Mobile Security Assurance through ArchiMate

Shuichiro Yamamoto* and Nobuhide Kobayashi
Nagoya University, Furo-cho Chikusa-ku, Nagoya Aichi 464-8601, Japan
syamamoto@acm.org, nobuhide@dcinc.co.jp

Abstract

ArchiMate is used to describe Business, Application and Technology Architectures models for Enterprise Architecture. Although ArchiMate provides mobile architecture modeling capability, it is not mentioned how to show the security related assuredness of mobile architecture modeled by ArchiMate. In this paper, a method to create security assurance cases from mobile architecture models in ArchiMate is proposed to argue the mobile security. The method can also create security assurance cases for EA models in ArchiMate.

Keywords: mobile security, mobile architecture, enterprise architecture, ArchiMate

1 Introduction

Complex mobile systems, especially where the boundaries of operation or ownership are unclear, are often subject to change: network environment change, new devices are introduced, regulations change, business partners are added, etc. So when the vulnerabilities of the mobile system can have a significant impact on security, income or reputation, it is critical that a process is in place to identify these threats and to update the mobile architecture by using the assurance cases and the agreements on accountability. It is also critical that a process is in place to detect vulnerabilities or threats, to understand the causes, and to prevent them from impacting the mobile system in the future.

In this paper, an assurance case creation method is proposed to argue the assuredness for mobile architecture models. The architecture models can be described by ArchiMate which is standardized to represent TOGAF (The Open Grope Architecture Framework). Assurance document is necessary to explain that mobile architecture models are secure. The security case is an approach to show the secured-ness of the target architecture based on the assurance case. Section 2 describes related work on approaches for mobile security cases. Section 3 describes the security case creation approach which is based on the structure of mobile architecture model in ArchiMate. In section 4, an example case study is presented. Discussions on the effectiveness of the proposed approach are shown in section 5. Our conclusions are presented in section 6.

2 Related work

The Open Group Real Time & Embedded Systems Forum focuses on standards for high assurance, secure dependable and complete systems. The Open Group announced the publication of the Dependability through Assuredness™Standard (O-DA) published by The Open Group Real-Time & Embedded Systems Forum[20]. 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. Dependability cases are necessary to assure dependable

IT CoNvergence PRActice (INPRA), volume: 4, number: 3 (September 2016), pp. 1-8
*Corresponding author: Nagoya University, Furo-cho Chikusa-ku, Nagoya Aichi 464-8601, Japan, Tel: +81-527894716

systems[5]. The DEOS (Dependability Engineering for Open Systems) process was proposed to manage dependability of complex systems by using dependability cases[19, 18, 2].

O-DA brings together and builds on The Open Group vision of Boundaryless Information Flow. The vision includes O-DM (Open Dependency Modeling) and Risk Taxonomy of The Open Group Security Forum, and Architecture models of The Open Group ArchiMate®Forum[3, 6]. However, the relationship between O-DA and ArchiMate concepts has not yet been clear. The safety case, the assurance case, and the dependability case are currently the focus of considerable attention for the purpose of providing assurance and confidence that systems are safe. Methods have thus been proposed for representing these using Goal Structuring Notation (GSN)[11, 12, 10, 13, 9]. GSN patterns were originally proposed by Kelly and McDermid[10]. In the absence of any clearly organized guidelines concerning the approach to be taken in decomposing claims using strategies and the decomposition sequence, engineers have often not known how to develop their arguments. It is against this backdrop that the aforementioned approaches to argument decomposition patterns –architecture, functional, attribute, infinite set, complete (set of risks and requirements), monotonic, and concretion–were identified by Bloomfield and Bishop[1]. An experimental result of argument patterns was reported by Yamamoto and Matsuno [24].

Howard, and Leblanc proposed the STRIDE model for analyzing security[4]. The acronym STRIDE was derived from the six threat categories, Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of services, and Elevation of privilege. Although the STRIDE model is useful, the relationship among human operations and system components is unclear. It is necessary to analyze security over business, application, and technology architectures in a systematic way. Preschern et al. proposed an approach to analyze security based on safety case[17]. Yamamoto et al. [23] proposed a security case pattern based on Common Criteria for the security domain. Patu and Yamamoto examines several case studies to create security cases[16, 15, 14]. Kaneko et al.[25, 8, 7] proposed the CC-case, which means Common Criteria case, to integrate security analysis over the life-cycle process.

Although Yamamoto recently proposed the method to create assurance cases based on ArchiMate models[21], the method to create security cases from ArchiMate was not mentioned. Yamamoto [22] discussed an approach to resolve conflict between safety and security by using attribute GSN.

3 Security case creation approach for mobile services

The approach uses ArchiMate to show mobile services in the three layered enterprise architecture, that is business, application and technology. After defining the mobile service architecture, the vulnerability analysis is conducted based on the architecture. Then, security measures are considered for the identified vulnerability. To validate the service security, security cases are described by integrating above artifacts such as identified vulnerability for architecture components, and measures.

3.1 Pattern of created security case

The created security case has a common structure as shown in Table 1. The security case is the application of the assurance case for the security domain.

The first level sub-goal claims state that concept elements and relationships of the model satisfy dependability principles. The second level sub-goal claim states that category of elements and their relationships among the model satisfy dependability principles.

The third level goals are decomposed by instances of concepts and relationships of the models.

The fourth level goals are decomposed by risks for the corresponding instances and are supported by the evidence to mitigate risks. Therefore, the fifth level of the assurance case consists of evidences for the fourth level goals.

In the course of the assurance case decompositions, XML definitions for the model, quality properties, and risk measures are used.

Table 1: Mobile security case pattern with ArchiMate

Hierarchy	Description
Root goal	The root goal states that the model shall satisfy security principle
Architecture layers and relationships	Root goal is decomposed by architecture layers and relationship of the ArchiMate model
Instances of layers and relationships	Third level goals are decomposed by instances of nodes and relationships of the ArchiMate model
Measures for instance vulnerability	Fourth level goals are decomposed by risks for the corresponding instances in the ArchiMate model
Evidence	Evidence supports to mitigate vulnerability of instances

4 CASE STUDY

The example study was conducted to evaluate the effectiveness of the proposed mobile security case creation method for assuring the security of a mobile content navigation service.

4.1 Target mobile service

The target system of the case study is a typical content navigation service through cell phones. The configuration of the service is shown in Figure 1 by using ArchiMate. The model describes BA (Business architecture), AA (Application architecture), and TA (Technology architecture). In BA, a simple mobile content access service process is described by triggering a tapping event. In AA, application function components and data are described. These elements are linked by realization relationship to achieve the corresponding processes in BA. There are five function components in AA. These are terminal browser, terminal AP, portal AP, AP down load, and content manager components in AA. In addition, there are four data sets including permission, personal, registration and Web content. Terminal browser, terminal AP, permission data and personal data are allocated to the cell phone device. In TA, there are five nodes, i.e., cell phone, SD card, GW, AP server, and DB server. Mobile network and Internet are also in TA. Portal AP, registration data and AP down loader are allocated to AP server. Content manager and Web content data are allocated to DB server.

4.2 Security case

After defining the mobile architecture model in ArchiMate, the vulnerability of the architecture is analyzed by checking each architecture elements based on the model. Then security case is created based on the vulnerability analysis.

Figure 2 shows the top level view of the security case. The security case is generic to all mobile services in ArchiMate, because it is independent of the detail architecture.

An example of the next level security case for BA is shown in Figure 3. The security case is decomposed by the business process described in BA. These claims are then decomposed into sub claims by

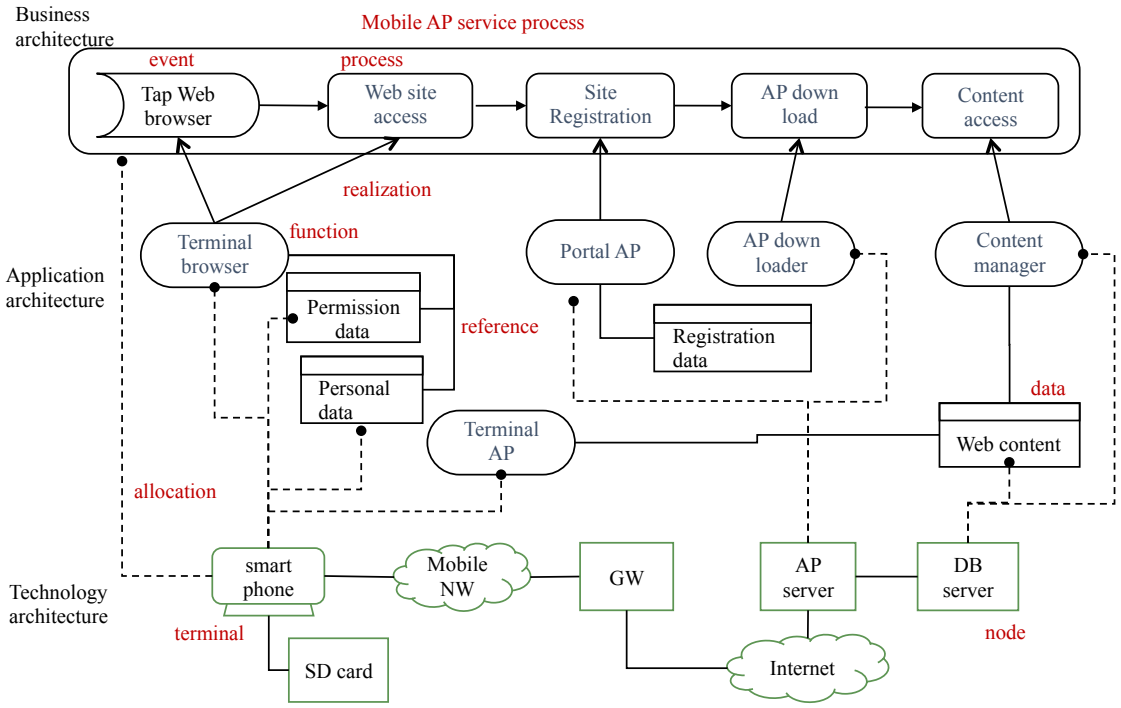

Figure 1: Example of a mobile service architecture in ArchiMate

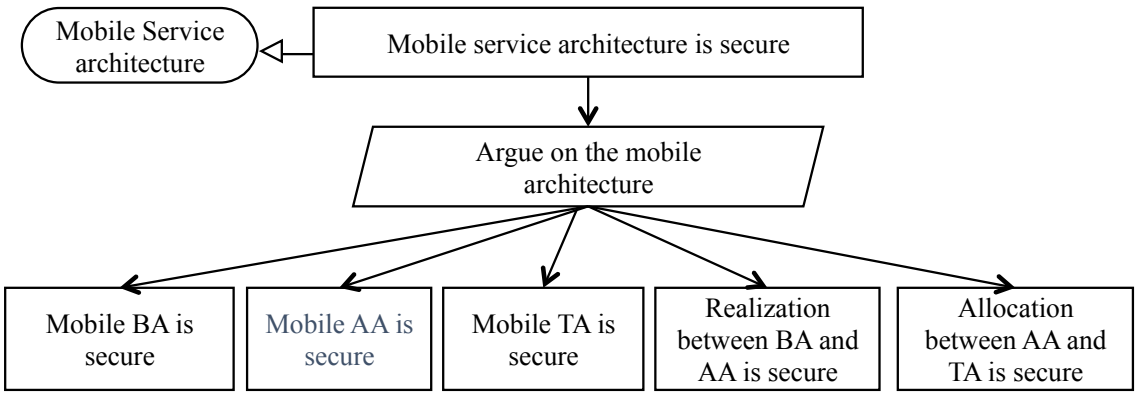

Figure 2: Example of the top level security case

analyzing vulnerability of each process. For example, in case of "Web site access is secure" claim, there are two sub claims that are corresponding to the vulnerability of fraud Web site and unsecure communication as shown in Figure 4. The validness of these two sub claims can be confirmed by measures, i.e., Web site rating and checking WiFi settings.

5 DISCUSSION

In this section, we discuss on the effectiveness, applicability, and limitation of the proposed method.

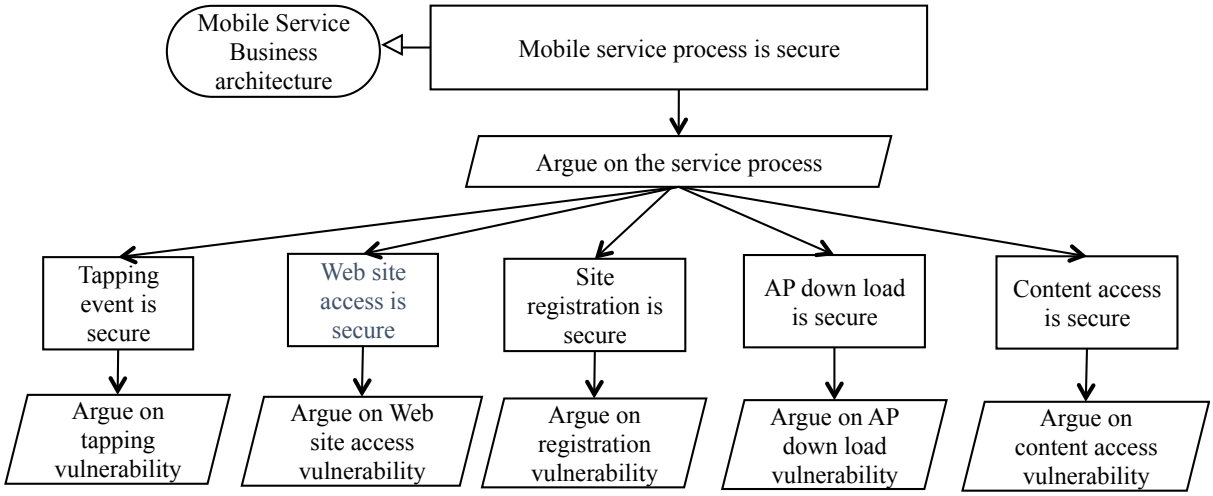

Figure 3: Example of the third level security case

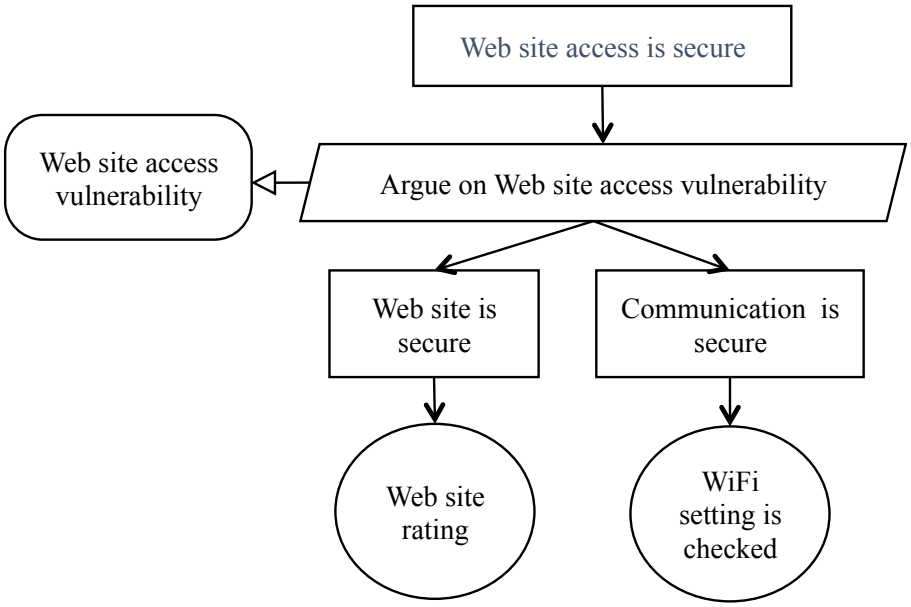

Figure 4: Example of the evidence level security case

5.1 Effectiveness

The case study on the security case creation method was executed to evaluate the effectiveness of the method. The running example showed the derivation from ArchiMate model to security case is easy and exhaustive. This showed the effectiveness of the creation method. Although the creation was only described for a single path of the security case for BA, it is clear the same decompositions can be derived for other paths as well as other architecture layers.

Moreover, the method can easily be combined with STRIDE analysis in the course of vulnerability identification based on ArchiMate models. This shows the proposed method has the capability to improve the secure architecture design productivity largely by using the security case pattern shown in Table 1.

5.2 Applicability

The applicability of the proposed method to ArchiMate is clear by the above discussions. The BA, AA, and TA described in ArchiMate models can be easily analyzed by checking vulnerability for every node. Any mobile architecture models in ArchiMate contain nodes and relationships among nodes. Therefore, the decomposition hierarchy defined by Table 1 can be applied to any mobile service models of ArchiMate. Therefore, the proposed approach can be applicable for any mobile models to assure mobile security in a systematic way.

5.3 Limitations

This paper only examines the effectiveness of the proposed method for one example architecture. More evaluations are necessary to validate the proposed method. The quantitative evaluation study for productivity of the proposed approach is necessary.

Mobile security architecture design method should also consider constraints for mobile devices. For example, mobile design consideration for portability, efficient resource consumption, and extensibility should be decided to evaluate mobile application architecture. These non-functional properties and security should be balanced in the architecture.

6 Summary

In this paper, a security case development method for mobile service is proposed to derive the argument decomposition structure based on ArchiMate model. The method also provides O-DA solutions for assuring security of business, application, and technology architecture of TOGAF. An example case study using the proposed method was shown for a mobile content access service. Discussions based on the case study showed the effectiveness and appropriateness of the proposed methods.

Future work includes the formalization of security case derivation process from ArchiMate models.

Acknowledgment

This work was supported by KAKENHI (24220001). This work has been conducted as a part of ”Research Initiative on Advanced Software Engineering in 2015” supported by Software Reliability Enhancement Center (SEC), Information Technology Promotion Agency Japan (IPA).

References

[1] R. Bloomfield and P. Bishop. Safety and assurance cases: Past, present and possible future – an adelard perspective. In *Proc. of the 18th Safety-Critical Systems Symposium (SCSS'10), Bristol, UK, LNCS*, pages 51–67. Springer-Verlag, February 2010.

[2] M. T. (Editor). *Open Systems Dependability, Dependability Engineering for Ever-Changing Systems*. CRC Press, 2012.

[3] V. Haren. *TOGAF®Version 9.1 A Pocket Guide*. Van Haren Publishing, 2011.

[4] M. Howard and D. LeBlanc. *Writing Secure Code: Practical Strategies and Proven Techniques for Building Secure Applications in a Networked World*. Microsoft Press, 2004.

[5] D. Jackson and M. Thomas. *Software for Dependable Systems: Sufficient Evidence?* National Academy Press, 2007.

[6] A. Josey. *ArchiMate®2.0, A Pocket Guide*. Van Haren Publishing, 2012.

[7] T. Kaneko, S. Yamamoto, and H. Tanaka. Cc-case as an efficient method of assurance case for the security risk management. In *Proc. of the 8th International Conference on Project Management (PROMAC'14), Kuala Lumur, Malaysia*, pages 1–1, December 2014.

[8] T. Kaneko, S. Yamamoto, and H. Tanaka. Cc-case for the system development over life-cycle process. In *Proc. of the 12th International Conference on Computer Security and Digital Investigation (ComSec'14), Kuala Lumur, Malaysia*, pages 1–1. SDIWC, March 2014.

[9] T. Kelly and R. Weaver. The goal structuring notation - a safety argument notation. In *Proc. of the Dependable Systems and Networks 2004 Workshop on Assurance Cases (DSNAC'04)*, pages 1–6, 2004.

[10] T. P. Kelly. *Arguing Safety, a Systematic Approach to Managing Safety Cases*. PhD thesis, Department of Computer Science, University of York, September 1998.

[11] T. P. Kelly. A six-step method for the development of goal structures. *ResearchGate*, pages 1–29, September 1999.

[12] T. P. Kelly and J. A. McDermid. Safety case construction and reuse using patterns. In *Proc. of the 16th International Conference on Computer Safety, Reliability and Security (SAFECOMP'97), York, UK*, pages 55–69. Springer-Verlag, September 1997.

[13] J. A. McDermid. Software safety: where's the evidence? In *Proc. of the 6th Australian Workshop on Industrial Experience with Safety Critical Systems and Software (SCS'01), Brisbane, Australian*, pages 1–5. Australian Computer Society, June 2001.

[14] V. Patu and S. Yamamoto. How to develop security case by combining real life security experiences (evidence) with d-case. *Procedia Computer Science*, 22:954–959, October 2013.

[15] V. Patu and S. Yamamoto. Identifying and implementing security patterns for a dependable security case - from security patterns to d-case. In *Proc. of the 16th International Conference on Computational Science and Engineering (CSE'13), Sydney, Australia*, pages 138–142. IEEE, December 2013.

[16] V. Patu and S. Yamamoto. A model to capture security threat patterns by complying with standards and lesson learned — archiving dependability for security assurance cases. In *Proc. of the 12th IEEE International Workshops on Enabling Technologies (ISSREW'13), Pasadena, California, USA*, pages 17–18. IEEE, November 2013.

[17] C. Preschern, N. Kajtazovic, and C. Kreiner. Security analysis of safety patterns. In *Proc. of the 20th Conference on Pattern Languages of Programs (PLOP'13), Monticello, Illinois, USA*, pages 1–38. Hillside, October 2013.

[18] D. project. Jst white paper deos-fy2011-wp-03j, 2011. `www.dependable-os.net/ja/topics/file/White_Paper_V3.0J.pdf`.

[19] D. project. `http://www.crest-os.jst.go.jp`, 2013.

[20] Real-Time and E. Systems. Dependability through Assuredness™(O-DA) Framework. `http://www.opengroup.org/news/press/open-group-releases-dependability-through-assuredness-%E2%84%A2-standar`, 2013.

[21] S. Yamamoto. An approach to assure dependability through archimate. In *Proc. of the 34th International Conference on Computer Safety, Reliability & Security (SAFECOMP'15), Delft, Netherlands, LNCS*, volume 9338, pages 50–61. Springer-Verlag, September 2015.

[22] S. Yamamoto. Assuring security through attribute gsn. In *Proc. of the 5th International Conference on IT Convergence and Security (ICITCS'15), Kuala Pumpur, Malaysia*, pages 1–5. IEEE, August 2015.

[23] S. Yamamoto, T. Kaneko, and H. Tanaka. A proposal on security case based on common criteria. In *Proc. of the Information & Communication Technology-EurAsia Conference (ICT-EURASIA'13), Yogyakarta, Indonesia, LNCS*, volume 7804, pages 331–336. Springer-Verlag, March 2013.

[24] S. Yamamoto and Y. Matsuno. An evaluation of argument patterns to reduce pitfalls of applying assurance case. In *Proc. of the 1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE'13), San Francisco, California, USA*, pages 12–17. IEEE, May 2013.

[25] C. V. Zhou, C. Leckie, and S. Karunasekera. Cc-case as an integrated method of security analysis and assurance over life-cycle process. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, 3(1):49–62, April 2014.

Author Biography

Shuichiro Yamamoto is a professor of Strategy Office, Information and Communications Headquarters, Nagoya University. Previously, he was engaged in the development of programming languages, CASE tools, network-based smart card environments, and distributed application development platforms. His research interests include distributed information systems, requirement engineering, ubiquitous computing, Knowledge creation and Knowledge management. He joined NTT in 1979. He received his B.S. in information engineering from Nagoya Institute of Technology in 1977, and his M.S. and Ph.D. in information engineering from Nagoya University in 1979 and 2000. He joined NTT DATA Corp. in 2002 and had been Deputy Senior Executive Manager (2002-2007). He had also been Research Fellow and Director of Research at the Institute of System Science (2007-2009), Research and Development Headquarters of NTT DATA Corp.

Nobuhide Kobayashi received the B.S. in information engineering from Nagoya institute of Technology in 1991, and M.S. in information engineering from Nara institute of science and technology in 1997. He is a manager of DENSO CREATE INC.. Previously, he is responsible for introducing the method related with AUTOSAR, ISO26262, productline to actual project in automotive software development. He has studied software design method in Nagoya University from 2015.