Received March 19, 2020, accepted April 10, 2020, date of publication April 14, 2020, date of current version April 30, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2987964*

An Evaluation of Quantitative Non-Functional Requirements Assurance Using ArchiMate

ZHENGSHU ZHOU, QIANG ZHI, SHUJI MORISAKI, AND SHUICHIRO YAMAMOTO



Graduate School of Informatics, Nagoya University, Nagoya 4648601, Japan

Corresponding author: Zhengshu Zhou (zhouzhengshu@gmail.com)

 **ABSTRACT** Goal-oriented NFR (Non-Functional Requirement) assurance approaches were used to qualitatively evaluate software architectures. Assurance cases using quantitative method have not been applied to evaluate NFR assurance for software architectures. This paper presents a system architecture evaluation method which is able to conduct quantitative NFR assurance evaluation for system architecture through ArchiMate. The paper also proposes an algorithm to automate the quantitative evaluation process. A questionnaire survey among software engineers and a case study on a vehicular safety monitor system were carried out to verify the necessity of the method. Additionally, we conducted an experimental design with 18 samples divided into 2 groups with the goal of comparing how the independent variables affect the dependent variables. The results of the experiment demonstrate that the proposed method achieves better NFR evaluation effect than the traditional approach. Moreover, compared with the traditional approach, the proposed method shortens the time for NFR evaluation. The proposed method is expected to be used at the early stage of software development projects for system NFR development, such as requirements analysis, system architecture design and system modeling. At present, the method has been applied by software engineers in a practical software project.



 **INDEX TERMS** Non-functional requirements, quality requirement, system architecture modeling, securityassurance, requirement engineering, ArchiMate.



**I. INTRODUCTION**

Software quality management is conducted throughout the whole process of software project development, includ-ing not only functional requirements management, but also non-functional requirements management [1]. A Functional Requirement de nes a function of a system or its component, where a function is described as a speci cation of behavior between input and output. Functional Requirements are also called behavioral requirements, because it is customary to use ‘‘should’’ sentences to describe them. For example, ‘‘The subway ticketing system should ask the user whether to print a receipt or not after the user purchases the ticket’’. Non-functional requirements (NFRs) refer to the indicators that specify criteria that can be used to judge the operation of a system, rather than the speci c behaviors of the system. NFR includes availability, reliability, usability, scalability, safety, security and maintainability [2]. For instance, an e-commerce system should have functions for product retrieval, sorting,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaobing Sun .



booking, online payment and e-mail noti cation, which are functional requirements. On the other hand, there are also other factors that affect the quality of the system, such as system reliability, user-friendly level, ability to avoid cyber-attack risks and ease of extending new functions and main-taining. These are non-functional requirements.

Unlike functional requirements which usually have clear qualitative and quantitative objectives, the realization and management of non-functional requirements are generally more dif cult than those of functional requirements. More-over, to prevent software quality accidents and avoid the risks of project development failure [16], in software design phase [28], that is to say, in the stage without the actual test environment for validation, it is necessary to evaluate the system NFR assurance level that the software system can reach [13], [20]. Therefore, NFR evaluation method for software design phase is considered to be indispensable [27].

There are three main types of goal decomposition analy-sis methods in current requirement engineering researches. The Fault Tree Analysis (FTA), the top-down deductive fault analysis methods combining low-level events with Boolean

|  |  |  |  |
| --- | --- | --- | --- |
| VOLUME 8, 2020 | This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/ | 72395 |  |
|  |  |

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

logic to analyze the undesired states of the systems, are mainly used in the elds of safety engineering and reliability engineering. The Softgoal Interdependence Graph (SIG) is used to represent NFR re nement structure, de ne the quality requirements and discuss whether the system can achieve the quality requirements. The Goal Structuring Notation (GSN) is a notation for visualizing the methods and thoughts that lead to the achievement of the objectives and the proper-ties that the system should achieve. However, these existing studies can only discuss a speci c kind of NFR such as safety or robustness, and cannot be used for quantitative evaluation and comparison of software architecture to help engineers make the optimal design decision. In view of this, this paper will propose a new method which is possible to conduct quantitative NFR evaluation for system architecture development based on a goal decomposition framework [29],

1. In this approach, not only NFR evaluation but also system visualization can be realized. A case study on an in-vehicle embedded system and an experiment are carried out to evaluate the effectiveness of the proposed method [17].

This paper is structured as follows. In this section, the research background, the purpose of this research and the result are described as an introduction. Next, in Section II, related researches are introduced. In Section III, the qual-itative NFR assurance approach through ArchiMate is described. In Section IV, a model-based quantitative NFR evaluation method using ArchiMate framework is proposed, and the detailed steps of this proposed method are de ned. In addition, an NFR assuring level calculation tool is pro-posed. In Section V, a case study is conducted to verify the effectiveness of this proposed method. After that, in Section VI, a controlled experiment is carried out to compare the proposed method with the traditional method. Last but not least, a summary of this research is made and further work is clari ed in Section VII. The contributions of our research in this paper can be summarized as follows.

A quantitative NFR assurance evaluation method based on ArchiMate is proposed.

The proposed method was applied to a vehicle embedded system for a case study, to verify the effectiveness of the proposed method and calculation tool.

We conduct a controlled experiment to compare our method with the traditional approach. The experimental data shows that the proposed method has achieved better perfor-mance in terms of ef ciency, effectiveness and stability in architecture NFR assurance evaluation than the traditional approach.

**II. RELATED WORK**

**A. NFR ASSURING APPROACHES**

As an important part of software quality assurance, software NFR evaluation method is essential for software engineering [18], [31]. To address this issue, many studies about soft-ware NFR have been done [6]. In 2004, Avizienis *et al.* had put forward the main de nitions relating to NFR, including

72396

attributes such as reliability, availability, safety, integrity, maintainability, etc. Ahmad *et al.* used LDA topic model to investigate the different NFRs software developers concen-trated on, the NFRs problems or dif culties the developers confronted with, and the evolving trends of different NFRs dif culty and focus on the evolution of time [43]. In [2], the authors rst made the basic de nitions for NFRs, and commented upon the properties that affect system quality. They also supplemented NFRs by additional de nitions, which address the threats to NFR, their attributes, and the means for their achievement. After that, D-Case, NFR frame-work and Fault Tree Analysis (FTA) and are the typical pro-posed NFR assurance solutions. GSN is a notation used for visualizing methods and thoughts that lead to the achievement of the purpose and properties that the system should achieve [5], [26]. The notation is a diagram that builds assurance cases through logic-based maps [11], [12]. D-case is an approach for modeling system safety cases using GSN. Safety case is a document for discussing the safety of the system based on the veri cation results and test results evidence, and assuring the system safety to system administrator and system users [30]. Fault tree analysis (FTA) is a top-down, deductive failure analysis approach in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower-level events [4]. FTA is mainly used in the elds of safety engineering and reliability engineering to analyze in which case systems can fail, to identify the best measures to reduce risks or to determine event rates of a safety accident or a particular system level failure. NFR framework is a qual-itative goal-oriented approach for NFR analysis [3], [14]. NFR claims and the dependencies among the NFR claims are modeled as Softgoal Independency Graph (SIG). Silva *et al.* proposed an approach for the de nition of NFR elicitation guides (ADEG-NFR) to support the requirement engineers in carrying out NFR elicitation and providing mechanisms for customer involvement in NFR de nition [38]. Köhl *et al.* proposed a standardized certi cation process in tandem with appropriate development techniques to achieve system NFR

1. DeVries and Cheng presented Soter [42], an approach for modeling and translating NFR models. Soter translates non-functional models into non-functional goal model frag-ments that can then be analyzed with the system-to-be goal model.

In addition to these qualitative approaches for NFR assur-ance, there are also plenty of quantitative methods. Kaiya *et al.* have presented Attributed Goal-Oriented RequirementsAnalysis (AGORA) to recognize the requirements con icts among stakeholders [9], in which they de ne preference and contribution for goals, and then carry out quantitative calculation by using attribute value expressions. Although the AGORA can be used to quantitatively analyze system requirements, it is not able to evaluate system architecture. In 2006, Saito *et al.* introduced a goal selection method based on KPI-based business attributes called Incremental Goal Evolution Process Methodology (IGEPM) [34]. This method was proposed to continuously improve goal graph based on

VOLUME 8, 2020

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

business conditions. Subsequently, Kokune *et al.* proposed the Fact-Based Collaboration Modeling (FBCM) [7], a goal dependency management approach based on a statistical analysis of the correlation coef cient for KPI values. The characteristic of FBCM is that it demonstrates the rationality of goal decomposition with statistical evidence. In 2014, Quantitative Assessment using NFR (QANFR) approach was introduced by Subramanian *et al.* to quantitatively analyze safety and security properties of system architecture [32]. In this approach, system safety and security properties are de ned as non-functional soft goals. The satis ce, contribu-tion, criticality metric and propagation rules are de ned on the Softgoal Interdependency Graph (SIG) for propagating attributes from bottom layer to evaluate the safety and secu-rity assurance level of the target system architecture. This approach, however, cannot clarify the quantitative relation-ships between child softgoals for goal decomposition. There-fore, in 2015, Yamamoto made an improvement by presenting the Soft Goal using Weight (SGW) approach [8], [33], which can be used to de ne the priority among sub softgoals for quantitative goal decomposition. Yamamoto also proposed a quantitative architecture evaluation method called Attribute GSN by extending traditional GSN [10]. In this method, system safety and security attribute values are assigned to the GSN nodes to conduct quantitative safety and security assurance. The assurance level of the system is calculated by accumulating the attribute values of the GSN nodes. Although the SGW approach and Attributed GSN presented a method for architecture evaluation by weight propagation and tabular calculation, the calculation method and evaluation steps were not clearly de ned. In 2018, Kuhn *et al.* proposed a scenario-based evaluation approach for embedded systems that enables automated evaluation of architecture concepts with respect to scenarios. This approach utilizes function nets contained in candidate architectures to derive main event chains of the system under development [39]. In 2019, Sumesh *et al.* presented a framework to capture the opposing NFR goals to achieve multi-objective optimization of inter-dependent stakeholders [40].

To summarize, although many efforts have been made to evaluate NFR, none of the previous work can be used to evaluate the system architecture for all NFR criteria at the same time, including reliability, availability, usability, safety, security, etc. Furthermore, the previous works cannot quanti-tatively compare system architectures to make optimal deci-sions during architecture development. In this paper, we aim to propose a method to help engineers deal with quantitative NFR evaluation in the system architecture design phase to drive the software development process rationally. This forms the major motivation of our study in this paper.

**B. ARCHITECTURE MODELING LANGUAGE**

This section describes an overview of ArchiMate, an EA (Enterprise Architecture) modeling language [15], [22],

1. ArchiMate was developed in the Netherlands from July 2002 to December 2004 as an industry-government

VOLUME 8, 2020

collaboration project. In 2008 the ownership and stewardship of ArchiMate were transferred to the Open Group and it is currently managed by the Open Group. It is supported by various tool vendors and consulting rms. ArchiMate provides a clear way to express the construction and behavior of business processes, organizational structures, information ows, software systems, and technology infrastructure based on the concepts of the IEEE 1471 standard. It offers an inte-grated architectural approach that describes and visualizes different architecture domains and their underlying relations and dependencies.

ArchiMate distinguishes itself from other modeling languages such as Uni ed Modeling Language (UML) and Systems Modeling Language (SysML) by its enterprise mod-eling scope. Table 1 describes the similarities and differences between UML, SysML and ArchiMate.

**III. NFR ASSURANCE USING ARCHIMATE**

As explained in the previous section, compared to UML and SysML [24], [25], [35], the main advantage of ArchiMate is that it can be used not only for application architecture

1. layer modeling, but also for business architecture (BA) layer modeling and the technology architecture (TA) layer modeling [15], [19]. Therefore, ArchiMate is suitable for both system visualization and logical exposition [21]. In this section, a modeling method for goal-oriented NFR assurance case development is proposed.

Fig.1 shows the meta-model of the approach. A top goal for assuring NFR is decomposed into sub-goals [14], each sub-goal is realized by a corresponding countermeasure, and then the countermeasures are assessed by evaluating the evidences of the target system. The evidence may be system architecture models, requirement de nition, detailed design speci cation, system test report or system operation record [1], [13].

In ArchiMate, attributes can be used to indicate the sign and the strength of the in uence relationship between two goals or the in uence relationship between a goal and an assessment element. The choice of possible attribute values

is left to the modeler, e.g., {CC, C, 0, , }. Here, CC means a strong positive in uence relationship, C means a positive in uence relationship. 0 means a neutral relation-ship or an unknown situation. means a negative in u-

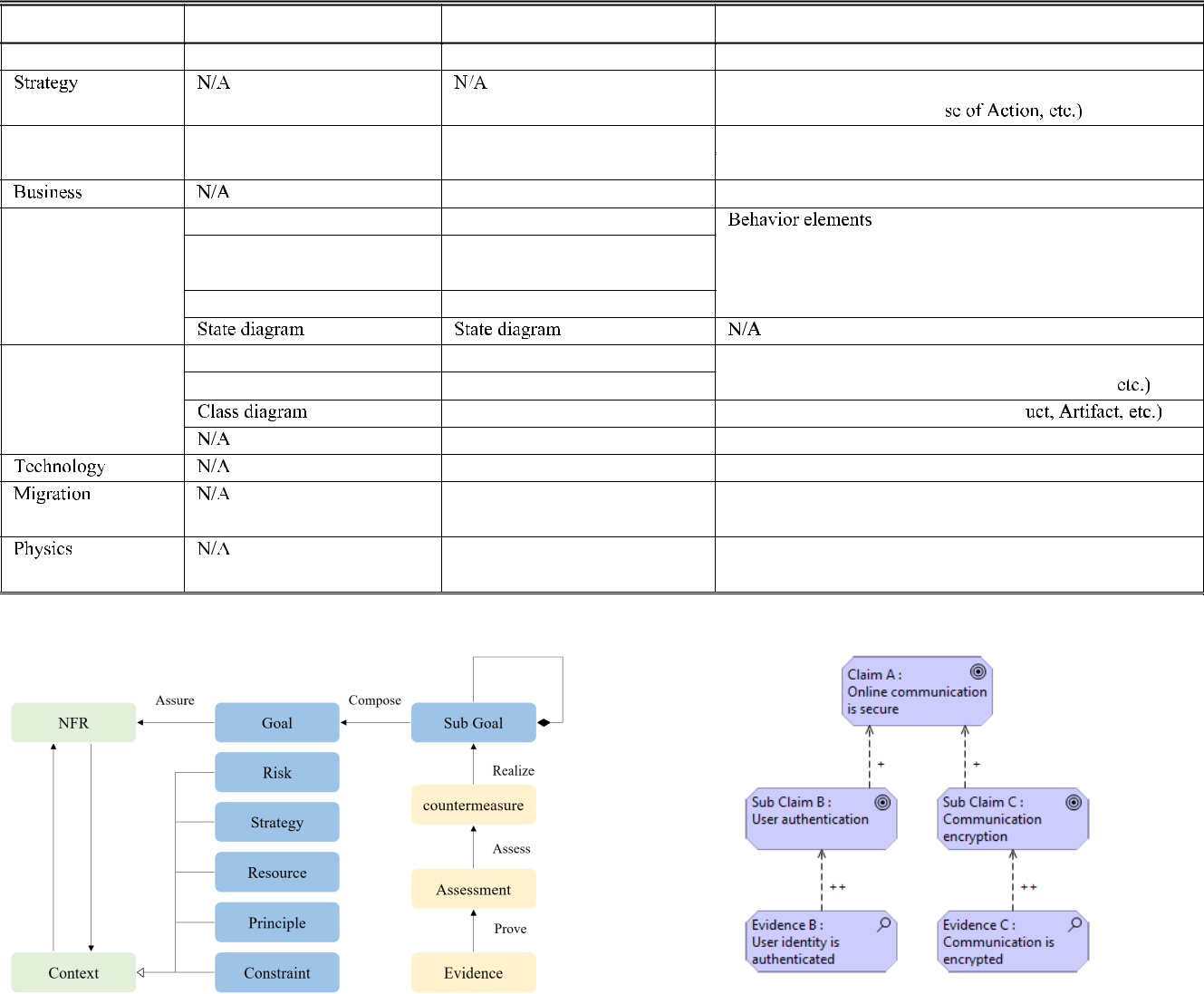
ence relationship, and means a strong negative in uence relationship. Fig.2 shows a simple example of a security assuring case for an online communication system [29]. First, ‘‘System is secure’’ is de ned as top security claim for system security assurance. Then, the top claim is decomposed into 2 sub-claims, ‘‘User authentication’’ and ‘‘Communication encryption’’. In the next step, these sub-claims are assessed according to actual evidences.

Besides qualitative values such as {CC, C, 0, , }, quantitative values also can be used to de ne NFR attributes. This will be suitable for quantitative NFR assurance in large-scale systems development, as the NFR assurance level de nition tends to be more complex. The detailed steps for quantitative NFR evaluation will be introduced in Section IV.

72397

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

**TABLE 1.** Comparison of UML, SysML and ArchiMate.



**FIGURE 2.** Security assurance for the online communication system using

**FIGURE 1.** A meta-model for NFR assurance. ArchiMate.

This approach is appropriate because it follows the descrip-tion rules of ArchiMate framework and thus, there is no need to modify or extend the original syntax of ArchiMate.

**IV. A PROPOSAL FOR QUANTITATIVE SYSTEM ARCHITECTURE NFR EVALUATION**

This section reveals the necessity of the quantitative NFR evaluation method through a survey. Next, the detailed steps for quantitative architecture NFR evaluation are described, and a tool developed by C programming language is also proposed.

**A. AN EVALUATION BY QUESTIONNAIRE**

To have more direct and empirical research, we have designed a questionnaire and conducted an anonymous questionnaire survey with 20 software project managers and domain experts

72398

to evaluate the proposed method. The proposed method was introduced to the participants before conducting the survey. The participants were asked to answer ve multiple-choice questions (see the appendix section) about quantitative NFR evaluation methods. The questions investigated necessity, utility, promotional value, adaptability and persuasiveness of the quantitative NFR evaluation methods, respectively.

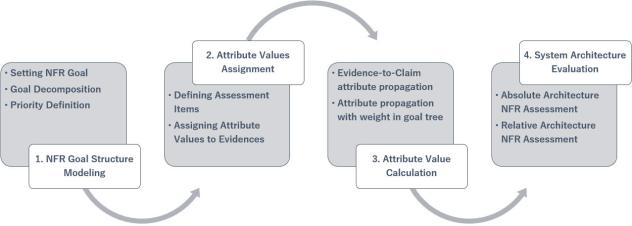
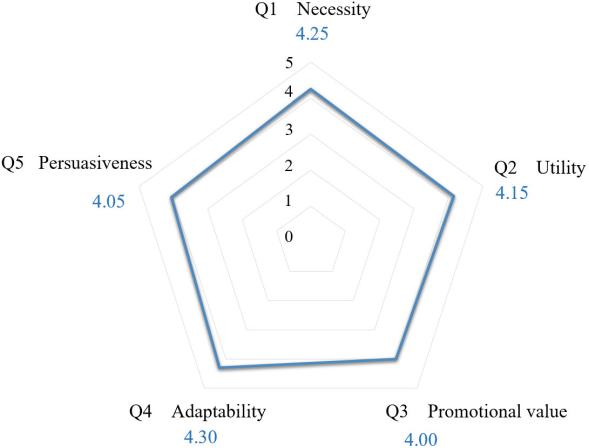
The scores that correspond to the option A E are 5,4,3,2 and 1 points, respectively. Fig.3 shows the results of the questionnaire for the 20 participants. We de ne option A and option B as positive answers, option C as neutral answer and option D and option E as negative answers. The positive answers, neutral answers and negative answers account for 82.0%, 18.0% and 0% of the total for the questions.

Among the ve survey questions, the highest score (4.30 points) is the question about adaptability, that is to say, the

VOLUME 8, 2020

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

**FIGURE 3.** The evaluation results of the questionnaire investigation.



**FIGURE 4.** An overview of the proposed method.

participants were optimistic that the quantitative NFR evalu-ation method can be used in practical software projects. The lowest score (4.00 points) is the question about promotional value, which means that the participants also expressed their hope to learn more about the actual effect of the method before using it. If it works, they will use it positively. In gen-eral, the participants expressed their positive attitude towards the quantitative NFR evaluation method, which indicated that in actual software engineering project, in addition to the tradi-tional qualitative NFR evaluation approaches, it is necessary to develop a quantitative NFR evaluation method.

1. **QUANTITATIVE ARCHITECTURE NFR EVALUATION PROCESS**

The evaluation of the system architecture NFR assurance level can be conducted by applying the following steps in sequence.

**Step 1. NFR Goal Structure Modeling Step 1-**‹*Setting NFR Goal*

Software NFR consists of a range of characteristics, such as reliability, security, safety and usability. However, in a certain system, not all characteristics of NFR are necessarily evaluable. Moreover, some aspects of system NFR may not be meaningful to evaluate. Therefore, it is necessary to sift the appropriate NFR characteristics. For instance, for aircraft control systems and automotive embedded systems, reliabil-ity and safety may be the most worthwhile NFR indicators.

VOLUME 8, 2020

However, for a web service system, usability and security will be the most important NFR indicators to evaluate.

**Step 1-**›*Goal Decomposition*

Each NFR goal is decomposed into several sub-goals, and the decomposition process should be proceeded recursively until the sub-goals cannot be further decomposed. To be clear, the viewpoint for goal decomposition can be system con guration-oriented, system risk-oriented, system process-oriented or other reasonable ideas.

**Step 1-**fi*Priority De nition*

Priority among sub-goals of the same parent NFR goal are decided by analyzing the system requirements and strategies. In this way, a tree structure of software NFR claims can be created.

**Step 2. Attribute Values Assignment Step 2-**‹*De ning Assessment Items*

In order to evaluate the assurance level of each sub-goal in the leaf node of the NFR goal tree, it is necessary to deter-mine the evaluation items corresponding to this sub-goal. For example, the evaluation item corresponding to the Sub Claim B shown in Fig.2 ‘‘User authentication’’ is ‘‘Will user identity be authenticated in the current system architecture design?’’.

Step 2-› *Assigning Attribute Values to Evidences* Attribute values for evidences are de ned by the following

cases. The meanings of the value assignment are de ned as follows. Of course, the value granularity can be freely adjusted according to the system features or system NFR assurance level de nitions.

2 : strongly satis ed

1 : satis ed

0 : neutral

-1 : unsatis ed

-2 : strongly unsatis ed

**Step 3. Attribute Value Calculation**

**Step 3-**‹*Evidence-to-Claim attribute propagation*

By using attribute propagation rule de ned in **De nition 1**, attribute values of the evidences are propagated to the NFR sub-claims in the deepest layer of the goal tree.

*De nition 1*(Evidence-to-Claim attribute propagation rule)

Let a claim C1 is proved by evidence E1. Because it is a one-to-one relationship, the attribute value is propagated between E1 and C1 in the form of direct propagation. That is, if the attribute of the evidence is N , then the attribute of the claim will be N .

**Step 3-**›*Attribute propagation with weight in goal tree*By using attribute propagation rule de ned in De nition 2, the top claim attribute can be calculated by calculating its sub-claims’ attributes recursively, and system architectures can be

quantitatively evaluated by NFR quality claims. *De nition 2*(Claim-to-Claim attribute propagation rule)

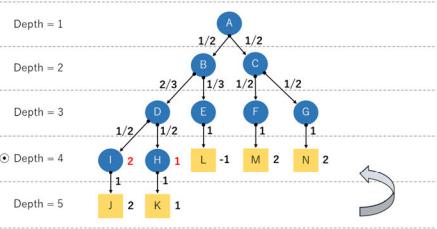
Let a parent claim have been decomposed into k sub-claims. And let P be the attributes of the parent claim. Let S1 , . . . , Sk be the attributes of k sub-claims, respectively. Similarly, let W1 ,. . . , Wk be the weight list of these k sub-claims. Here, the weight list should satisfy the following equation: W1 C W2 C . . . C Wk D 1.

72399

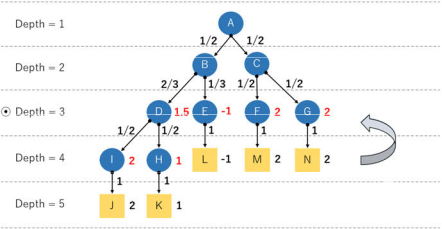
Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate



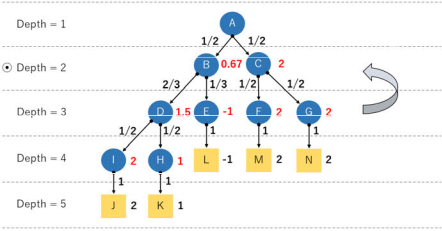
**FIGURE 5.** Step 1.



**FIGURE 6.** Step 2.



**FIGURE 7.** Step 3.



**FIGURE 8.** Step 4.

Then the attribute value P of the parent claim is calculated

P

by the following equation. P D ( iD1,k Si Wi), where

P

iD1,k Wi D1.

To illustrate the procedure and to lend concreteness to the ideas expressed in this section, an example of a calculation process is shown in the gures from Fig.5 to Fig.9.

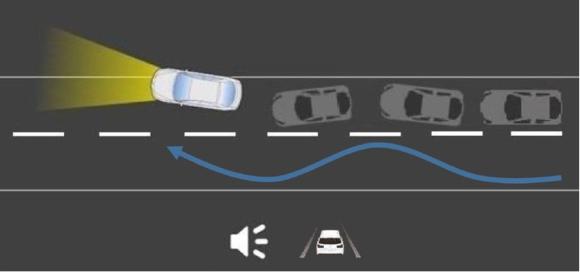
**C. IMPLEMENTATION**

ArchiMate can not only realize system architecture modeling and NFR assurance at the same time, but also its develop-ment environment *Archi* [37] provides a model description language export function. The model description language export les consist of elements.csv, properties.csv and rela-tions.csv. The contents contained in these les are shown in Table 2. In view of this, we realized that a calculation tool

72400

**FIGURE 9.** Step 5.

**TABLE 2.** The contents contained in model description files.



**FIGURE 10.** Vehicle swaying alert system.

can be developed to read model data and calculate attribute value automatically, to save calculating time and improve quantitative calculation accuracy.

First, by reading the relations.csv and elements.csv les, the hierarchical location of each node can be ascertained. Then, starting from the deepest layer, the attribute values are propagated and calculated. To avoid duplication or omission of attribute values calculation, the calculation of the upper layer should be started only after the calculation of the lower layer is completed. The attribute values are calculated and memorized when the calculation of the top claim attribute is completed, calculation results of each node will be output.

When using this tool, the input is elements.csv le, proper-ties.csv le and relations.csv le, which are exported from the NFR assurance model developed by using ArchiMate. The output is the attribute value of each node (including the top claim) in the NFR assurance model. The algorithm of this calculation tool can be described as shown in Algorithm 1.

**V. CASE STUDY OF THE PROPOSED METHOD**

In this section, a case study on a vehicle swaying alert system named VSAS was conducted to verify the effectiveness of the proposed approach. VSAS is designed to warn the driver in the case of vehicle swaying and lane departure due to sleepiness or distraction to ensure safety in driving.

VOLUME 8, 2020

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

**Algorithm 1** Automated Attribute Calculation Algorithm

**Input:** elements.csv, properties.csv, relations.csv

**Output:** Trace of Attribute Values Propagation

* R :D Root
* List L :D Leaf
* **begin** Phase 1 : File reading

4**while** (!EndOfFile)

|  |  |  |
| --- | --- | --- |
| 5 | Struct Relation | Relations File |
| 6 | Struct Property | Properties File |
| 7 | Struct Element | Elements File |

* **end while** 9 **end** Phase 1
  1. **begin** Phase 2 : Hierarchy analysis
  2. **foreach** (leaf in List L)
  3. **if** (leaf.isRelatedby(parent) & parentisNotRoot) **then**
  4. parent.height D leaf.height C 1
  5. parent leaf
  6. **end if**
  7. **end foreach**
  8. **end** Phase 2
  9. H :D R.height
  10. i :D 0
  11. **begin** Phase 3 : Attribute value propagation
  12. **foreach** (leaf in List L)
  13. **if** (leaf.idDElement.id & leaf.idDProperty.id) **then**
  14. leaf.attribute Property.value
  15. **end if**
  16. **end foreach**
  17. **while** (i<DH)
  18. **foreach** (child in List node, where child.height D i)
  19. **if** (child.isCalculatedDfalse &
  20. child.id D Relation.source.id &
  21. parent.id D Relation.target.id) **then**
  22. parent.attribute C D child.attribute child.weight
  23. child.isCalculated D true
  24. **end if**
  25. **end foreach**
  26. iCC
  27. **end while**
  28. **end** Phase 3

**A. VEHICLE SWAYING ALERT SYSTEM (VSAS)**

The reason why this vehicle-mounted embedded system is used to conduct the case study is that it is a safety system that can protect user’s life and property. Therefore, it is worth-while to discuss its non-functional requirements assuring level, such as safety and reliability [17]. As shown in Fig.11, the Vehicle Swaying Alert System (VSAS) consists of meter, sensors, buzzer, and ECUs (Electronic Control Units). Meter and ECUs are connected by CAN (Controller Area Network).

VOLUME 8, 2020

Sensors, buzzer, ECUs and microcomputer are connected by an electric circuit.

**Meter**

Data processing

Real-time information display

**Shift Position Sensor**

Monitor the gear position of vehicle

Transmit gear position data to vehicle AMT ECU

**Steering Angle Sensor**

Measure steering angle

Transmit steering angle data to vehicle body ECU

**Vehicle-mounted Camera**

Detect the relative position of vehicle and lane line

Transmit image data to vehicle image processing ECU

**Vehicle Speed Sensor**

Measure vehicle speed

Transmit vehicle speed data to vehicle body ECU

**Vehicle body ECU**

Convert analog signal of vehicle speed to digital signal Transmit digital signal of vehicle speed to meter microcomputer

**Engine ECU**

Convert analog signal of engine speed to digital signal Convert analog signal of fuel injection quantity to digital signal

Transmit digital signal of engine speed to meter microcomputer

Transmit digital signal of fuel injection quantity to meter microcomputer

**Image Processing ECU**

Raw data processing

Vehicle stability level detection

Transmit vehicle state signal to meter microcomputer

**Buzzer**

Send out warning

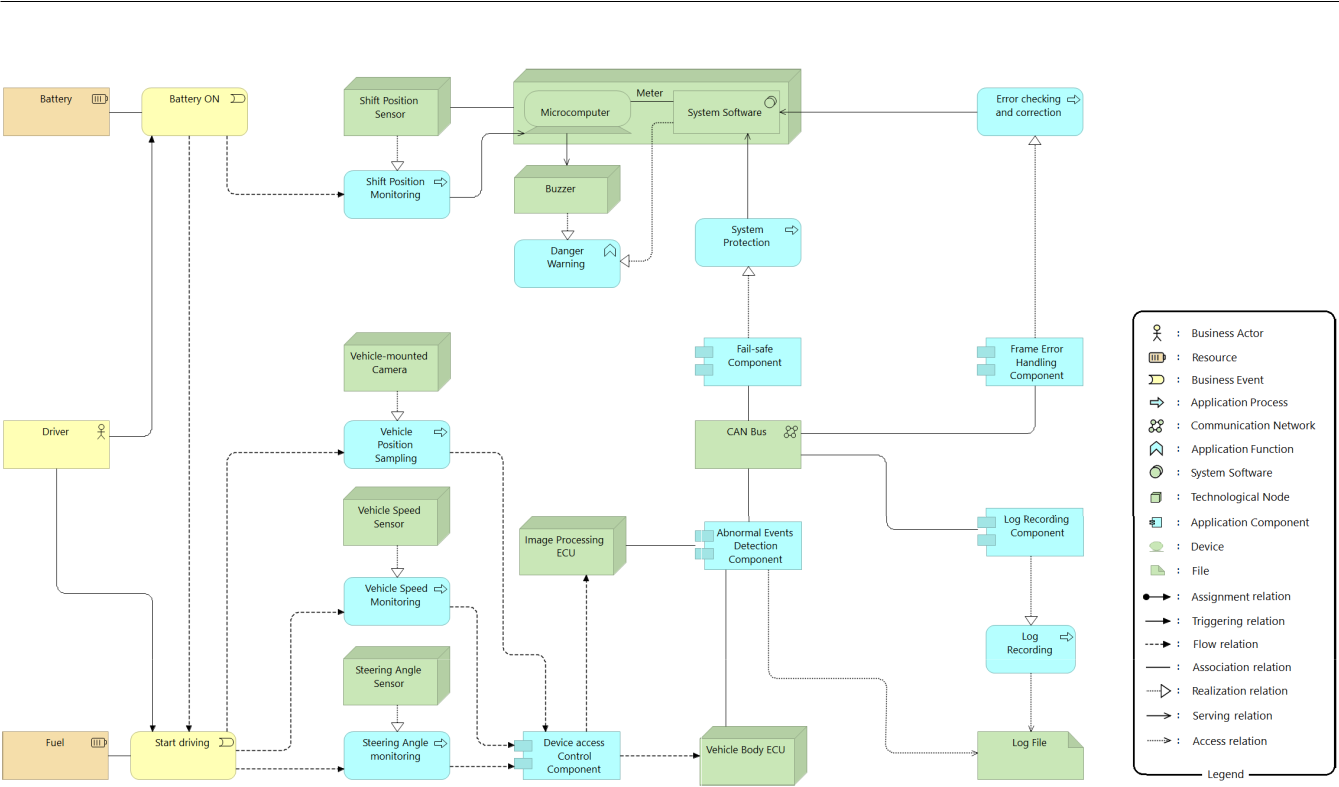
The vehicle embedded system is a driving safety protection system. After starting driving, the microcomputer judges the stability level of the vehicle by detecting the frequency of lane departure and the range and frequency of steering angle change. When the vehicle is detected to be unstable, the speed is greater than the high-speed threshold (Vh D 30km/h), and the shift state is in the Drive mode or Reverse mode, the sys-tem will remind the driver of the vehicle swaying, and advise the driver to stop and take a rest. When the system detects vehicle body swaying and gives an alarm, the TFT display screen of the meter displays a blinking warning indicator of lane departure and the buzzer also issues a warning tone at the same time.

1. **QUANTITATIVE NFR ASSURANCE OF VSAS USING ARCHIMATE**

***Step 1. NFR Goal Structure Modeling.*** The top NFR claim,‘‘VSAS is safe and reliable’’, is decomposed into safety and reliability sub-goals, G2 and G3. The weight de nitions for these sub-goals are 2/3 and 1/3 respectively, as for this

72401

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate



**FIGURE 11.** The system architecture of VSAS in ArchiMate.

vehicle embedded system, safety is more important than reliability. There are two safety goals, G4 and G5, evaluat-ing the following two system safety countermeasures hav-ing equal importance: hardware safety and software safety. There are also three reliability sub-goals, G6, G7 and G8, ‘‘Fault-avoidance’’, ‘‘Fault-tolerance’’ and ‘‘Fault-recovery’’. Fault-avoidance and Fault-recovery are more important than fault-tolerance for VSAS, therefore, the weight de nitions for these sub-goals are 2/5, 1/5 and 2/5, respectively. These sub-goals are further decomposed into sub-goals recursively until they can no longer be decomposed.

***Step 2. Attribute Values Assignment.*** In the next step,to evaluate the assurance level of each sub-goal in the leaf node of the NFR goal tree, we assign an attribute value representing the assurance level of each sub-goal to each evidence node. In this way, an NFR goal tree structure for VSAS can be developed.

***Step 3. Attribute Value Calculation.*** After the develop-ment of the goal tree for NFR evaluation is completed, based on the propagation rules proposed in Section IV, the attribute of the reliability assuring level, availability assuring level, safety assuring level and reliability assuring level for VSAS are calculated as follows.

**Safety**

(0 1 1 2/5C2 1 1 2/5C(-2) 1 1 1/5) 1/2C((2 1 2/3C2 1 1/3) 1 1/2C2 1 1 1/2)

1/2 D 1.2

**Reliability**

((-2) 1 1 1/2C2 1 1 1/2) 2/3C(2 1 1 1/2 C 2 1 1 1/2) 1/3 D 0.67

72402

Therefore, the total value for the top claim ‘‘VSAS is safe and reliable’’ is 0.93, by the weight list of {2/3, 1/3}. Fig.12 shows the quantitative NFR assuring level evaluation for VSAS. It may be well to set ‘‘1: satis ed’’ as the cutoff value. In this case, it means that the current system architec-ture of VSAS is substandard and needs to be improved.

To verify the correctness of the manual calculation results explained above, and show the effectiveness of the attribute calculation tool proposed in Section IV, the calculation tool is used to automate the calculation steps. Then, the auto-matically calculated result is compared with the manually calculated result showed above. The execution result of this attribute calculation tool coincides with the manually calcu-lated value, indicating that the calculation result is correct.

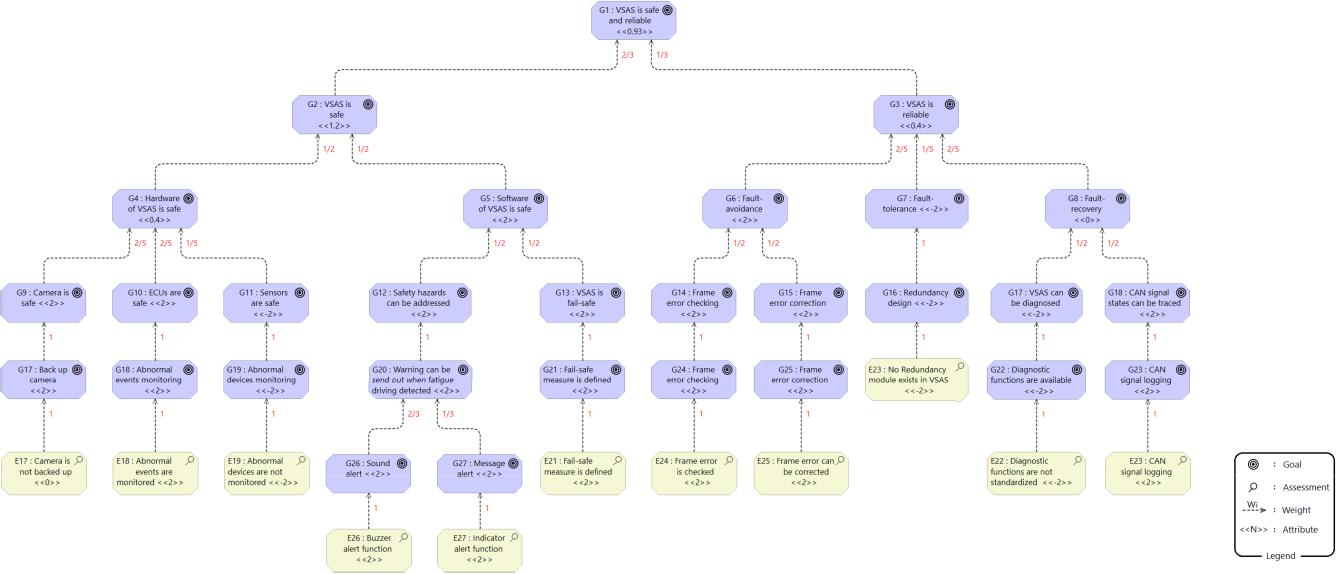
**VI. EXPERIMENT**

To verify the effectiveness of the proposed quantitative NFR evaluation method, the controlled experiment is conducted in this section to compare the proposed method (Group B) with the traditional method (Group A). Fig.13 shows an overview of the experiment.

In the Japanese software industry, there is no uniform method to evaluate system architecture, system architecture evaluation is therefore conducted independently by each soft-ware company. Among these different methods, the most commonly used method is a qualitative method. In this method, architecture design reviewers check the developed architecture models, and during their inspections, they put forward criticisms, opinions or suggestions that will be then recorded in a review result document. The subjects of Group

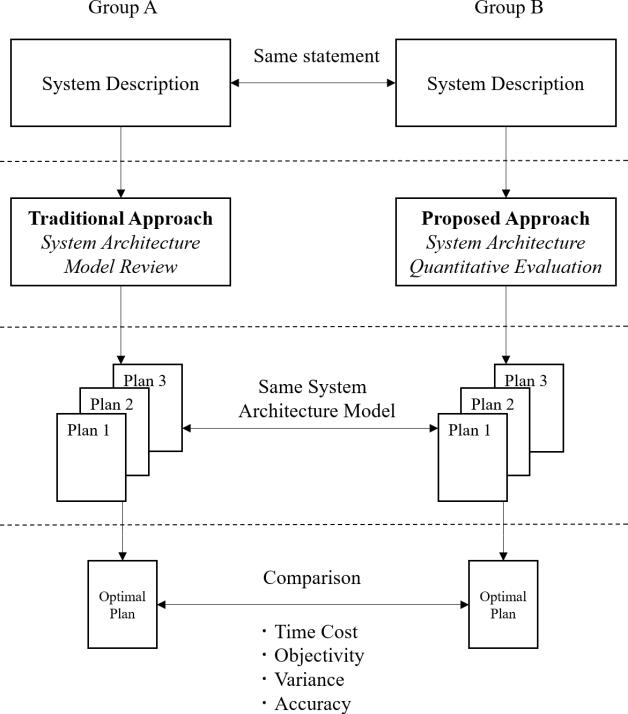
VOLUME 8, 2020

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate



**FIGURE 12.** Quantitative NFR assuring level evaluation for VSAS.

**FIGURE 13.** Overview of the controlled experiment.



A were asked to carry out the experiment with this method. The reason why we use this method as a traditional method for the comparative experiment is that it has already been used by some software companies for decades, and there is no evidence that it has obvious defects.

Since the previous section has illustrated that the proposed method can be used for the evaluation of safety and reliability, for proving the generality of the proposed method for NFR

VOLUME 8, 2020

evaluation, in this controlled experiment, we ask the subjects to use the proposed method to evaluate security, usability and maintainability of the system architecture of VSAS.

**A. EXPERIMENT PROCESS**

The experiment process to evaluate the proposed method is designed as follows.

[Step 1] System introduction

Both Group A and B receive a half-hour explanation for the experimental object system VSAS.

[Step 2] Method seminar

Members of Group A receive an hour-long seminar on the system architecture model review method (the traditional method). Members of Group B receive an hour-long semi-nar on the quantitative system architecture NFR evaluation method (proposed method) and the ArchiMate framework.

[Step 3] Pre-experiment

A pre-experiment is conducted to determine the time limited-term under time pressure condition.

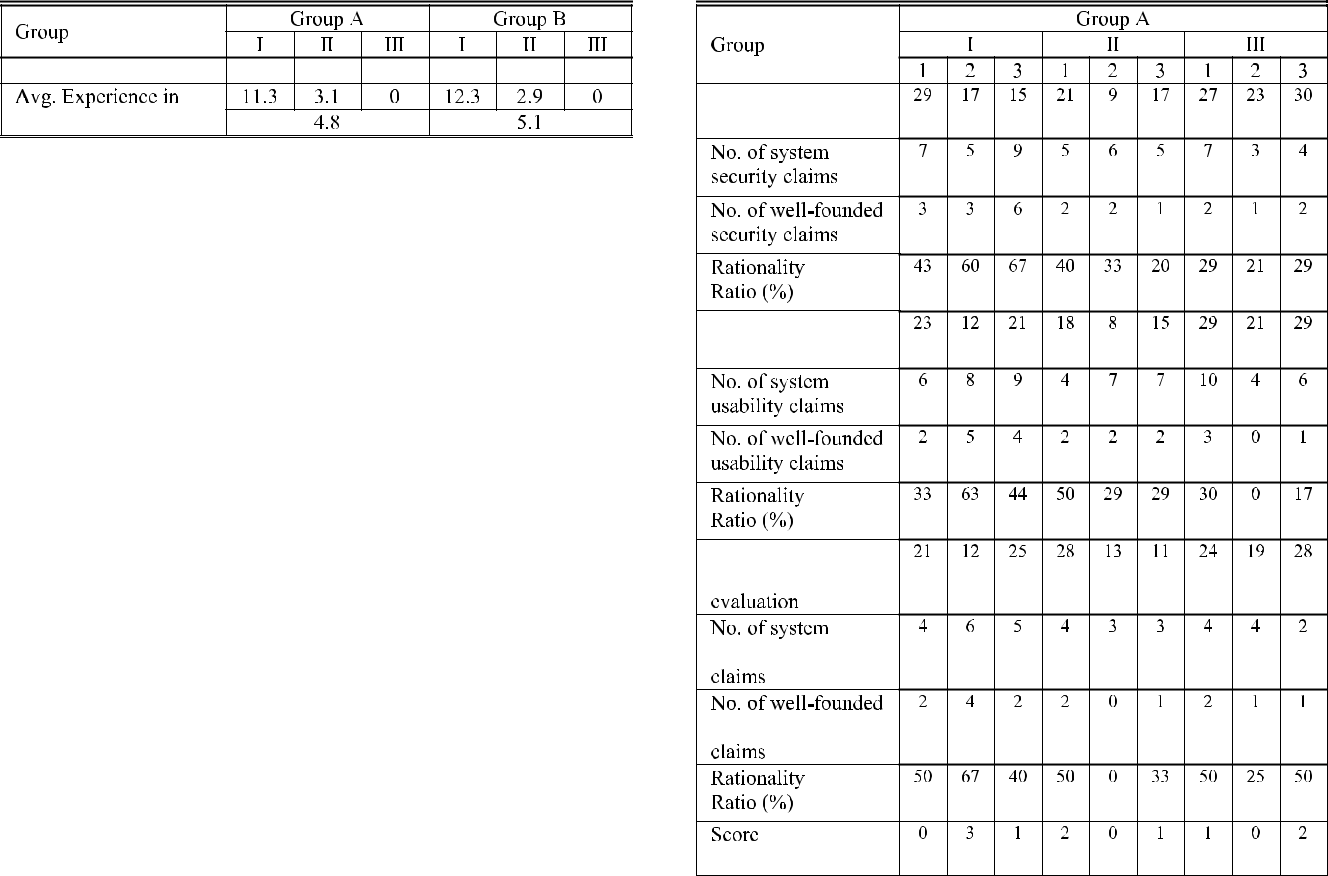
[Step 4] Group experiment

We prepare three system architecture development designs for VSAS: Plan 1, 2 and 3. Each member of Group A uses the traditional method to decide the optimal design for system security (Question I), decide the optimal design for system usability (Question II), and decide the optimal design for system maintainability (Question III) from Plan 1, 2 and 3. Meanwhile, subjects also have to give the corresponding claims and reasons. Each member of Group B uses the pro-posed method to accomplish the same task. The time limit for answering each question is 30 minutes. Each correct answer is worth 1 point. The answers decided by independent system designing experts who are outsiders of this research and experiment, were set as the correct answers for the 3 questions beforehand.

72403

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

**TABLE 3.** Group configuration of the experiment. **TABLE 4.** Experimental results of group A.



**B. GROUP CONFIGURATION**

12 software engineers and 6 university students serve as subjects in the controlled experiment. First, the subjects were divided into three teams, according to the years of experience in system evaluation. Team I: 10 years or more, Team II: 0-5 years, Team III: no work experience. Then, the members of Group A and Group B were chosen by lot from the 3 teams, in other words, the subjects of the three teams were grouped into Group A or Group B with the same probability (50%). Furthermore, the subjects were single-blinded to make the experimental result objective (Team members are randomly assigned, so single-blind is enough). Table 3 describes the detailed group con guration of this experiment. The average numbers of years of system evaluation experience for group A and group B members are 4.8 and 5.1, respectively. T-test shows that there is no signi cant difference in the experience level between group A and group B (at D 0.05 level).

**C. EXPERIMENTAL HYPOTHESIS**

Before implementing the controlled experiment, we formu-late the following 4 hypotheses.

[Hypothesis 1]

*H*0: There is no difference between the traditionalmethod and the proposed method in time cost for NFR evaluation.

*Ha*: The time it takes to evaluate system architecture withthe proposed method is smaller than that with the traditional method.

[Hypothesis 2]

*H*0: There is no difference between the traditional methodand the proposed method in objectivity of work results.

*Ha*: The work results of the proposed method have moreobjectivity than those of the traditional method.

[Hypothesis 3]

*H*0: There is no difference between the variances of theexperimental performance of the proposed method and the traditional method.

*Ha*: The variance of the experimental performance ofthe proposed method is smaller than that of the traditional method.

[Hypothesis 4]

*H*0: There is no difference between the traditional methodand the proposed method in accuracy of NFR evaluation results.

*Ha*: The accuracy of NFR evaluation results for theproposed method is higher than that of the traditional method.

72404

**D. EXPERIMENTAL DATA**



The experimental results for the traditional method and the proposed method are summarized in Table 4 and Table 5, respectively. These tables show the time spent by each subject in evaluating system architecture, the number of system NFR claims made by each subject, the number of well-founded system NFR claims (system NFR claims supported by ver-i able evidences), rationality ratio of these NFR claims and each subject’s score in the experiment. The rationality ratio is de ned as the number of well-founded NFR claims divided by the number of total NFR claims. The number of the claims is counted by the number of independent evaluation viewpoints. For example, if the content of one claim is similar to another, the two claims are only counted once. On the other hand, if one claim contains two different evaluation viewpoints, it will be counted twice. Statistical examination and analysis for the experimental results will be described in the next section.

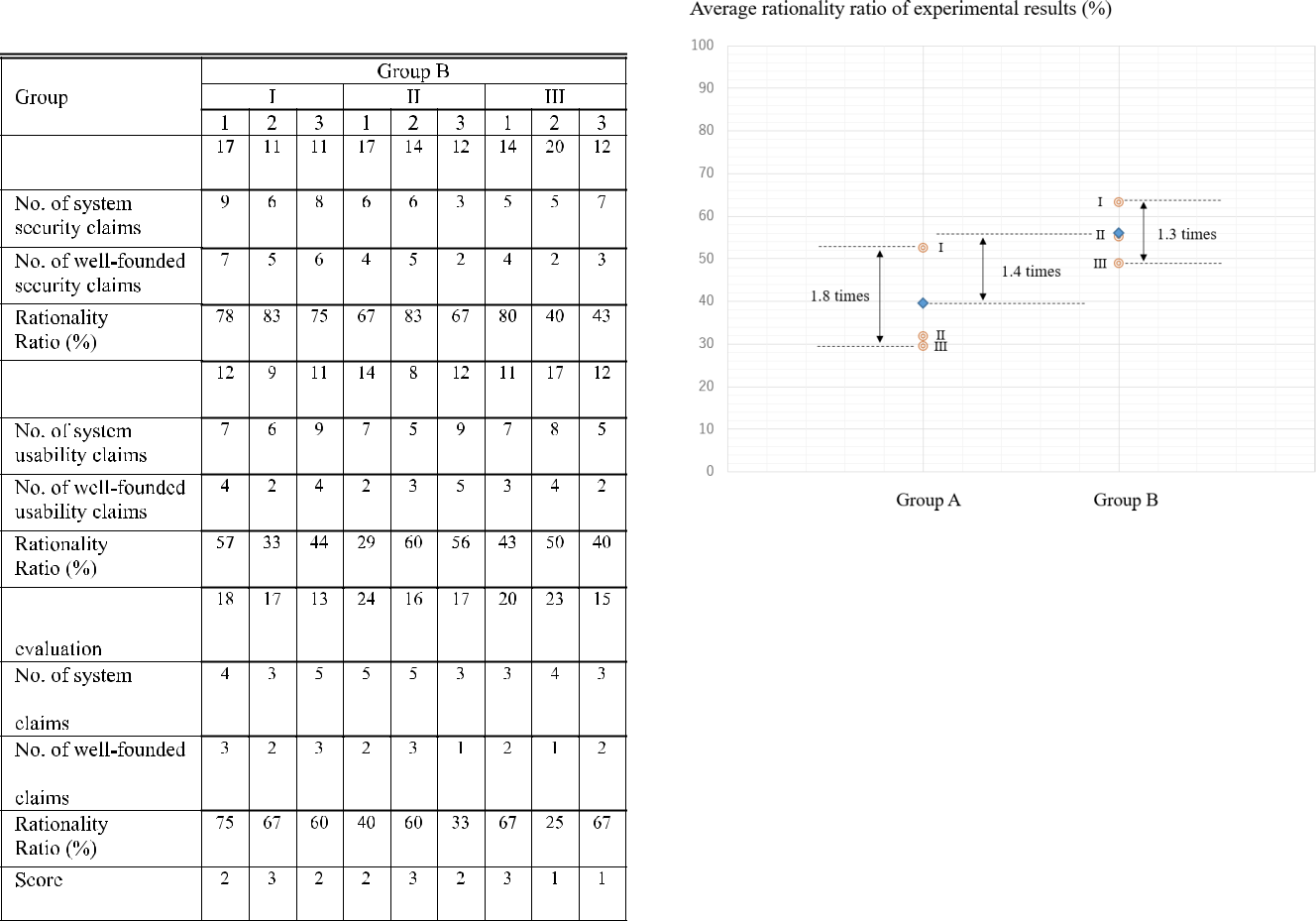
**E. ANALYSIS OF THE EXPERIMENTAL RESULTS**

In this section, the experimental results will be analyzed to test the hypotheses mentioned above. The objective of the experiment is to compare the averages of two independent samples, and the sample size is relatively small (n < 30).

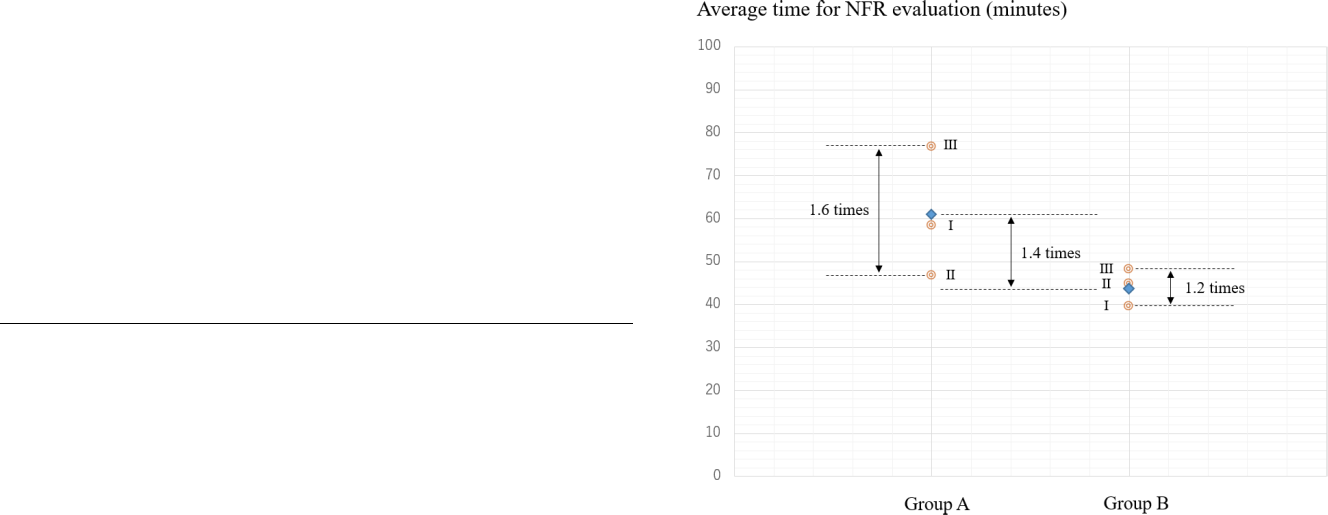
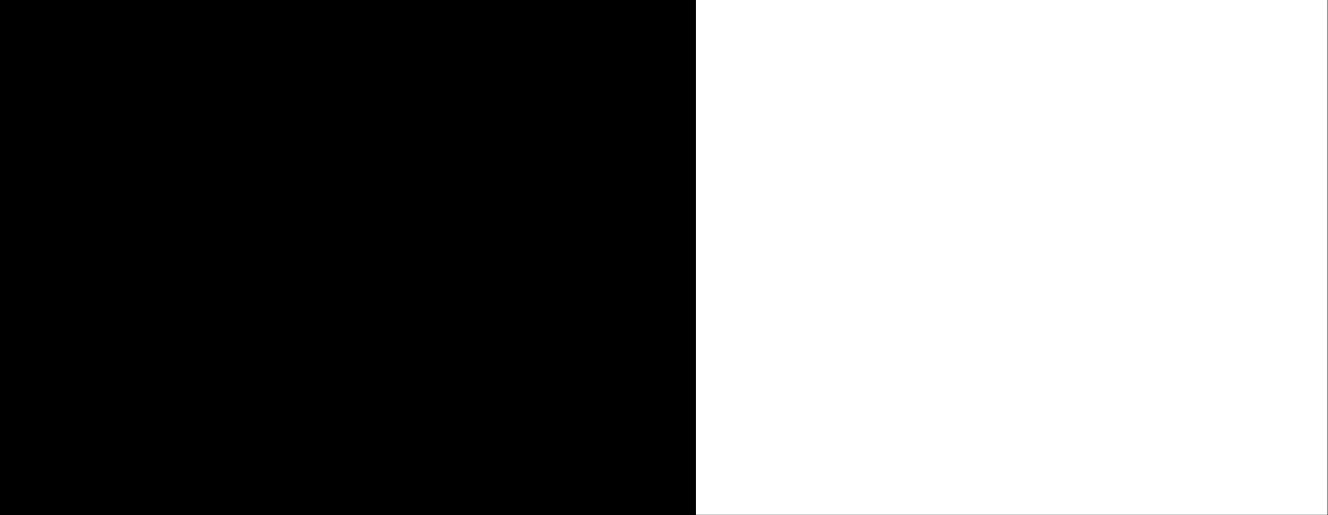
VOLUME 8, 2020

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

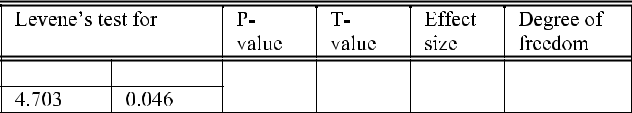
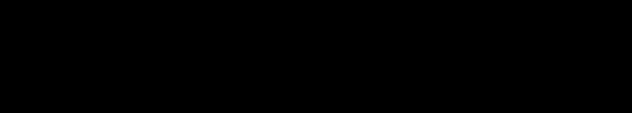
**TABLE 5.** Experimental results of group B.



**FIGURE 14.** Comparison of experimental time.



**TABLE 6.** Statistical testing of time cost for NFR evaluation.



**FIGURE 15.** Comparison of the rationality of experimental results.

Furthermore, the population follows a normal distribution and the population standard deviation is unknown, but sample mean and standard deviation are available. Therefore, we use the independent sample T-test to analyze the experimental results. And we will use F-test to analyze variance.

***Hypothesis 1***

[Observation 1] The mean time for using the proposed method to evaluate system NFRs is smaller than that of the traditional method.

The mean time spent for Group A and B to complete the system NFRs evaluation was 60.6 minutes and 44.1 minutes, respectively. The time cost for Group B to evaluate system NFRs is 72.8% that of Group A. As pp-plot shows that the experimental data follows a normal distribution and levene’s test illustrates that the variances are unequal, we use welch’s T-test to test the data. T-test shows that the difference in mean time for NFR evaluation is signi cant at D 0:05 level. The detailed statistical testing results are shown in Table 6.

VOLUME 8, 2020

According to Observation 1, null hypothesis *H*0 that there is no difference between these two methods in time cost for NFR evaluation should be rejected and thus alternative hypothesis *Ha* can be con rmed.

***Hypothesis 2***

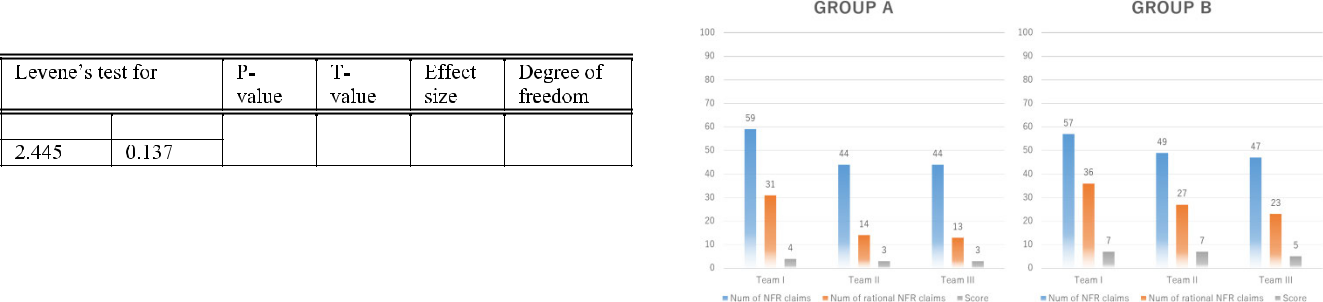
[Observation 2] The mean rationality ratio for using the proposed method to evaluate system NFRs is higher than that of the traditional method.

The subjects of Group A had put forward 147 system NFR claims. Among these claims, 58 (39.5%) were proved to be supported by actual evidences shown in the system architecture design plans. On the other hand, 153 system NFR claims were put forward by the subjects of Group B. Among these claims, 86 (56.2%) claims were proved to be supported by actual evidences. The mean rationality ratio for using the proposed method to evaluate system NFRs is 1.4 times that of Group A. PP-plot implies that the experi-mental data follows a normal distribution and levene’s test

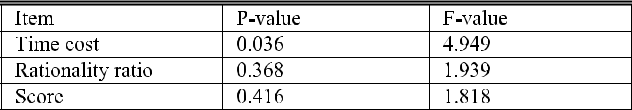
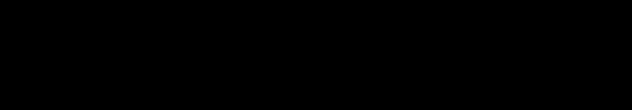
72405

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

**TABLE 7.** Statistical testing of rationality ratio for NFR evaluation.



**TABLE 8.** Statistical testing for variance comparison.



**FIGURE 16.** Comparison of the accuracy of experimental results.

**TABLE 9.** Statistical testing of accuracy for nfr evaluation.

illustrates that the variances are equal. T-test shows that the difference of mean rationality ratio is signi cant at D 0:05 level. The detailed statistical testing results are shown in Table 7.

According to Observation 2, null hypothesis *H*0 that there is no difference between these two methods in objectivity should be rejected and alternative hypothesis *Ha* is con rmed.

***Hypothesis 3***

[Observation 3] The variance of the time cost for evaluating NFRs of the proposed method is smaller than that of the traditional method.

[Observation 4] There is no signi cant difference between the variances of the rationality ratio for using the proposed method and the traditional method to evaluate system NFRs.

[Observation 5] There is no signi cant difference between the variances of the mean score for using the proposed method and the traditional method to evaluate system NFRs.

F-test shows that the variance of the time cost for evaluat-ing NFRs of the proposed method is smaller than that of the traditional method, the signi cant level is D 0:05. How-ever, F-test also shows that there is no signi cant difference between the variances of the rationality ratio and the score for using the proposed method and the traditional method to evaluate system NFRs. The detailed statistical testing results are shown in Table 6.

According to Observation 3, 4 and 5, null hypothesis *H*0 that there is no difference between the variances of the exper-imental performance is not rejected.

***Hypothesis 4***

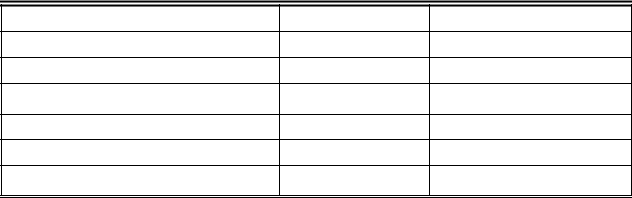
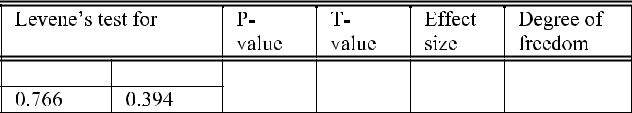
[Observation 6] The mean score for evaluating system NFR with the proposed method is higher than that of the traditional method.

The mean scores of Group A and B in the controlled exper-iment were 1.11 points and 2.11 points, respectively. The mean score for Group B to evaluate system NFRs is 1.9 times that of Group A. PP-plot implies that the experimental data follows a normal distribution and levene’s test illustrates that the variances are equal. T-test shows that the difference of mean score is signi cant at D 0:05 level. The detailed statistical testing results are shown in Table 9.

According to Observation 6, null hypothesis *H*0that there is no difference between these two methods in accuracy

72406

**TABLE 10.** Statistical testing of proposed method and traditional method.



should be rejected and thus alternative hypothesis *Ha* can be con rmed.

**VII. DISCUSSION**

**A. EFFECTIVENESS OF THE PROPOSED METHOD**

To verify the effectiveness of our method, we applied the proposed method and algorithm to evaluate the NFR assur-ing level of a vehicle embedded system. As shown in the case study above, the proposed method is useful to evaluate the system architecture for NFR assurance in the embedded application domain. And it is proved that the method can be used for identifying, decomposing and quantitatively evalu-ating NFRs.

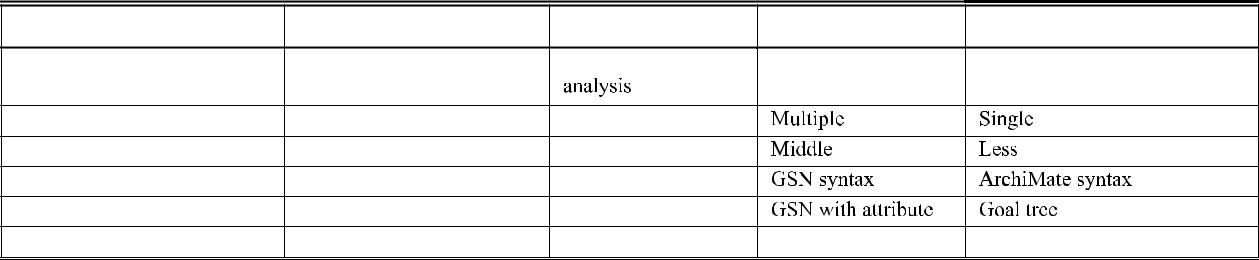
In the previous section, we have conducted a controlled experiment to compare our method with the traditional archi-tecture model review method. Table 10 shows the comparison of the experimental results for our method and the traditional method. The comparison result implies that our method has achieved better performance in terms of ef ciency, effective-ness and objectivity. In addition, the variance of experimental data using the proposed method is smaller than that of the traditional approach, although it is not signi cant at D 0:05 level.

Table 11 lists the comparison of the proposed method and the related work. Using the existing approaches for NFR assessment, engineers have to use UML or SysML for system modeling, and then use their NFR evaluation methods to

VOLUME 8, 2020

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

**TABLE 11.** Comparison of the proposed method and existing quantitative NFR assurance approaches.



conduct NFR modeling. If engineers use our method, the two separate processes can be just integrated with ArchiMate. Moreover, the grammar of our method is relatively con-cise. Arrow symbols representing the interaction between NFR claims are used to connect parent-goals and sub-goals. Attribute values are added to the goals, and weights are added to the relations to reduce the nodes for discussions on goal decomposition, which will reduce the complex-ity of the model and improve the readability. The existing approaches use expressions or tabular calculation to quanti-tatively assess the NFR grade, which may lead to calculation errors while taking more time. In the related work, calcula-tion processes are not traceable. We proposed a calculation tool for attribute values calculation to solve these problems. Besides, the proposed method models NFRs with Archi-Mate, which brings af nity to business scenario and system architecture.

**B. CONTEXTUALIZATION OF THE PROPOSED METHOD**

At present, there are 3 contexts of the proposed method that we have envisaged and validated.

**Absolute Architecture NFR Assessment**

First, the proposed method can be used to quantitatively evaluate system NFR assurance level to decide if system NFRs are satis ed or not, which had been discussed in the case study section. In the case study, the system NFR evalua-tion result is smaller than the system NFR assurance criterion, which implies that the system architecture shown in Fig.11 is substandard in terms of system safety and reliability, and needs to be improved.

**Relative Architecture NFR Assessment**

In addition, the proposed method can be used to quanti-tatively evaluate multiple system architectures to decide the optimal architecture design plan for NFR assurance, which was con rmed in the experiment section.

**Automatic reasoning on NFR assurance models**

The method also can be used to support system devel-opment decision making for NFR assurance. In particular, it can be integrated with greedy algorithms to solve the CSP (Constraint Satisfaction Problem) of how to obtain optimal NFR assuring performance by selecting appropriate NFR countermeasures under constraint conditions. We have conducted an experiment on this suggestion by using a

VOLUME 8, 2020

modi ed knapsack algorithm, the results will be reported elsewhere.

**C. THREATS TO VALIDITY**

In the proposed method, the attribute values of the NFR claims can be calculated, but the proportion of the sub-claims in the deepest layer among the whole claims is not clear. Therefore, it is dif cult to clarify which NFR goals or func-tions are important for the system. It is necessary to devise an approach to calculate the ratio at which each NFR goal occupies the total. The tree structure with weights and depth corresponds to the decomposition structure of system NFR goals and the priority and evaluation result of each sub-goal. The priority, assurance level, and attribute value of each node need to be validated. If engineers using this method do not understand the priority of each sub-goal correctly, or the result of evaluating the assurance level of the underlying goal is not accurate enough, the overall evaluation result of the system NFR will not be suf cient and biased. The weight assignment for the NFR claims in the NFR goal tree depends on the engineer’s development experience and understanding level of the target system, so the weight assignment may be subjective. Therefore, software engineer needs to have a high professional level. In addition, a software development team can use multiple experienced engineers to collaborate on NFR evaluation at the same time to obtain a relatively objective result. In addition, a Likert scale (1-5) in terms of importance might be easier for engineers to decide instead of continuous values. A better approach to propose the weights would be set up for future work of the study.

In this paper, only a case study was conducted, it is signi cant to apply this method to more software sys-tems to con rm the effectiveness of the proposed method. Additionally, although we have conducted a questionnaire survey to con rm the feasibility of the proposed method, it is important to conduct experiments in software compa-nies to obtain empirical evidences to further prove that the method can be used in the software industry. In Section VI, a controlled experiment was conducted and experimental results were analyzed, these results do not fully demonstrate the superiority of the proposed method. It is necessary to increase the number of subjects and do more experiments to improve the objectivity and comprehensiveness of the evaluation. Although we have used random grouping and

72407

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

single-blind evaluation strategy in the experiment, and controlled the subjects’ average experience in architecture evaluation, understanding level of the target system, and experience with ArchiMate to ensure the reliability of the experiment, some other factors such as subjects’ experi-ence with related studies (e.g., FTA, NFR Framework and GSN) were not controlled in the experiment. It is necessary to strictly control all the experimental variables to further improve the reliability of the experiment. Furthermore, repli-cation is required to measure the stability or reliability of the experiment.

The proposed method was implemented in an easy to follow case study, where safety and reliability are clear NFRs. However, it is not clear how this could be introduced in systems with a high number of interdependencies where the goals are not easily mapped into a tree or to parts of a system. Moreover, lacking a strong mathematic background is also a weakness of the proposed method. Without a formal algebraic structure, the method can only model NFRs for relatively small products. It is meaningful to integrate math-ematical approaches with the proposed method. Therefore, when conducting NFR evaluation for large-scale complex systems, we recommend that engineers use this method to evaluate the key parts of the system rather than the whole system.

**D. CONCLUSION AND FUTURE WORK**

In this paper, we proposed a quantitative architecture NFR evaluation method. Our method utilizes a goal decomposition tree to clarify the NFR properties that the target system should possess. The attribute value of each sub-goal is determined by assessing whether this goal can be achieved in the sys-tem architecture. The overall NFR satisfaction of the system is quantitatively evaluated by attribute value propagation. Through the case study in Section V and the controlled experiment in Section VI, we con rmed that our method can be used for architecture safety, security, reliability usability and maintainability evaluation, and it is obvious that the proposed method can be used for the evaluation of other NFR properties such as extendibility, performance and portability, based on its goal-oriented process. The experimental results con rmed our hypotheses and showed that our method is effective and ef cient. Although the proposed method has not been popularized in the software industry, the results of the questionnaire survey show that the method is con-tributive in the domain and it is anticipated. The proposed method is expected to be used at the early stage of software development projects for system NFR development, such as requirements analysis, system architecture design and system modeling.

However, it should be noted that there is still much room for the perfection of this research due to the limitation men-tioned in the previous section. Solving the de ciencies of the proposed method will become the future work of our research in this paper.

72408

**APPENDIX**

**QUESTIONS FOR QUESTIONNAIRE SURVEY**

[Q1] Do you think that the evaluation of system archi-tecture design needs a quantitative evaluation method, rather than a qualitative one?

1. Very necessary
2. Necessary
3. A little
4. Not really
5. Not necessary at all

[Q2] Do you think that the quantitative NFR assurance level evaluation method is useful for the evaluation of NFRs?

1. Very useful
2. Useful
3. A little
4. Not really
5. Not useful at all

[Q3] To what extent do you want to use the NFR assurance level evaluation method for software development projects?

1. I want very much to use it
2. I want to use it
3. After effectively improved, I may try to use it
4. Not really
5. I do not want to use it at all

[Q4] Do you think that the conventional system development process can be changed to deploy the quantitative NFR eval-uation method to software development projects?

1. It can be changed easily
2. It can be changed
3. It is dif cult to change
4. It cannot be changed
5. It cannot be changed absolutely

[Q5] Do you think that customers can agree to use the NFR evaluation method in software development processes?

1. It can be agreed
2. If the reason for using the proposed method is explained, it can be agreed
3. If the effectiveness of the proposed method is con-rmed, it can be agreed
4. It is dif cult to agree
5. It cannot be agreed

**REFERENCES**

1. I. Sommerville, *Software Engineering*, 10th ed. London, U.K.: Pearson, 2015.
2. A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, ‘‘Basic con-cepts and taxonomy of dependable and secure computing,’’ *IEEE Trans.* *Dependable Secure Comput.*, vol. 1, no. 1, pp. 11 33, Jan. 2004.
3. L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos, *Non-Functional Require-ments In Software Engineering*. Norwell, MA, USA: Kluwer, 2000.
4. E. Ruijters and M. Stoelinga, ‘‘Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools,’’ *Comput. Sci. Rev.*, vols. 15 16, pp. 29 62, Feb. 2015.
5. E. Ruijters and R. Weaver, ‘‘The goal structuring notation A Safety argument notation,’’ in *Proc. Dependable Syst. Netw. Workshop Assurance* *Cases*, Jul. 2004, p. 6.
6. S. Yamamoto, H. Kaiya, K. Cox, and S. Bleistein, ‘‘Goal oriented requirements engineering: Trends and issues,’’ *IEICE Trans. Inf. Syst.*, vols. E89 D, no. 11, pp. 2701 2711, Nov. 2006.

VOLUME 8, 2020

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

1. A. Kokune, M. Mizuno, K. Kadoya, and S. Yamamoto, ‘‘FBCM: Strategy modeling method for the validation of software requirements,’’ *J. Syst.* *Softw.*, vol. 80, no. 3, pp. 314 327, Mar. 2007.
2. S. Yamamoto, ‘‘An approach for evaluating softgoals using weight,’’ in *Proc. Inf. Commun. Technol.-Eurasia Conf.*, Daejeon, South Korea, 2015,
   1. 203 212.
3. H. Kaiya, H. Horai, and M. Saeki, ‘‘AGORA: Attributed goal-oriented requirements analysis method,’’ in *Proc. IEEE Joint Int. Conf. Require-ments Eng.*, Essen, Germany, Sep. 2002, pp. 13 22.
4. S. Yamamoto, ‘‘Assuring security through attribute GSN,’’ in *Proc. 5th Int.* *Conf. IT Converg. Secur. (ICITCS)*, Kuala Lumpur, Malaysia, Aug. 2015,
   1. 1 5.
5. S. Yamamoto, Q. Zhi, and S. Morisaki, ‘‘A composite dependability for enterprise archtecture,’’ in *Proc. Int. Conf. Knowl. Based Intell. Inf. Eng.* *Syst.*, Belgrade, Serbia, Jan. 2018, pp. 1130 1137.
6. S. Yamamoto and Y. Matsuno, ‘‘An evaluation of argument patterns to reduce pitfalls of applying assurance case,’’ in *Proc. 1st Int. Workshop* *Assurance Cases Softw.-Intensive Syst. (ASSURE)*, San Francisco, CA,USA, May 2013, pp. 12 17.
7. L. Chen, M. Ali Babar, and B. Nuseibeh, ‘‘Characterizing architec-turally signi cant requirements,’’ *IEEE Softw.*, vol. 30, no. 2, pp. 38 45, Mar. 2013.
8. Y. Yu, J. Mylopolos, E. Yu, L. Linda, and L. Liu, ‘‘Software refactoring guided by multiple soft-goals,’’ in *Proc. Working Conf. Reverse Eng.*, Victoria, BC, Canada, 2003, pp. 1 12.
9. *The Open Group, ArchiMate 3.0 Speci cation*, Van Haren Publishing, ’s-Hertogenbosch, The Netherlands, 2016.
10. E. Grandry, C. Feltus, and E. Dubois, ‘‘Conceptual integration of enterprise architecture management and security risk management,’’ in *Proc. 17th* *IEEE Int. Enterprise Distrib. Object Comput. Conf. Workshops*, Vancouver,BC, Canada, Sep. 2013, pp. 114 123.
11. W. Abbass, A. Baina, and M. Bellafkih, ‘‘Improvement of information system security risk management,’’ in *Proc. 4th IEEE Int. Colloq. Inf. Sci.* *Technol. (CiSt)*, Oct. 2016, pp. 182 187.
12. S. Yamamoto and S. Morisaki, ‘‘A case study on architecture quality assurance service using o-da,’’ in *Proc. Int. Conf. ENTERprise Inf. Syst.*, Porto, Portugal, Oct. 2016, pp. 189 193.
13. S. Yamamoto and N. Kobayashi, ‘‘Mobile security assurance through ArchiMate,’’ in *Proc. Int. Symp. Mobile Internet Secur.*, Taichung, Taiwan, 2016, pp. 10 20.
14. N. Mayer and C. Feltus, ‘‘Evaluation of the risk and security overlay of archimate to model information system security risks,’’ in *Proc. IEEE 21st* *Int. Enterprise Distrib. Object Comput. Workshop (EDOCW)*, Quebec City,QC, USA, Oct. 2017, pp. 106 116.
15. S. Yamamoto, ‘‘An approach to assure dependability through archimate,’’ in *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, 2015, pp. 50 61.
16. G. Wierda, *Mastering ArchiMate A Serious Introduction to the Archi-Mate Enterprise Architecture Modeling Language*, vol. 2, Amsterdam, TheNetherlands: R&a, 2014.
17. M. Lankhorst, *Enterprise Architecture at Work Modeling Communica-tion and Analysis*. Berlin, Germany: Springer, 2013.
18. J. Jacobson, I. Booch, and G. Rumbaugh, *Uni ed Modeling Language* *Reference Manual*. Boston, MA, USA: Addison-Wesley Professional,2004.
19. S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML :* *The Systems Modeling Language*, 3rd ed. Burlington, MA, USA: MorganKaufmann, 2014.
20. K. Tim and R. Weaver, ‘‘The goal structuring notation A safety argument notation,’’ in *Proc. Dependable Syst. Netw. Workshop Assurance Cases*, 2004, p. 6.
21. R. Fischer, S. Aier, and R. Winter, ‘‘A Federated Approach to Enterprise Architecture Model Maintenance,’’ *Enterprise Model. Inf. Syst. Archit.*, vol. 2, no. 2, pp. 14 22, Nov. 2007.
22. B. Len, C. Paul, and K. Rick, *Software Architecture in Practice*, 2nd ed. Boston, MA, USA: Addison-Wesley Professional, 2003.
23. S. Yamamoto, T. Kaneko, and H. Tanaka, ‘‘A proposal on security case based on common criteria,’’ in *Proc. Inf. Communicatiaon Technol. Int.* *Conf.*, Yogyakarta, Indonesia, Mar. 2013, pp. 331 336.
24. P. Bishop and R. Bloom eld, ‘‘A Methodology for Safety Case Devel-opment,’’ *Ind. Perspect. Safety-Crit. Syst.*, vol. 20, no. 1, pp. 194 203, 1998.
25. H. Fujita, Y. Matsuno, T. Hanawa, M. Sato, S. Kato, and Y. Ishikawa, ‘‘Tools for dependability benchmarking with simulation and assur-ance,’’ in *Proc. Int. Conf. Dependable Syst. Netw.*, Boston, MA, USA, 2012, pp. 1 8.

VOLUME 8, 2020

1. N. Subramanian and J. Zalewski, ‘‘Quantitative assessment of safety and security of system architectures for cyberphysical systems using the NFR approach,’’ *IEEE Syst. J.*, vol. 10, no. 2, pp. 397 409, Jun. 2016.
2. N. Kobayashi, S. Morisaki, N. Astumi, and S. yamamoto, ‘‘Quantitative Non Functional Requirements evaluation using softgoal weight,’’ *J. Inter-net Services Inf. Secur.*, vol. 6, no. 1, pp. 37 46, 2016.
3. S. Saito and S. Yamamoto, ‘‘The incremental goal evolution process methodology,’’ in *Proc. Int. Conf. Adv. Inf. Syst. Eng. Trusted Inf. Syst.*, Jun. 2006. pp. 1 9.
4. S. Sarma, S. Udupa, V. K. Agarwal, and K. Parameshwaran, ‘‘Software system architecture modeling using UML,’’ in *Proc. 58th Int. Astron.* *Congr.*, 2007, pp. 1 15.
5. D.-C. Committee, *D-Case Editor A Typed Assurance Case Editor*. Accessed: Feb. 4, 2019. [Online]. Available: http://www.dcase.jp/p/ eeditor.html
6. P. Beauvoir. *Archi Open Source ArchiMate Modelling*. Accessed: May 1,

2018. [Online]. Available: https://www.archimatetool.com/

1. A. Silva, P. R. Pinheiro, A. Albuquerque, and J. Barroso, ‘‘Evaluation of an approach to de ne elicitation guides of non-functional requirements,’’ *IET Softw.*, vol. 11, no. 5, pp. 221 228, Oct. 2017.
2. T. Kuhn, P. Antonino, and A. Morgenstern, ‘‘Automated scenario-based evaluation of embedded software and system architectures,’’ in *Proc. Sci.* *Inf. Conf.*, 2018, pp. 452 469.
3. S. Sumesh, A. Krishna, and C. M. Subramanian, ‘‘Game theory-based reasoning of opposing non-functional requirements using inter-actor dependencies,’’ *Comput. J.*, vol. 62, no. 11, pp. 1557 1583, Nov. 2019.
4. M. A. Kohl, K. Baum, M. Langer, D. Oster, T. Speith, and D. Bohlender, ‘‘Explainability as a non-functional requirement,’’ in *Proc. IEEE 27th Int.* *Requirements Eng. Conf. (RE)*, Sep. 2019, pp. 363 368.
5. B. DeVries and B. Cheng, ‘‘Goal-based modeling and analysis of non-functional requirements,’’ in *Proc. ACM/IEEE 22nd Int. Conf. Model* *Driven Eng. Lang. Syst. (MODELS)*, Munich, Germany, Sep. 2019,pp. 261 271.
6. A. Ahmad, C. Feng, K. Li, S. M. Asim, and T. Sun, ‘‘Toward empirically investigating non-functional requirements of iOS developers on stack over-ow,’’ *IEEE Access*, vol. 7, pp. 61145 61169, 2019.

ZHENGSHU ZHOU received the master’s degree in software engineering from Nagoya University, Japan, where he is currently pursuing the Ph.D. degree with the Graduate School of Informatics. He was a Software Engineer with the Japanese Software Industry. His research interests include software engineering, information security, and system dependability assurance.



QIANG ZHI received the master’s degree in information science from Hiroshima University, in 2016. He is currently pursuing the Ph.D. degree with the Graduate School of Informatics, Nagoya University, Japan. He was a Software Engineer with the Japanese Software Industry. His main research interests include software engineering and information security.



72409

Z. Zhou *et al.*: Evaluation of Quantitative NFRs Assurance Using ArchiMate

SHUJI MORISAKI received the D.E. degree in information science from the Nara Institute of Science and Technology, Japan, in 2001. He was a Software Engineer with the Japanese Software Industry. He is currently an Associate Professor with the Graduate School of Informatics, Nagoya University, Japan. His research interests include empirical software engineering and software quality.



SHUICHIRO YAMAMOTO received the Dr.Eng. degree from Nagoya University, in 2000. He joined the Nippon Telegraph and Telephone Public Cor-poration (NTT), in 1979. He engaged with the development of CASE tools, network-based smart card environments, and distributed application development platforms. He was with NTT DATA, in 2002. He was a rst Fellow with NTT DATA Research and Development Headquarters, in 2007. He moved to Nagoya University, as a Professor,



in 2009, where he is currently a Professor with the Graduate School of Informatics.



72410 VOLUME 8, 2020