reviewing Assurance Arguments - A Step-by-Step Approach. In Proceedings of Workshop on Assurance Cases for Security - The Metrics Challenge, Dependable Systems and Networks (DSN), July (2007)
20. Robin E. Bloomfield and Bev Littlewood. Multi-legged arguments: The impact of diversity upon confidence in dependability arguments. In DSN, pages 25-34. IEEE Computer Society, 2003.
21. Ewen Denney, Ganesh Pai, and Josef Pohl. AdvoCATE: An assurance case automation toolset. In Frank Ortmeier and Peter Daniel, editors, SAFECOMP Workshops, volume 7613 of Lecture Notes in Computer Science, pages 8-21. Springer, 2012.
22. Ewen Denney, Ganesh Pai, and Ibrahim Habli. Towards measurement of confidence in safety cases. In Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement, ESEM '11, pages 380-383, Washington, DC, USA, 2011. IEEE Computer Society.
23. John B. Goodenough, Charles B. Weinstock, and Ari Z. Klein. Eliminative induction: a basis for arguing system confidence. In Proceedings of the 2013 International Conference on Software Engineering, ICSE '13, pages 1161-1164, Piscataway, NJ, USA, 2013. IEEE Press.
24. Woo-Chul Cho and Debbie Richards. Improvement of precision and recall for information retrieval in a narrow domain: Reuse of concepts by formal concept analysis. In Web Intelligence, pages 370-376. IEEE Computer Society, (2004)
25. Chindamaikul, K., Toshinori, T.: Constructing Assurance Case using Information From an Issue Tracking System In: IPSJ/SIGSE Software Engineering Symposium (SES2013), Tokyo, Japan, (2013)