International Journal of Service and Knowledge Management
International Institute of Applied Informatics
2021, Vol. 5, No. 1, 1 – 14

An Experience to Design O-DA Knowledge

Shuichiro Yamamoto *, Junkyo Fujieda †,
Akira Fukushima †, Kiichi Kawano †

Abstract

Although the dependability of architecture is widely expected in the digital age, it is not clear what content shall the courseware to learn the dependability has. In this paper, we propose the knowledge design approach on the open dependability to certify enterprise architects who have capability to develop dependable architecture. The knowledge design approach is based on the open standard of The Open Group named O-DA that means Open Dependability through Assuredness. The knowledge configuration and development process of the knowledge design approach have described.

Keywords: Open Dependability, Assurance Case, Quality Assurance, O-DA, ArchiMate.

1 Introduction

System failures after service deployment not only seriously affects society and service customers, but also services. It has a significant impact on the company's business opportunities and the business environment. Continuation of optimal service to the service provider. Minimization of damage in case of system outage, early restoration of service, prevention of recurrence due to similar factors, accountability and Responsibility for management result is required.

However, today's systems are increasingly complex and diverse, including clouds, IoT, big data, wireless, cognitive (AI), SNS, etc., and demands for safety and security are also increasing. For this reason, maintaining social and business Dependability (reliability) at the time of system failure has become difficult.

"Open Group (The Open Group) released in 2013" Safety and Reliability Verification Standard The Open Dependability through Assuredness ™ Standard (O-DA) "is an international standard from Japan, making decisions on problem solving. It was an open standard for the stakeholders to manage risks.

* Nagoya University, Nagoya, Japan
† Research Environment for Grobal Information Society, Inc., Tokyo, Japan

In Japan, more than 1,000 certified architects of TOGAF which is the standard of Enterprise Architecture. Also, there were dozens of Japanese certifiers of ArchiMate, a design language standard that visualizes the EA design based on TOGAF. Based on this architecture capability, we judged that the soil promoting assurance has been achieved. We decided to develop the Teaching Course towards to extend O-DA.

It is a general application economically and also in terms of capability, especially for applications where Human interaction Blocks involving humans are the main object, for each TOGAF ADM phase / process, input dataset, application process, Then, for each possible mutual relationship with that output data, confirm the viewpoint. The philosophy of version up design to O-DA 2.0 is to verify systems that have no defect on safety, functional, and performance, by strengthening processes such as assurance case argument with syllogism. This is a concrete first step on the basis of architecture, and of course it is difficult to expect completion at once. Therefore, by advancing modeling with TOGAF and ArchiMate in EA-based design process, we can promote the oversighted assurance process. This is to provide design and verification processes, to clarify the responsibility system, and to maintain and improve the reliability of business.

2 O-DA

The Open Group standardized the O-DA (Open Dependability through Assuredness) [1] as the framework for assuring Open System dependability [2]. Before O-DA, it is not clear how to assure the architectural dependability of enterprise systems as well as how to apply system assurance methods, such as the goal structuring notation (GSN), for enterprise architecture.

The O-DA standard is based on TOGAF (The Open Group Architecture Framework) [3] and it outlines the set of valuable knowledge for mitigating risk associated with dependability of complex interoperable systems based on assurance cases. The assurance case is used to assure the target of evaluation based on claims, strategies, context, and evidences. GSN is used to describe assurance cases [5-8]. Fig.1 outlines knowledge configuration of O-DA framework. The O-DA framework is decomposed by AADM (Assured Architecture Development Method) corresponding to Architecture Development Method (ADM) of TOGAF. ADM consists of the following phases.

Phase P: Preliminary activities are achieved to develop enterprise architecture.

Phase A: Architecture vision is defined.

Phase B: Business architecture is developed

Phase C: Information system architecture is developed.

Phase D: Technology architecture is developed.

Phase E: Opportunity and solutions are clarified to realize the enterprise architecture to integrated business, information, and technology architecture.

Phase F: Transition architecture is defined to achieve the target architecture from the base line architecture.

Phase G: Implementation and governance activities are achieved for the target enterprise architecture.

Phase H: Change management of the realized target enterprise architecture is controlled.

In case of AADM, assurance cases are used to build consensus among stakeholders to ensure dependability of the target enterprise architecture in the course of ADM phases. The enterprise architecture can be assured by developing assurance cases in all the phases of AADM. The O-DA application knowledge provides Architecture based assurance case engineering (ABACE) [11, 12], FABACE (Formal ABACE), Assurance case review method [15], SPRME (Subject, property, Risk, Measure, Ev-idence) method, Assurance case capability index, and O-DA template [14]. FABACE provides a method to develop evidence by formal methods such as B [20], and Event-B [21, 22]. The O-DA application knowledge utilizes elementary knowledge, such as, TOGAF, ADM, ArchiMate, Assurance case, and Formal methods.

3 O-DA Knowledge

The O-DA knowledge consists of the Core and Empirical knowledge. The Core knowledge includes Basic concepts of the open dependability. The Empirical knowledge includes practical knowledge based on the application experience of the Core knowledge. The table 1 shows the configuration of the knowledge.

Table 1: Knowledge Configuration

Category	Items
Core	Background knowledge Syllogism Requirements engineering Assurance case O-DA essentials ArchiMate
Empirical	O-DA template Two stage review

3.1 Core Knowledge

The core knowledge include the six knowledge on background, syllogism, requirements engineering, assurance case, O-DA, and ArchiMate.

3.1.1 Background knowledge

Although there are many seminars to learn assurance cases, certification capability has not been provided. Moreover, the improvement method of quality assurance using assurance case is not clarified by current assurance case seminars. To resolve the above issues, the O-DA courseware provides case studies by advanced experts, lectures for quality assurance, voluntary workshop using participants problems, and examinations to certify participants knowledge.

According to the announcement of the American National Institute of Standards and Technology (NIST) in 2002, in a normal application, a system error after operation.

The recovery cost was 30 times the cost of finding and correcting errors in the design stage. Also, in 2009, there were eight more complicated aircraft and defense system manufacturing companies reporting 44 times with software cooperative research based on CMU 's NIST thesis.

According to the results of the NIST survey, 70% of software errors occur during requirements definition and design phase. It is also reported only 3.5% of errors were found and solved at this upper stage. 20% of errors was happens in the unit test, 16% of errors is found and resolved.

If we can find 53.5% to 63.5% errors from the upstream design stage, there is a possibility that the cost of total software testing can be reduced by 50% to 70%.

3.1.2 Syllogism

The Syllogism provides the fundamental notions to infer conclusions based on propositions and evidence. The following three steps are used to deduce the conclusion.

The major premise holds.

And a minor premise also holds.

Therefore, the conclusion is deduced.

From the point of syllogism, Claim, argument, and evidence correspond to conclusion, major premise, and minor premise. Therefore, syllogism provides the fundamental rational thinking process to understand the assurance case knowledge. It is difficult to describe assurance cases without the skill of syllogism.

3.1.2 Requirements engineering

The communication problem of requirements is first explained. This is because the most common cause of system failures is the requirements communication problem. Then, the requirements engineering process is overviewed from the point of quality assurance. The dependability attributes and functional safety standard are explained. Moreover, the basic concepts of hazard analysis are necessary for achieving the quality assurance.

3.1.3 Assurance case

The graphical notation of assurance case is briefly introduced. Then assurance case patterns are explained to understand argument structures. The category of assurance case decomposition patterns are target based, reference model based, conditional, inference based, evidence based and reuse based. The numbers of argument patterns for each category are 15, 10, 7, 4, 11, and 2. Each pattern describes problem, premise, alternatives, merit &demerit and remarks for application.

3.1.4 O-DA essentials

The purpose of O-DA, overview, terminology, and future directions are introduced. Then, we explain the O-DA framework that are constituted by dependability, development of assurance cases, accountability, failure response and change management cycles. We also explain the guidelines that are the structure and examples of assurance cases. Moreover, O-DA template is introduced to explain the typical business case of O-DA. The template describes the set of assurance cases for an Architecture Quality Evaluation Service.

3.1.5 ArchiMate

We introduce the overview of ArchiMate 3.0. A method to describe assurance cases using ArchiMate is then explained for assuring an elevator control system, smart card application, and IoT service security.

3.2 Empirical Knowledge

The empirical knowledge is created through the workshop. Participants of the workshop study and apply the O-DA knowledge to perform quality assurance using the actual architecture development cases of their own. Participants experience is extracted through quality assurance activities by practicing the assurance process. Consideration on organizational approach is based on how to construct architecture design, solution, and operation divisions. The followings are examples of the empirical knowledge based on the application of O-DA.

3.2.1 O-DA template [14,24]

The O-DA template has been proposed to clearly define the relationship between O-DA phases and ArchiMate concepts. The application of the O-DA template for the automotive sector has also been evaluated.

3.2.2 Two Staged Review [23]

A method of two stage review of design documents was defined based on TOGAF. ADM (Architecture Development Method) was used to improve design document review processes by implementing the two-stage third-party review process into real software development projects. The two staged review consists of two perspectives of "natural language" and "quality characteristics".

The case study for the Japanese middleware development projects showed that the project where the two-stage third-party review was carried out had lesser failures in System Test and later processes as compared to the project where it was not carried out.

4 Knowledge Development

4.1 Knowledge Development Process

The two layered knowledge of O-DA is developed by the following process. There are four activities consists of systematize, apply, collaborate, and extend. The knowledge components are systematized into the core O-DA knowledge. The core theoretical knowledge is explicitly described. The core O-DA knowledge is applied to empirical cases and then the empirical knowledge collaboration is occurred. In the course of the collaboration, the validated empirical knowledge is used to extend the core knowledge.

As the core knowledge is the theoretical knowledge, the above knowledge creation process can be represented by the Figure 1. We call the knowledge creation process as SACE for Systematize, Apply, Collaborate, and Extend.

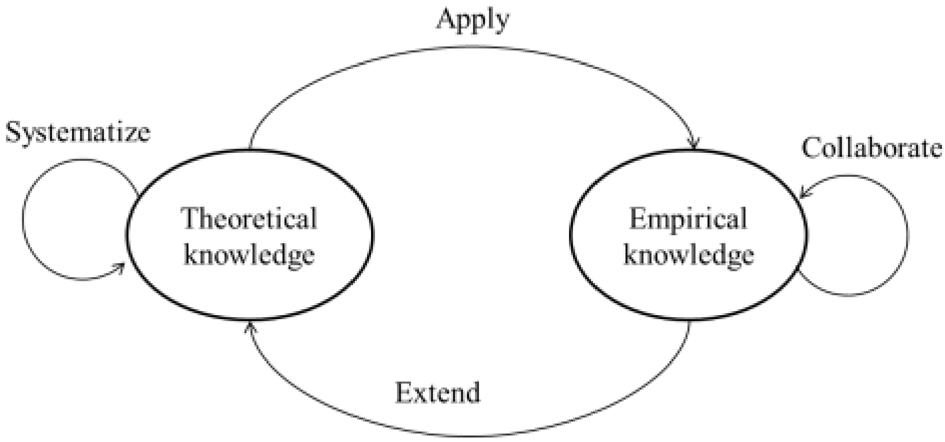

Figure 1: Knowledge Creation Process

4.2 Case Studies

This section shows additional O-DA knowledge development cases according to four processes.

4.2.1 Systematize

The Model Based Jobs Theory (MBJT) [38] is invented by integrating the Goal Oriented Requirements Engineering (GORE) [27, 28] and ArchiMate knowledge as follows. The MBJT is designated based on the anatomy of the Jobs Theory (JT) proposed by Christensen [37]. The essential concept of JT consists of customer, concerns, situation, cause, job, progress, and solution. These concepts are respectively mapped to stakeholder, concerns, problematic situation, cause, process, future goal and solution that are GORE concepts. Moreover, GORE concepts can also be represented by ArchiMate icons that are actor, value, driver, assessment, business process, goal, and requirement.

The meta model of the MBJT is shown in Figure 2. The generic MBJT pattern diagram is able to describe based on the meta model.

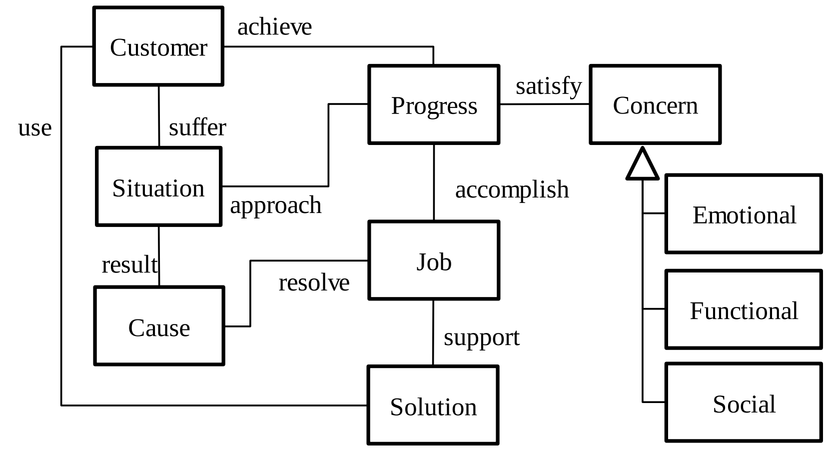

Figure 2: Meta model of MBJT knowledge

4.2.2 *Apply*

The MBJT knowledge is applied to the Healthcare domain to develop Healthcare business model patterns [39, 40]. The five key business model elements ASOMG of e-Health services have been extracted by analyzing existing e-Health business models. The elements of ASOMG are not depended on the e-Health domain. Actor, Service and Object are corresponded to Subject, Verb, and Object. SVO is the basic elements of natural language statements. Means and goals are also generic. This consideration derives the generality of the ASOMG structure. The meta model of e-Health service has been developed based on ASOMG as shown in Figure 3.

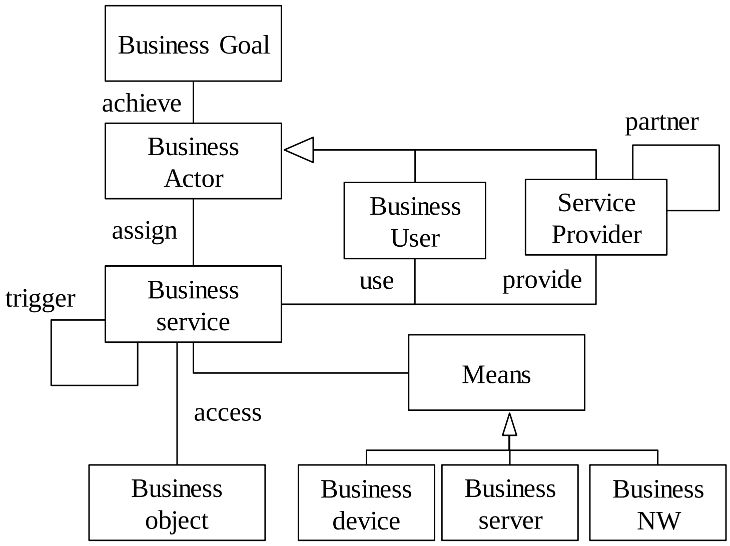

Figure 3: Meta model of Business model knowledge

The ArchiMate pattern of e-Health service has been developed by mapping ASOMG elements to the corresponding ArchiMate elements. The e-Health Business Modeling Method has been proposed based on ASOMG, and the ArchiMate pattern using the meta model of e-Health service. The applicability of the proposed e-Health Business Modeling Method has been evaluated by the case study on e-Health services. Although the e-Health Business Modeling Method was created by existing e-Health business models, the resulted method is not depended on the e-Health domains. Therefore, the proposed method is expected to apply various business domains including e-Business, e-learning, and e-Government.

4.2.3 Collaborate

The IMSA (Intra Model Security Assurance) [41] approach provides the high efficiency to assure security, because it can directly assure security of assets in the same diagrams without exchanging different diagrams. ArchiMate and Assurance case are collaborated to develop IMSA knowledge that use ArchiMate to describe assurance cases. Figure 4 integrates meta-models of architecture and security case. The meta-model of architecture consists of the target of assurance, elements and relationships. The target of assurance represents the system as a whole. The meta-model of security case consists of target of assurance, property, risk, counter measure, and evidence. The evidence will be realized by elements of the target system.

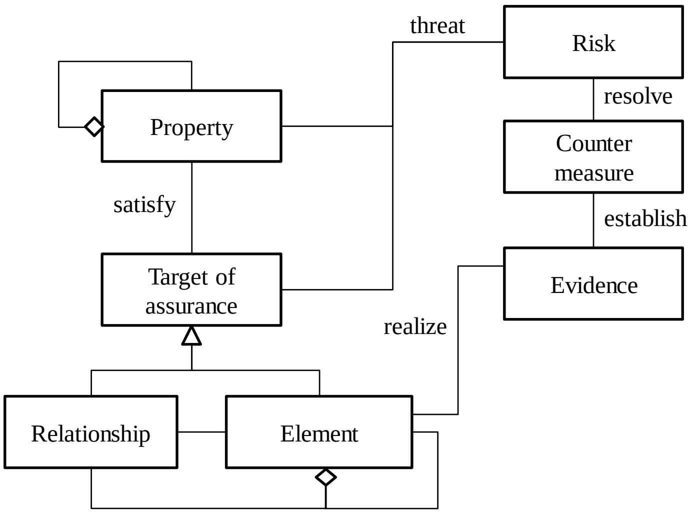

Figure 4: Meta model of IMSA knowledge

4.2.4 Extend

The EA modeling approach using ArchiMate is formalize to extend the thinking process as the Aspect Analysis method [42]. Although ArchiMate is the language to model Enterprise Architecture using diagrams, it is not easy to learn and use correctly ArchiMate because it has more than 60 graphical icons. The generic knowledge how to use ArchiMate is highly expected. As ArchiMate is designed from the point of aspects consists of the behavior, passive and active structure, we proposed the aspect analysis approach.

For Business, Application and Technology layers of Enterprise Architecture, verbs and nouns that represent behavior, object and subject are described. Verbs are corresponded to behaviors of architecture layers. The nouns are assigned to passive and active structure column. The motivation elements are omitted in the aspect analysis table. The reason is to reduce the complexity of ArchiMate diagrams.

The Table 2 shows the template of aspect analysis table. The elements of the table are derived from natural language sentences. Therefore, S, V, and O in the template represent Subject, Verb, and Object, respectively.

Table 2 : Aspect analysis table template.

Layer	Behaviour	Passive structure	Active structure
Business Architecture	V+O	O	S
Application Architecture	V+O	O	S
Technology Architecture	V+O	O	S

The behaviour elements are described in the form of <verb> <noun>, where <noun> shows the object word. The passive structure elements are described in the form <noun>. The <noun> can be a composite noun. The active structure elements are nouns that represent subject words. Composite nouns are used to specify subjects.

The aspect analysis method is extracted and extended from the ArchiMate application experiences of EA practitioners.

5 Related Work

The Open Group Real Time & Embedded Systems Forum focuses on standards for high assurance, secure dependable and complete systems [1]. At the heart of this O-DA (Open Dependability through Assuredness) standard, there is the concept of modeling dependencies, building assurance cases, and achieving agreement on accountability in the event of actual or potential failures. Assurance cases are necessary to assure architectures of dependable systems [1, 2]. Assurance cases are used to show validness of claims by evidence. GSN (Goal Structuring Notation) was also used to describe assurance cases [7-10]. The DEOS process was proposed to manage dependability of complex systems by using dependability cases [1, 2]. The dependability case is an assurance case for assuring dependability. The DEOS process [2] is an integrated iterative process containing the change accommodation cycle and the failure response cycle.

O-DA will benefit organizations relying on complex systems to avoid or mitigate the impact of failure of those systems. O-DA includes the DEOS process mentioned before. The Change Accommodation Cycle and the Failure Response Cycle that together provide a framework for these critical processes. O-DA brings together and builds on The Open Group vision of Boundaryless Information Flow. These concepts include O-DM (Open Dependency Modeling) and Risk Taxonomy of The Open Group Security Forum, and Architecture models of The Open Group ArchiMate® Forum [4]. ArchiMate can be to describe enterprise architecture models [5,6]. Approaches to assure architecture were proposed by using ArchiMate [11, 12]. The O-DA template [14] has been proposed to clearly define the relationship between O-DA and ArchiMate concepts. The application of the O-DA template for the automotive sector has been shown [24].

Perroud and Inversini proposed the Enterprise Architecture Patterns, EAP, as practical solutions for IT-Architecture problems [13]. Although EAP showed 3 business, 5 support, and 5 infrastructure patterns, no pattern to integrate all the architecture layers was considered. The O-DA template can be considered as the pattern of EA Patterns, because it contains all EA artifacts through ADM processes.

Nonaka and others proposed SECI framework as the knowledge creation process of organizations [25,26]. The SECI framework defines four knowledge transformation between explicit and tacit knowledge. The combination process of SECI model is a means of systematizing explicit knowledge that strategically and analytically integrates and combines explicit knowledge expressed from tacit knowledge within an organization in order to systematize explicit knowledge. The SECI model only explains the process of knowledge creation. Therefore, the SECI model is difficult effectively to use as the method of designing knowledge.

Duncan [27] defined the organizational ability to perform incompatible strategic actions as Ambidexterity. For example, continuous improvement of existing businesses and creation of new businesses require different capabilities. Improving the continuous improvement of existing businesses is the capability to deepen existing products and services. On the other hand, creating a new business is the capability to search for new products and services. This deepening and exploration capabilities are likely to conflict with each other. The ambidexterity defined the conflict between the two capabilities, deepening and exploration, but did not clarify the interaction process of these capabilities. Also, Duncan has not explicitly explained the knowledge required for the deepen and explore capabilities.

Teece [28] defines the dynamic process between two different capabilities to combine resources, including knowledge, to create new value and transform an organization. The process integrates an ordinary capability for efficient use of management resources and a dynamic capability for creating new value in response to changes in the business environment. Dynamic capability requires co-specialization to create values that complementarily combines different resources. By deploying Teece's product co-specialization to IT, Queiroz [29] defined the concept of co-specialization of business processes and IT. Teece's dynamic and ordinary capabilities correspond to the exploration and deepening of duality. For this reason, Popadiuk et al. [30] consider the relationship between duality and co-specialization.

The SACE proposed in this paper defines four activities between theoretical and empirical knowledge. The SECI clarifies knowledge dissemination process. In contrast, SACE is focusing on the knowledge evolution through empirical studies.

Quartel et al. [31,32] proposed a method to integrate EA models with GORE (Goal Oriented Requirements Engineering). Teka et al. [33] compared expressive power of TROPS [34] and NFR (Non Functional Requirements) framework [35] by using ARMOR. ARMOR is a visual language to describe motivation model of EA by using goals and requirements. Boness et al. [36] integrate Goal Oriented method and EA models by using a meta model.

6 Discussion

6.1 Effectiveness

The knowledgeware has been successfully lectured to practitioners. This shows the effectiveness of the proposed knowledge. Moreover, practitioners who learned the knowledge started to apply the knowledge to their own projects. The knowledge application working group include the following four projects as shown in Table 3. Each project is allocated to different TOGAF phase. Four corporations are assigned to each project. In the course of these projects, the theoretical O-DA knowledge is evaluated and the empirical knowledge is extracted.

Table 3: Team configuration

Team	Phase	Design	Evaluation &Test	ArchiMate
A	Preliminary Architecture vision	Concept of operations	Operation evaluation	Motivation layer
B	Business architecture	Requirements elicitation	Operational process	Business layer
C	Information Architecture	Requirements definition	System test	Application layer
D	Technology architecture	External design	Integration test	Technology layer

6.2 Limitation

The number of knowledge development cases was four. It is necessary to have more different kinds of applications to evaluate the effectiveness of the SACE process. Although application studies were qualitatively evaluated, it is also necessary to evaluate the impact of O-DA quantitatively. As Knowledge Creation Process SACE between theoretic and empirical knowledge is generic, SACE can be applied to various business domains. Case studies on SACE for business domains other than system development are needed.

7 Conclusion

In this paper, the O-DA knowledge design was proposed. The O-DA knowledge has been lectured to the Enterprise Architects. The knowledge consists of theoretical and empirical knowledge. It also includes knowledge evolution process named SACE. The result shows the effectiveness of the O-DA knowledge. For example, the participants of the study group have started to apply the knowledge and try to extract empirical knowledge in their own projects.

Future work includes another theoretical study of O-DA knowledge for improvement knowledge integration with other safety critical knowledge [16, 17] and formal method knowledge [20, 21,22]. It is also necessary to include empirical knowledge by case study of O-DA core knowledge.

Acknowledgements

Many thanks to the reviewers for their valuable comments to improve this paper.

References

[1] The Open Group, “Dependability through Assuredness (O-DA) Framework,” November 2013.

[2] M. Tokoro., “Open Systems Dependability,” CRC Press, May 2015.

[3] The Open Group, “TOGAF Version 9.1,” Van Haren Publishing, 2011.

[4] The Open Group, “ArchiMate 3.0 Specification,” Van Haren, 2016.

[5] M.c lankhorst et al., “Enterprise Architecture at Work -- Modeling Communication and Analysis, ” Third Edition, Springer, 2013.

[6] G. Wierda, “Mastering ArchiMate – A Serious Introduction to the ArchiMate Enterprise Architecture Modeling Language,” Edition II, The Netherlands Published by R&a, 2014.

[7] T. Kelly, “A Six-Step Method for the Development of Goal Structures,” York Software Engineering, 1997.

[8] T. Kelly, and J. McDermid, “Safety Case Construction and Reuse using Patterns,” University of York, 1997.

[9] T. Kelly, “Arguing Safety, a Systematic Approach to Managing Safety Cases,” PhD Thesis, Department of Computer Science, University of York, 1998.

[10] T. Kelly and R. Weaver, “The goal structuring notation–a safety argument notation,” In Proceedings of the dependable systems and networks 2004 workshop on assurance cases, 2004.

[11] S. Yamamoto., “An approach to assure Dependability through ArchiMate,” In International Conference on Computer Safety, Reliability, and Security, Springer, 2015, pp. 50–61.

[12] S. Yamamoto and N. Kobayashi., “Mobile Security Assurance through ArchiMate,” In The 2016 International Symposium on Mobile Internet Security, 2016.

[13] S. Yamamoto and S. Morisaki., “A case study on architecture quality assurance service using O-DA,” In Conference on ENTERprise In- formation Systems 2016.

[14] T. Perroud and R. Inversini, “Enterprise Architecture Patterns—Practical Solutions for Recuring IT-Architecture Problems,” Springer, 2013.

[15] S. Yamamoto, S. Morisaki, “A System Theoretic Assurance Case Review,” ICCSE 2016, 11th International Conference on Computer Science & Education (ICCSE), 2016, pp. 992 - 996.

[16] S. Yamamoto., "A Knowledge Integration Approach of Safety-critical Software Development and Operation based on the Method Architecture," In Procedia - Procedia Computer Science, Elsevier Masson SAS, 2014, pp. 1718–1727.

[17] S. Yamamoto., "A Systematic Knowledge Education Approach for Safety-Critical System Development," Procedia - Procedia Computer Science, vol. 60, 2015, pp. 960–967.

[18] Shuichiro Yamamoto, "An approach for evaluating softgoals using weight," ASIAARES 2015, 2015, pp. 203-212.

[19] N. Kobayashi, S. Morisaki, N. Atsumi, and S. Yamamoto, "Quantitative Non Functional Requirements evaluation using softgoal weight," Journal of Internet Services and Information Security (JISIS), vol.6, no.1, 2016, pp37-46.

[20] A. Jean-Raymond, "The B-Book: Assigning Programs to Meanings," Cambridge University Press, 1996.

[21] Event-B, http://www.event-b.org/index.html. Home of Event-B and the Rodin Platform. 2008.

[22] I. Abbassi, M. Kmimech, N. B. Hadj-Alouane and W. Gaaloul, "Modeling and Verifying the Transactional and QoS-aware Services Composition Using Event-B," IEEE 23rd International WETICE Conference, 2014, pp. 313 – 318.

[23] H. Tsuchiya, S. Yamamoto, Y. Murakami, T. Yanagisawa, N. Kobayashi, and J. Wan, "TWO-STAGE THIRD-PARTY REVIEW PROPOSAL IN SOFTWARE DEVELOPMENT," Procedia Computer Science, vol. 126, 2018, pp.1187-1196.

[24] N. Kobayashi, S. Yamamoto, "An Evaluation of O-DA template," Proc. EAIS2017, 2017, pp.263-268.

[25] I. Nonaka, "A Dynamic Theory of Organizational Knowledge Creation," Organization Science vol.5, 1994, 14-37.

[26] I. Nonaka, and H. Takeuchi, "The knowledge creating company," New York: Oxford University Press, 1995.

[27] R. Duncan, "The Ambidextrous Organization: Dealing Dual Structures for Innovation," In R. Kilman, L. Pondy, D. Slevin(eds.), The Management of Organization Design: Strategies and Implementation, North Holland, pp.167-188, 1976.

[28] D. Teece, "Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy," Research Policy (15:6), pp. 285-305, 1986.

[29] M. Queiroz, "Business Process and IT Cospecialization: Conceptualization and Suggestions for Future Research," 23rd Americas Conference on Information Systems, pp.1-10, 2017.

[30] S. Popadiuk, A. Luz, C. Kretschmer, "Dynamic Capabilities and Ambidexterity: How are These Concepts Related?," RAC, Rio de Janeiro, v. 22, n. 5, art. 1, pp. 639-660, setembro/ outubro, 2018.

[31] D. Quartel, W. Engelsman, H. Jonkers, M. van Sinderen, "A Goal-Oriented Requirements Modelling Language for Enterprise Architecture," Proc. International Enterprise Distributed Object Computing Conference, 2009, pp.3 – 13.

[32] W. Engelsman, D. Quartel, H. Jonkers, M. van Sinderen, "Extending enterprise architecture modelling with business goals and requirements," Enterprise Information Systems, vol.5, no.1, 2011, pp.9–36.

[33] A. Teka, N. Condori-Fernández, I. Kurtev, D. Quartel, W. Engelsman, "Change impact analysis of indirect goal relations: Comparison of NFR and TROPOS approaches based on industrial case study," Proc. Second IEEE International Workshop on Model-Driven Requirements Engineering (MoDRE), 2012, pp.58 – 67.

[34] J. Mylopoulos, J. Castro, M. Kolp, "Tropos: A Framework for Requirements-Driven Software Development." in Information Systems Engineering: State of The Art And Research Themes, 2000, pp.261–273.

[35] L. Chung, B. Nixon, E. Yu, J. Mylopoulos, "Non-Functional Requirements in Software Engineering," Kluwer Academic Publishers, 2000.

[36] K. Boness, R. Harrison, "The Synergies between Goal Sketching and Enterprise Architecture," Proc. MoDRE 2015, pp.46-52.

[37] C. Christensen, R. Hall, K. Dillson, D. Duncan, "Competing Against Luck," HarperCollins Publishers LLC, USA, 2016.

[38] S. Yamamoto, "MBJT-- Model Based Jobs Theory," Japan Society for Information and Management 75th Annual Conference, 2017, pp.237-240 (in Japanese).

[39] S. Yamamoto, N. Olayan, J. Fujieda, "e-Healthcare Service Design using Model Based Jobs Theory," Proc. InMed2018, Procedia Computer Science, 2018, pp.198-207.

[40] S. Yamamoto, N. Olayan, J. Fujieda, "Using ArchiMate to Design e-Health Business Models," Acta Scientific Medical Sciences vol.2, no.7, 2018, pp.18-26.

[41] Q. Zhi, S. Yamamoto, S. Morisaki, "IMSA-- Intra Model Security Assurance, " Journal of Internet Services and Information Security, vol.8, no.2, (DOI:10.22667/JISIS.2018.05.31.033),2018, pp. 18-32.

[42] S. Yamamoto, Q. Zhi, Z. Zhou, "Aspect Analysis towards ArchiMate Diagrams," Proc. KES 2019, Procedia Computer Science, vol.159, 2019, pp. 973-980.