ACTA SCIENTIFIC MEDICAL SCIENCES (ISSN: 2582-0931)

Volume 3 Issue 9 September 2019

Review Article

Composite Safety Assurance for Healthcare Devices

Qiang Zhi*, Zhengshu Zhou, Shuji Morisaki and Shuichiro Yamamoto

Graduate School of Informatics, Nagoya University, Japan

***Corresponding Author:** Qiang Zhi, Graduate School of Informatics, Nagoya University, Japan.

Received: July 12, 2019; **Published:** August 02, 2019

Abstract

The safety of medical devices is critical, and the safety assurance is necessary for the production and development of medical devices. For this, we propose a method to describe safety assurance between healthcare system components in this paper. The method uses ArchiMate, which is an enterprise architecture modeling language, to express the relationships of system components. A case study on the insulin pump system, which is a medical device, is carried out to explain the method. Moreover, a comparison with the traditional composite dependability assurance approach named as the d* framework is explained to show the effectiveness of the method. The comparison results show that the proposed approach is more suitable for ensuring safety in safety-critical healthcare system architecture.

Keywords: Composite Safety Assurance; d* Framework; Healthcare Device; Critical System; System Architecture

Abbreviations

GSN: Goal Structuring Notation; EA: Enterprise Architecture; IMSA: Intra Model Security Assurance; UML: Unified Modeling Language; SysML: Systems Modeling Language.

Introduction

Some classes of system are critical systems where system failure may result in injury to people, damage to the environment, or extensive economic losses [1]. Examples of critical systems include embedded systems in medical devices, such as an insulin pump. As system failure may lead to user injury, the development of medical device often requires safety assurance.

Interdependency should be clarified for managing healthcare devices. It is necessary to describe interdependence of system actors for clarifying the safety assurance. That is to say, the interdependency of the system components, and the internal dependability of the system components should be proved to assure the dependability of a system. Although Yu [2] showed that the network of intentions among the actors can be represented using the i* framework, the problem of how to treat the dependability of systems has not been solved.

Some other methods were proposed to achieve dependability assurance. The purpose of developing safety case is to ensure the safety of a system. The Goal Structuring Notation (GSN) [3] was proposed and widely used to develop assurance and safety cases. The argument patterns [4] had been proposed to help engineers develop assurance cases. Besides, a security argument pattern and security case based on common criteria [5] has been proposed for assuring security. In the absence of any clearly organized guidelines concerning the approach to be taken in decomposing claims using strategies and the decomposition sequence, engineers often do not know how to develop their arguments. For this, assurance cases were summarized and prospected by Bloomfield and Bishop [6]. Besides, d* framework [7-10], which is a derivative of GSN, is used to assure system dependability. In d*framework, an actor node is used to relate the assurance case. The premise of development using the d*framework is the existence of a collaboration diagram. Therefore, the scope of the d*framework is limited when assuring

dependability between system components. Moreover, in terms of visualization, the performance of d* framework is not outstanding.

Goal-oriented approaches [11] are applied to analyze healthcare processes [12,13]. Enterprise Architecture (EA) [14] can be used to model medical systems. For example, Eldein [15] discussed EA for the cloud service of mobile healthcare. Ahsan [16] designed and provided the insight of an EA approach to process architecture for healthcare management. Yamamoto proposed an ArchiMate pattern to analyze e-health business model [17]. Zhi proposed a method to assure the dependability between business actors [18], but the assurance between system components is not clarified.

This paper proposed a composite safety assurance approach to describe dependability arguments among actors in safety-critical systems. Our work is accomplished through ArchiMate [19,20], which is a modeling language for development of enterprise architecture models. ArchiMate provides a clear method for visualizing construction and business processes: operation, organizational structure, information flow, application service, and technology infrastructure. The remainder of this paper is organized as follows. We first define the safety assurance method, and give a case study of insulin pump system to illustrate our method. Then a comparison with d* framework is carried out to show the effectiveness of the proposed method.

Composite safety assurance

In this section, the metamodel for composite dependability is proposed. We have proposed a intra model security assurance (IMSA) [21] method to describe security assurance, but the dependability relationships between system components cannot be treated, for the interdependence between system components, we propose the metamodel as shown in Figure 1, which shows the metamodel of the composite goal concept. The depender component depends on the composite goal. The depender component achieves the composite requirements that realize the composite goal and countermeasure through composite requirements.

Next, we use ArchiMate to describe the composite goal metamodel. Table 1 shows the mapping relationships between the metamodel and ArchiMate elements.

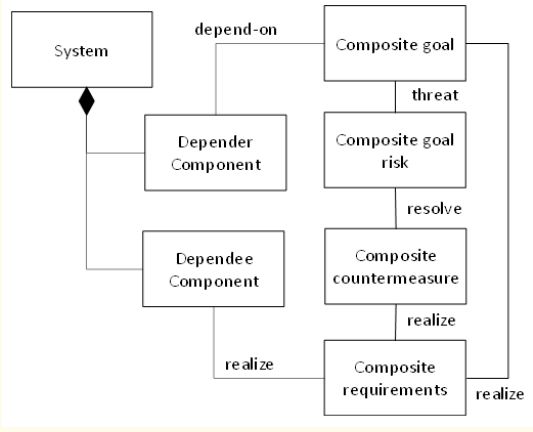

Figure 1: Metamodel of composite goal.

Composite goal meta model	ArchiMate elements
Stakeholder, System, Component	Actor, Application component, Node
Composite Goal	Driver
Composite goal risk	Assessment
Composite countermeasure	Goal
Composite requirements	Requirements

Table 1: The mapping between composite goal metamodel and ArchiMate elements.

Figure 2 shows the definition of safety case in ArchiMate. The interrelationship between elements is described by the influence and realization relationship in ArchiMate. The realization relationship is used between countermeasure and requirements. The influence relationship between safety goal, risk, and countermeasure is negative.

To describe the dependability goals in d* framework, a method for mapping the depend-on relationship between actors has been proposed. Suppose that an actor X depends on another actor Y for the safety goal. The safety goal which realized by the actor Y will support the safety of the Actor X. In this paper, the depend-on relationship is defined by the association and realization relationships in ArchiMate. Figure 3 shows an example of the depend-on

relationship using ArchiMate. In this figure, a patient is associated with the safety goal expressed as "Blood sugar is balanced," and the insulin pump realizes this goal.

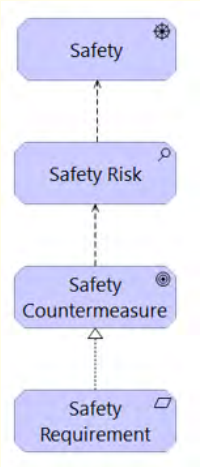

Figure 2: Example of Safety Case in ArchiMate.

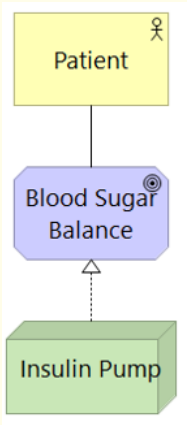

Figure 3: Example of composite safety relationship.

According to the above approach, the composite safety between patient and insulin pump is defined in ArchiMate as follow in Figure 4.

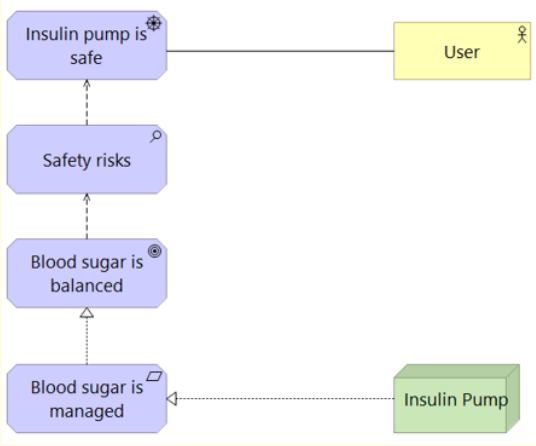

Figure 4: Using ArchiMate to describe Composite safety between Patient and Insulin pump.

Composite safety assurance method

We introduce the steps of the composite safety assurance method in this section.

STEP 1: Describe system architecture with ArchiMate

System architecture is a generic discipline to handle systems, and it is the study of early decision making in complex systems [22]. Systems modeling language (SysML) [23] and Unified Modeling Language (UML) [24] are applied to model system architecture. However, these modeling languages only focus on modeling software and system architecture. In order to achieve composite safety assurance, we use ArchiMate to describe system architecture and assurance case.

STEP 2: Identify composite safety goals between components

Complexity theory implies that system components are interdependent to the extent where changes in one component may affect another, or result in failure of interconnected systems [25]. Identifying, understanding, and analyzing critical architecture interdependencies are essential [26]. Thus, it is necessary to identify the interdependency between system components. But what safety goals should be set between system components is not within the scope of this paper. We have introduced how to use ArchiMate to define the composite safety assurance relationships, and will fur-

ther clarify this method based on an case study of insulin pump in the next section.

STEP 3: Safety goals elicitation

In order to ensure the safety of a critical system, safety goals should be extracted after confirming the relationships between system components. For goals elicitation, the risks of a system should be grasped. Safety goals can be derived from the corresponding risks. Figure 3 showed an example of safety goal, which is "Blood sugar balance".

STEP 4: Requirements elicitation for safety goals

In safety assurance, the requirements are necessary for the realization of a goal. Here, the requirement intended to finally support an elaborated goal, such as the verification results of tests or techniques. Namely, the requirements should be met by a software system in order for that system safe and stable [27]. In Figure 4, the requirement for the safety goal is "Blood sugar is managed".

STEP 5: Safety goals assurance using composite requirements

A safety goal should be realized by requirements. In general, each requirement should have corresponding evidence. In Figure 4, the corresponding evidence for the requirement is "Insulin pump", which is a system actor in the system architecture.

Case study of the proposed method

In this section, to illustrate the proposed approach, we use the 5-steps method described in the previous section to analyze an insulin pump, which is a medical device. The insulin pump in this case study is for personal use. In recent years, insulin pump is gradually accepted, and its safety has also widely received attention [28,29]. Zhang analyzed the hazards for the insulin pump [30].

STEP 1: Describe insulin pump system architecture with ArchiMate

We modeled the insulin pump system architecture using ArchiMate as shown in Figure 5. It is an ArchiMate model to illustrate how the insulin pump software transforms an input blood sugar level to a sequence of commands that drive the insulin pump. This is an embedded system, which collects the information from a sensor and controls a pump that delivers a controlled dose of insulin to a patient. In this paper, we discussed only the software related safety issues. The safety issues related to hardware and environment, such as battery and extreme environment will not be discussed.

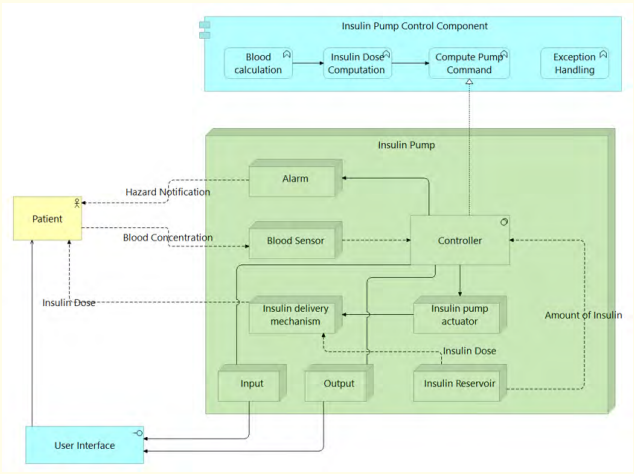

Figure 5: Insulin pump system in ArchiMate.

In the figure, the patient interacts with the insulin pump through the user interface. The patient can receive the information from the output device and inputs command through the input device using the user interface.

The insulin is administered to the patient by the insulin pump via a delivery path, which composed of the insulin reservoir, the insulin delivery mechanism, and the blood sensor. The insulin reservoir and the insulin delivery mechanism are monitored and administered by the insulin pump actuator and controller. The pump delivery mechanism can make insulin delivered from the pump to the patient at a prescribed time or rate. The insulin pump control component includes blood calculation function, insulin dose computation function, compute pump command function and exception handling function. For the exception handling, if the software fails, the safe dose of insulin will be set and insulin pump will alerts.

A software-controlled insulin delivery system might work by using a microsensor embedded in the patient to measure some blood parameter that is proportional to the sugar level. Then the blood parameter will be sent to the pump controller. This controller computes the sugar level and the amount of insulin that is needed. At last, it sends signals to a miniaturized pump to deliver the insulin via a permanently attached needle.

STEP 2: Identify composite safety goals between insulin pump system components

Next, we would like to explain the safety issues about the insulin pump. Obviously, the insulin pump system is a safety-critical system. Safety assurance is necessary in the development process. If the pump fails to operate or does not operate correctly, then the patient's health may be damaged or they may fall into a coma because their blood sugar levels are too high or too low. Therefore, the system must meet two essential requirements as follow.

1. The system should provide insulin when insulin is required.
2. The system should perform reliably and deliver the correct amount of insulin to offset current blood glucose levels.

Here, we analyze the insulin pump working process as follow

1. Blood sensor" measures blood sugar level.
2. "Blood calculation function" analyzes the result of blood sugar level.
3. "Insulin dose computation function" calculates the insulin dose.
4. "Compute pump command function" conducts based on calculation results.
5. "Insulin delivery mechanism" delivers insulin through infusion set according to the command.

STEP 3: Safety goals elicitation in insulin pump architecture

For this architecture, we analyze the 5 depend-on relationships among these actors as follow.

1. "Blood sensor" depends on "The sensor is dependable" for "Blood calculation function".
2. "Blood calculation function" depends on "The blood calculation is dependable" for "Insulin dose computation function".
3. "Insulin dose computation function" depends on "The insulin computation is dependable" for "Compute pump command function".
4. "Compute pump command function" depends on "The pump command is dependable" for "Insulin pump actuator".
5. "Insulin delivery mechanism" depends on "The insulin delivery is dependable" for "Patient".

We add these depend-on relationships as safety goals into the system architecture by using the method previously mentioned. Figure 6 shows the depend-on relationship on the insulin pump system in ArchiMate.

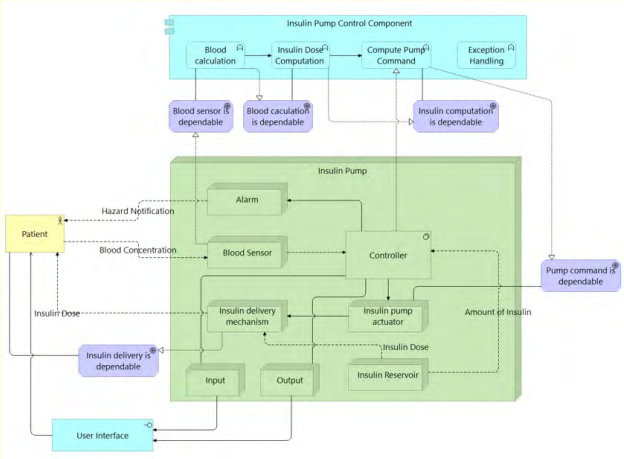

Figure 6: Depend-on relationship on the insulin pump system in ArchiMate.

STEP 4: Requirements elicitation for insulin pump safety goals

As previously mentioned, the corresponding requirements are required for safety goals. For this, there should be requirements correspond to the depend-on relationships mentioned above. We analyze these requirements as follows.

1. For "Blood sensor", the value range of blood sugar should be defined. If the measured blood sugar level is outside this range, it should stop working or deliver safe insulin dose, and alerts at the same time.
2. For "Blood calculation function", the analysis algorithm should be defined. If the data is abnormal, it should stop working or deliver safe insulin dose, and alerts at the same time.
3. For "Insulin dose computation function", the method of calculating insulin dose based on the blood sugar level should be defined, if the computation result is abnormal, it should stop working or deliver safe insulin dose, and alerts at the same time.
4. For "Compute pump command function", the pump command and exception handling should be defined.
5. For "Insulin pump delivery mechanism", the control of insulin dose should be defined. If the insulin dose is abnormal, it should stop working or deliver safe insulin dose, and alerts at the same time.

STEP 5: Insulin pump safety goals assurance using composite requirements

We implement the requirements into the insulin pump system architecture as shown in Figure 7.

1. For safety goal "The sensor is dependable", the corresponding requirements are "Exception handling" , "Alert" and "Measurement rule is defined". The evidences that support the requirements are "Exception handling function", "Alarm device" and "Blood sensor".
2. For safety goal "The blood calculation is dependable", the corresponding requirements are "Exception handling", " Alert" and "The calculation rule is defined". The evidences that support the requirements are "Exception handling function", "Alarm device" and "Blood calculation function".
3. For safety goal "Insulin computation is dependable", the corresponding requirements are "Exception handling", "Alert" and "Computation rule is defined". The evidences that support the requirements are "Exception handling function", "Alarm device" and "Insulin dose computation function".
4. For safety goal "Pump command is dependable", the corresponding requirements are "Exception handling", "Alert" and "Pump command is defined". The evidences that support the requirements are "Exception handling function", "Alarm device" and "Compute pump commands function".
5. For safety goal "Insulin delivery is dependable", the corresponding requirements are "Exception handling", "Alert" and "Insulin dose is defined". The evidences that support the requirements are "Exception handling function", "Alarm" and "Insulin delivery mechanism".

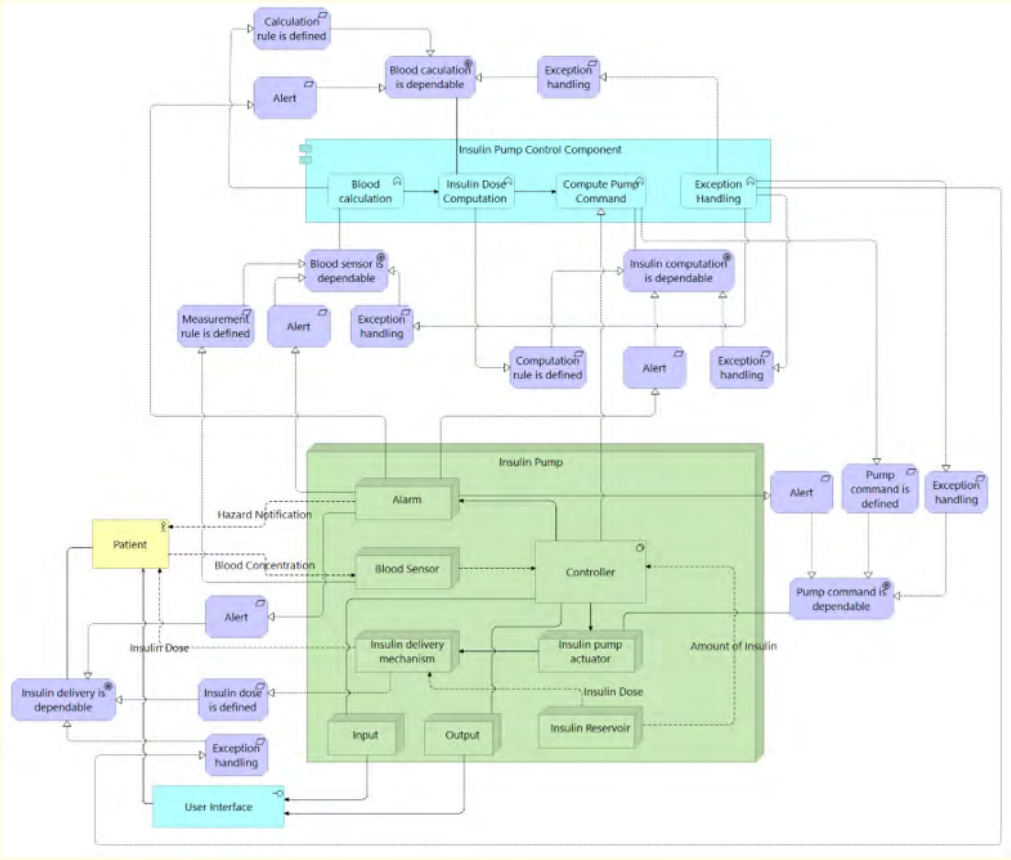

Figure 7: Composite safety on the insulin pump system in ArchiMate.

According to the above steps, we have achieved safety assurance for the insulin pump system architecture.

Discussion

In previous sections, we proposed an approach to develop composite safety assurance through ArchiMate. Moreover, a case study of insulin pump safety was carried out to illustrate this approach. To verify the effectiveness of the proposed approach, we compare it with d*framework, which is a traditional method to assure the composite dependability.

Table 2 compares the proposed method to the d*framework at system components, safety claim, and relationship. For the system components, d*framework only uses the module node or actor. In the proposed method, system components were represented by the nodes of business architecture layer, application architecture layer and technology architecture layer in ArchiMate as shown in Figure 4, Figure 5, Figure 6 and Figure7, such as Component Business actor, Interface, Function, Device, and System software. System components can be more vividly described in ArchiMate. Besides, in the proposed method, we defined the safety assurance rules by using the nodes of Driver, Assessment, Goal, and Requirement in ArchiMate. In d* framework, safety claim consists of Context, Strategy and Evidence[30] [31]. Moreover, in the proposed method, we defined components relationships by using Realize, Association, Influence, and Serving relationships in ArchiMate. However, the relationship between components in d* framework is the depend-on relationship. The proposed method can more clearly describe the relationship between system components.

Items	ArchiMate	d*framework
System components	Component	Module Node
	Business actor	Actor
	Interface	-
	Function	-
	Device	-
	System software	-
Composite Safety Claim	Driver	Goal
	Assessment	Context
	Goal	Strategy
	Requirement	Evidence
Relationship	Realize	-
	Association	-
	Influence	Depend-on
	Serving	-

Table 2: Comparison of Proposed method and d*framework.

In summary, the proposed method is superior to the d* framework in describing the system components and the relationships. For the safety assurance of healthcare systems of medical devices, the proposed approach is effectively applicable.

Effectiveness

As previously mentioned, the ArchiMate diagram is effective in safety assurance for safety-critical systems. The effectiveness is summarized as follows.

Because system components can be directly defined with system architecture in ArchiMate, the relationships between system components are further clarified. In addition, the arguments of the assurance case can be easily defined using the motivational elements and the relations in ArchiMate as previously mentioned. Besides, from the visualization perspective, using ArchiMate to describe dependability relationships has advantages over the traditional method.

Limitations

In this paper, only one case study of insulin pump was carried out, and only partial safety issues of software were analyzed. Besides, we did not consider the quantitative comparison with the traditional approach. The comparative experiment to quantitatively evaluate the productivity and quality should be carried out to verify the effectiveness.

Conclusion

In this study, a composite safety assurance method was proposed for safety-critical system architecture. First, the safety assurance model between system components was explained, then the mapping relationships were defined using ArchiMate based on this model.

The insulin pump system was carried out to explain this approach, and the study showed that the composite safety assurance between system components could be well treated using ArchiMate. Finally, a comparison between ArchiMate and d*framework was conducted. The effectiveness and superiority of the proposed method were also proven by analyzing the system components, composite safety claim, and relationship.

The significance of the proposed method in terms of the system structure, is that it can directly assure the safety-critical system architecture. However, because the d*framework uses UML, it cannot directly assure models of system architecture.

In the paper, although we gave only one case study, which is the insulin pump system, for the safety assurance, this method can be

extended to other safety-critical system architecture. In future work, it is necessary to apply the proposed method to additional safety-critical systems to confirm its effectiveness. It is also important to determine if others in different fields can understand the dependability of the system through the proposed approach, thus, a survey regarding the versatility of the proposed method will also be undertaken.

Bibliography

1. I Sommerville. Software Engineering (10th Edition), Pearson, (2015).
2. E Yu. Social modeling for requirements engineering, The MIT Press, (2011).
3. K Tim 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).
4. S Yamamoto and M Yutaka. "An Evaluation of Argument Patterns to Reduce Pitfalls of Applying Assurance Case". in Proceedings of ASSURE, San Francisco, USA, (2013).
5. S Yamamoto., *et al*. "A Proposal on Security Case Based on Common Criteria". in Information and Communicatiaon Technology - International Conference, Yogyakarta, Indonesia, 2013.
6. P Bishop and R Bloomfield. "A Methodology for Safety Case Development". *Industrial Perspectives of Safety-critical Systems* (1998): 194-203.
7. T. Saruwatari and S. Yamamoto. "D* framework creation procedure from collaboration diagram". *IT CoNvergence PRActice* 2.2 (2014): 43-54.
8. S Yamamoto and Y Matsuno. "d* framework: Inter-Dependency Model for Dependability". in International Conference on Dependable Systems and Networks, Boston, USA, 2012.
9. T Saruwatari., *et al*. "A comparative study of d∗framework and GSN". in IEEE International Symposium on Software Reliability Engineering, Pasadena CA (2013).
10. T Saruwatari and S Yamamoto. "Creation of Assurance Case Using Collaboration Diagram". in Asian Conference on Availability, Reliability and Security, Bali, Indonesia (2014).
11. AV Lamsweerde. "Goal-oriented requirements engineering: a guided tour". in Proceedings Fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada (2001).
12. M Hagglund. "A new approach for goal-oriented analysis of healthcare processes". *Studies in Health Technology and Informatics* (2010): 1251-1255.
13. Y An., *et al*. "Collaborative social modeling for designing a patient wellness tracking system in a nurse-managed health care center". in Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, Pennsylvania (2009).
14. M Lankhorst. Enterprise Architecture at Work, Springer-Verlag Berlin Heidelberg (2013).
15. A Eldein., *et al*. "Enterprise architecture of mobile healthcare for large crowd events". in International Conference on Information and Communication Technology and Accessibility, Muscat, Oman (2017).
16. K Ahsan., *et al*. "Healthcare Modelling through Enterprise Architecture: A Hospital Case". in Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA (2010).
17. S Yamamoto., *et al*. "Using ArchiMate to Design e-Health Business Models". *Acta Scientific Medical Sciences* 2.7 (2018).
18. Q Zhi., *et al*. "Visualized Assurance Approach for Enterprise Architecture". *Journal of Information and Communication Convergence Engineering* 17.2 (2019): 117-128.
19. The Open Group, ArchiMate 3.0 Specification, Van Haren Publishing (2016).
20. P Beauvoir. "Archi – Open Source ArchiMate Modelling".
21. Q Zhi., *et al*. "IMSA - Intra Model Security Assurance". *Journal of Internet Services and Information Security* 8.2 (2018): 18-32.
22. B Cameron., *et al*. System Architecture, Pearson Education Limited (2015).
23. S Friedenthal., *et al*. A Practical Guide to SysML: The Systems Modeling Language, Morgan Kaufmann (2014).
24. J Jacobson., *et al*. Unified Modeling Language Reference Manual, Addison-Wesley Professional (2004).

25. RF Stapelberg. "Infrastructure Systems Interdependencies and Risk Informed Decision Making (RIDM): Impact Scenario Analysis of Infrastructure Risks Induced by Natural, Technological and Intentional Hazards". *Systemics, Cybernetics and Informatics* 6.5 (2008).

26. SM Rinaldi., *et al*. "Identifying, understanding, and analyzing critical infrastructure interdependencies". *IEEE Control Systems Magazine* 21.6 (2001): 11-25.

27. AV Lamsweerde. Requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley (2009).

28. LP Plotnick., *et al*. "Safety and Effectiveness of Insulin Pump Therapy in Children and Adolescents With Type 1 Diabetes". *Diabetes Care* 26.4 (2003): 1142-1146.

29. L Henimann., *et al*. "Insulin Pump Risks and Benefits: A Clinical Appraisal of Pump Safety Standards, Adverse Event Reporting, and Research Needs A Joint Statement of the European Association for the Study of Diabetes and the American Diabetes Association Diabetes Technology W". *Diabetologia* 38.4 (2015): 716-722.

30. Y Zhang., *et al*. "A Hazard Analysis for a Generic Insulin Infusion Pump". *Journal of Diabetes Science and Technology* 4.2 (2010): 263-283.

31. DEOS. "D-CASE PROCESS". DEOS.

32. Y Matsuno., *et al*. "A Dependability Case Editor with Pattern Library". in International Symposium on High Assurance Systems Engineering, San Jose, CA, USA (2010).

Volume 3 Issue 9 September 2019