Content-Based Conformance Assurance Between Software Research Documentation and Design Guideline

Jong-Hwan Shin[1], Du-San Baek[1], Byungjeong Lee[2],
and Jung-Won Lee[1](✉)

[1] Department of Electrical and Computer Engineering,
Ajou University, Suwon, Republic of Korea
{sjhl334,jungwony}@ajou.ac.kr, whitedusan@gmail.com
[2] Department of Computer Science and Engineering,
The University of Seoul, Seoul, Republic of Korea
bjlee@uos.ac.kr

Abstract. Research-oriented software groups are groups that carry out research on original technology for software. The groups on development phase experience poor documentation because of two reasons. One is the lack of resources (i.e. time, costs) since the development phase is much shorter than their research phase. The other one is that the artifacts they worked on research phase are rarely used on the development documents. Therefore, we propose a method that can reduce poor documentation regarding their research documents and development (R&D) documents. We construct design guidelines from best practices and represent it by queries of semantics-aware traceability links. Then, we use a semi-automated method of conformance assurance between R&D documents with guidelines. Finally, we provide an explanatory guideline to assessment results. We evaluated documents generated from our previous R&D project to show the possibility of our method. Our method can help software R&D project documents for better quality with reduced time.

Keywords: Conformance assurance · Relevance Link Information Model (RLIM) · Traceability · Documentation quality

1 Introduction

Research-oriented software group is a group to study original technology for software. These groups need to develop the software to demonstrate their research. However, their developments experience poor documentation. Because their development is done in much less time than the research phase, therefore they have very little resources in time, money and manpower for documentation [8]. Moreover, they can't directly use their research artifacts when they writing development documents since research artifacts are composed of highly-abstracted contents while development documents are composed of detailed requirements [4].

With poor documentation, they face 'technical debt'. Technical Debt is a term which describes 'the extra development work acquired when engineers take shortcuts

J.J. (Jong Hyuk) Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*,
Lecture Notes in Electrical Engineering 421, DOI 10.1007/978-981-10-3023-9_149

that fall short of best practices [2]'. Although not all technical debt is bad, technical debt grows along development because it has interests just like financial debt. Therefore, unpaid technical debt may be presented as delay of project from resource-consuming tasks like fixing complicated bugs and fault localization. Many existing studies address poor documentation since technical debt from poor documentation is frequent. However, they focus on requirements traceability of development documents and they don't consider semantics for high-level qualities such as conformance between two items. Their approaches cannot improve research artifacts because the research artifacts do not usually have requirements for development.

In this paper, we propose a content-based method for assurance of the quality of research documentation including development documents to address problems described above. First, we define design guidelines, which reflect best practices of software development with consideration of research artifacts. A design guideline is made of a goal model and an explanatory guideline explains the goal model for user feedback. Then, we transform the design guidelines as queries of semantic-aware traceability links for automatic evaluation. We used an expert system to automatically evaluate documents with the transformed guidelines. The result of evaluation is given as form of a report that user can easily understand.

2 Related Work

Since poor documentation is one of major reason of technical debt [2], there have been many studies for improving software documents. First, there are automated approaches for analysis and evaluation of document quality [7, 9, 12]. Second, there are some approaches that help minimize time-consuming tasks for reviewers on review process [3, 13].

Wilson et al. [13] perform keyword-based analysis and quality measurements of software requirements. They set quality related terms and measured how frequently those terms are presented. This work shows the need for quality control of software documents and even a simple method can improve document quality, but this can't evaluate high-level quality like traceability directly. Jain et al. [9] uses controlled natural language approach for requirements analysis. The method performs lexical analysis for conformance of template. Then, it performs semantics analysis for completeness with state machines of each requirement. The system also generates a helpful message for unfinished specifications. However, this method only focuses the syntax of requirements. Dautovic et al. [7] uses visitor pattern to traverse document contents and simple rule-checking mechanism for quality measurement. However, the rules they created focus on the structural and format therefore semantics cannot be investigated.

Shen et al. [12] represented traceability of documentation contents with simple linked list instead of complicated graph. They gave the traceability linked list to regulator for helping regulatory review process. However, they didn't consider the evaluation of document quality and didn't considered explanatory guidelines which is useful for developers. Antonino et al. [3] they suggest parameterized safety requirement templates to ensuring traceability throughout software documentations in safety-critical system domain. They used controlled natural language to avoid ambiguity and safety requirement decomposition pattern which is model-based structural guideline for expression of safety requirements.

Above methods cannot be used directly to solve problems what we focus. The limited ability regarding accuracy and evaluation scope is the main constraint of automated approaches. In case of review support method, reviews of research artifacts are not considered. Also, reviews which performed by untrained reviewer are not considered.

3 Content-Based Conformance Assurance

We take following approaches for assurance of conformance between design guideline and software research documentations. Since most of software standards that reflect best practices of software development requires requirements traceability, our method takes model-driven traceability approach. However, we extend traditional traceability to semantic-aware traceability for trace between research documents and development documents. Also, we develop a design guideline extracted from software standards for provide knowledge to our method. Further, we transformed the guideline into semantic rules and checking conformance of guideline in automated manner for reducing efforts of assessment. Finally, we present a series of explanatory guidelines which explains rationale which shows the source of the metrics to users to give better understanding of the best practices.

3.1 Relevance Link Information Model (RLIM)

Traceability is an ability to establish links between source artifacts and target [1]. This attribute is essential for every software standard. The existing traceability mainly focus on artifacts which include requirements. Also, a trace link means transitive relation while research documents have non-requirement contents and non-transitive relations [4]. We extend trace link to Relevance Link (RL) in [4] these limitations. A RL composed with 2 major components. Corresponding Items is two configuration items having relevance and Relevance Rule is a rule which corresponding items should follow. We defined 7 relevance rules in [4]. Therefore, A Relevance Link becomes a mapping between the corresponding items and relevance rule.

3.2 Expert Assessment Goal Model

We propose an Expert Assessment Goal Model which aims to representing knowledge for best practices of software development and direct measurement of conformance. We created goal models from three domains, each goal model guidelines based on international software standards, which are IEC 12207, ISO 26262 and IEC 62304. This guideline structure is combined structure of Goal, Question, Metric model [6] and Goal Structuring Notation [10] for satisfying both goals. Figure 1 shows the detailed goal model structure.

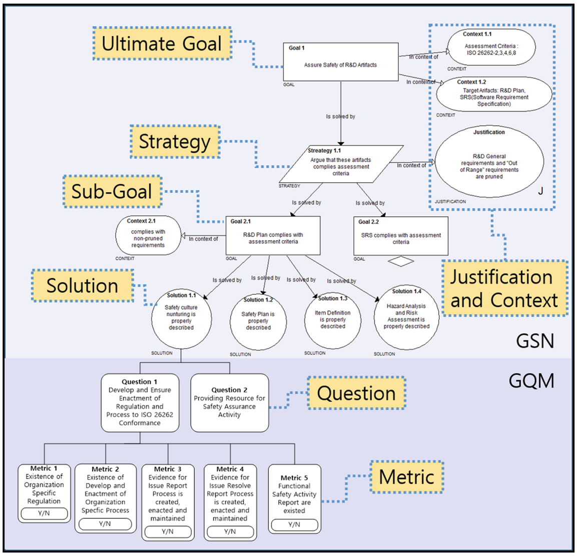

Ultimate Goal
Goal 1
Assure Safety of R&D Artifacts
GOAL
In context of
Context 1.1
Assessment Criteria : ISO 26262-2,3,4,6,8
CONTEXT
In context of
Context 1.2
Target Arifacts: R&D Plan, SRS(Software Requirement Specification)
CONTEXT
Is solved by
Strategy
Streategy 1.1
Argue that these artifacts complies assessment criteria
STRATEGY
In context of
Justification
R&D General requirements and "Out of Range" requirements are pruned
J
JUSTIFICATION
Sub-Goal
Is solved by
Is solved by
Context 2.1
complies with non-pruned requirements
CONTEXT
In context of
Goal 2.1
R&D Plan complies with assessment criteria
GOAL
Goal 2.2
SRS complies with assessment criteria
GOAL
Justification and Context
Solution
Is solved by
Is solved by
Is solved by
Is solved by
Solution 1.1
Safety culture nunturing is properly described
SOLUTION
Solution 1.2
Safety Plan is properly described
SOLUTION
Solution 1.3
Item Definition is properly described
SOLUTION
Solution 1.4
Hazard Analysis and Risk Assessment is properly described
SOLUTION
GSN
GQM
Question 1
Develop and Ensure Enactment of Regulation and Process to ISO 26262 Conformance
Question 2
Providing Resource for Safety Assurance Activity
Question
Metric 1
Existence of Organization Specific Regulation
Y/N
Metric 2
Existence of Develop and Enactment of Organization Specfic Process
Y/N
Metric 3
Evidence for Issue Report Process is created, enacted and maintained
Y/N
Metric 4
Evidence for Issue Resolve Report Process is created, enacted and maintained
Y/N
Metric 5
Functional Safety Activity Report are existed
Y/N
Metric

Fig. 1. This shows the goal model structure of Expert Assessment Goal Model. *Ultimate Goal* is a quality that documents should achieve and divided by *Sub-Goal*s. Each *Sub-Goal* is divided by *Solutions*. Below a *Solution,* there is a *Question* and a *Metric* for measurement of solution. There also are *Justification* and *Context* to showing the adequacy of a goal model to standards. Additionally, we annotated the source of each *Metric*.

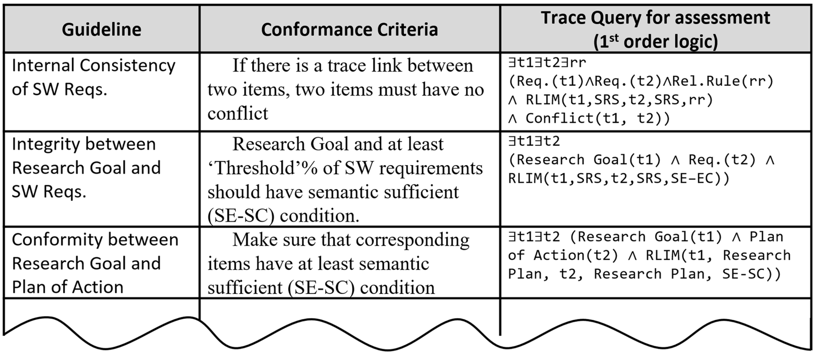
3.3 Conformance Assurance

We utilize a combined method of above methods to check conformance. First, we bridge the gap between RLIM and Expert Assessment Goal Model by translating the goal model into rules that contain relevance rules. The translations can be done differently by the scope of measurement. Next, RLIM-based expert system [5] checks conformance using rules and document RLIM. This system utilizes a rule engine which can evaluate the conformity of rule therefore we can check the conformance of design guidelines as rules with this system in automated manner. However, in case of metrics which needs semantics analysis, contents need be evaluated with other review systems that can evaluate semantics.

4 Application

We conducted a preliminary experiment to verify effectiveness of our method from our previous project. First, we transformed the metrics on the guidelines into rules. Each metrics can be defined as a query for certain relevance links or contents of documents. Figure 2 shows some examples of transformed guidelines.

Next, documents of interest are should transformed into RLIM. In this paper, we let developers to manually build document RLIMs for the best accuracy of the assessment. Then we conducted the assessment using RLIM expert system. The results of assessment displayed as shown in Table 1. Also, the explanatory guidelines given as Fig. 3.

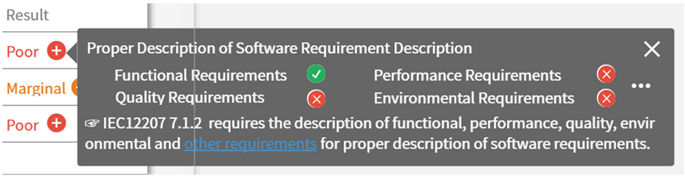


Guideline	Conformance Criteria	Trace Query for assessment (1st order logic)
Internal Consistency of SW Reqs.	If there is a trace link between two items, two items must have no conflict	∃t1∃t2∃rr (Req.(t1)∧Req.(t2)∧Rel.Rule(rr) ∧ RLIM(t1,SRS,t2,SRS,rr) ∧ Conflict(t1, t2))
Integrity between Research Goal and SW Reqs.	Research Goal and at least 'Threshold'% of SW requirements should have semantic sufficient (SE-SC) condition.	∃t1∃t2 (Research Goal(t1) ∧ Req.(t2) ∧ RLIM(t1,SRS,t2,SRS,SE-EC))
Conformity between Research Goal and Plan of Action	Make sure that corresponding items have at least semantic sufficient (SE-SC) condition	∃t1∃t2 (Research Goal(t1) ∧ Plan of Action(t2) ∧ RLIM(t1, Research Plan, t2, Research Plan, SE-SC))

Fig. 2. The example of transformed guidelines into trace query. RLIM(*t1, ci1, t2, ci2, rr*) means a relevance link which has corresponding item *t1* from *ci1* and *t2* from *ci2* with relevance rule *rr*. *Ele*(x) means the element type of x is '*Ele*'.

Table 1. The result of our preliminary application. Users receive an evaluation report in this form. We found that documentation problems of risk management process on Software R&D Plan. Also, we found that our Software Requirement Specification meet very little criteria.

Target document	Evaluation criteria	Conformance score
Software R&D Plan	Research goal, motivation, trends	Acceptable (67 %)
	Organization description	Good (88 %)
	Research strategy	Good (83 %)
	Risk management process	Poor (0 %)
Software Requirement Specification	Software requirement description	Poor (22 %)
	Requirement analysis	Marginal (33 %)
	Software test planning	Poor (0 %)



Proper Description of Software Requirement Description

Functional Requirements	✓	Performance Requirements	✗
Quality Requirements	✗	Environmental Requirements	✗

☞ IEC12207 7.1.2 requires the description of functional, performance, quality, environmental and other requirements for proper description of software requirements.

Fig. 3. This shows an explanatory guidelines of the evaluation results. The detail of assessment is provided to user for a basis of useful feedback.

5 Conclusion

In this paper, we propose a content-based method of conformance assurance between software research documents and design guideline. We first extract design guideline from software standards. Then, we transformed guidelines into rules for automated evaluation. We also used an expert system which can evaluate the conformance of rules transformed from guidelines. With our method, we expect the improvement of document quality of research groups who need to develop software. In future works, we will improve the guideline for more helpful and build practical applications for documentation model and evaluation.

Acknowledgement. This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF- 2014M3C4A7030504).

References

1. Aizenbud-Reshef, N., Nolan, B.T., Rubin, J., Shaham-Gafni, Y.: Model traceability. IBM Syst. J. **45**, 515–526 (2006)
2. Allman, E.: Managing technical debt. Commun. ACM **5**, 50–55 (2012)
3. Antonino, P.O., Trapp, M., Barbosa, P., Sousa, L.: The parameterized safety requirements templates. In: Proceedings of the 8th IEEE/ACM International Symposium on Software and Systems Traceability, pp. 29–35 (2015)
4. Baek, D., Lee, B., Lee, J.W.: Content-based configuration management system for software research and development document artifacts. KSII Trans. Internet Inform. Syst. **10**, 1404–1415 (2016)
5. Baek, D., Shin, J., Lee, B., Lee, J.: Toward development of a traceability model measuring compliance with guidelines. In: 11th KSII Asia Pacific International Conference on Information Science and Technology, pp. 37–38 (2016)
6. Basili, V.R.: Software modeling and measurement: the Goal/Question/Metric paradigm (1992)
7. Dautovic, A., Plosch, R., Saft, M.: Automatic checking of quality best practices in software development documents. In: Proceedings of the International Conference on Quality Software, pp. 208–217 (2011)
8. Goble, C.: Better software, better research. IEEE Internet Comput. **18**, 4–8 (2014)
9. Jain, P., Verma, K., Kass, A., Vasquez, R.G.: Automated review of natural language requirements documents: generating useful warnings with user-extensible glossaries driving a simple state machine. In: Proceedings of the 2nd Annual Conference India Software Engineering, pp. 37–45 (2009)
10. Kelly, T., Weaver, R.: The goal structuring Notation–A safety argument notation (2004)
11. Shen, W., Lin, C.L., Marcus, A.: Using traceability links to identifying potentially erroneous artifacts during regulatory reviews. In: Proceedings of the 7th International Workshop on Traceability in Emerging Forms of Software Engineering, pp. 19–22 (2013)
12. Thitisathienkul, P., Prompoon, N.: Quality assessment method for software requirements specifications based on document characteristics and its structure. In: Proceedings of the 2nd International Conference Trustworthy Systems and Their Applications, pp. 51–60 (2015)
13. Wilson, W.M., Rosenberg, L.H., Hyatt, L.E.: Automated analysis of requirement specifications. In: Proceedings of the International Conference on Software Engineering, pp. 161–171 (1997)