International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 9, Issue 5, pp 213-222, May-2025
https://doi.org/10.47001/IRJIET/2025.905028

Security System in UML Diagrams: A Literature Review

[1]*Asmaa Hadi Albayati, [2]Taghreed Riyadh Alreffaee, [3]Anfal A. Fadhil

[1,2,3]Department of Software, College of Computer Sciences and Mathematics, University of Mosul-Iraq
*Corresponding Author's E-mail: asmashade77@uomosul.edu.iq

***Abstract* - Although software artifacts nowadays have a variety of quality attributes, security has become less responsive for a variety of reasons. For example, different organizations have varied definitions of security, and as a result, they implement security metrics differently, making it difficult to determine which attributes should be taken into account when evaluating security. Due to the failure of early security design, secure software development remains a study area in many firms. Several scholars have recently proposed that security engineering be incorporated into the early phases of system modeling. This idea entails using the Unified Modeling Language (UML) for various abstractions of systems. However, the majority of these studies dealt with security injection without considering the creation of security infrastructure, producing code for both functional and non-functional sides simultaneously. This article has been reviewed The most significant studies attempt to enhance the productivity of software applications and broad Integrating security at every stage of the software development process, as opposed to only the implementation phase.**

Keywords: software security, Unified Modeling Language (UML), class diagram, use case diagrams, Security requirements.

I. INTRODUCTION

Developing secure software is one of the main objectives of software engineering. As development continues, this software's network security has gotten even weaker. One of the main causes of this is the disregard for developing secure software systems from the viewpoint of developers. Software security was formerly applied only after the system was finished. System developers fixed the security flaws when they were found, usually by an attacker. Numerous security failures in the business demonstrate the ineffectiveness of this philosophy, also referred to as the penetrate-and-patch method [1].

Software engineers are becoming increasingly concerned about security vulnerabilities in their software as a result of the growth of computer networks and the growth of Internet-based services. The cause is the lack of effort spent evaluating these features, which results in the distribution of software that is insecure to all users. Model-based testing is becoming more and more common as a solution to this issue. A number of the works that offer standards to model different elements are related to security characteristics [2].

Numerous software modeling-related quality criteria, including modularity, testability, modifiability, reusability, security, and others, have been documented in the literature. [3] [4]. Because software systems are so vital these days, security has emerged as one of the most significant quality criteria. The wealth of recent literature on secure software development further supports this [5].

Based on previous study, a number of research questions were discussed, the fundamental questions in this research are as follows:

1. What are the main approaches that can be utilized for security specification, and how can UML diagram security be applied?
2. How can security specifications be applied to each approach?
3. Do these schemes use the majority of security algorithms?
4. What are the potential specification techniques for a particular security requirement, and if any, which is the best one?

This work is arranged as follows. Sections 2, 3, and 4 provide an overview UML diagram and the importance of their use by researchers at every stage of software development, as well as the importance of achieving the principle of security in these stages, as well as a list of previous studies that addressed these topics, and finally conclusions and future studies.

II. UNIFIED MODELING LANGUAGE

The Unified Modeling Language is a general-purpose, standardized modeling language used in "object-oriented software engineering". UML provides a set of graphic notation tools to rapidly create visual models of object-oriented

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 5, pp 213-222, May-2025

https://doi.org/10.47001/IRJIET/2025.905028

software systems [6]. Throughout the software development life-cycle and across many implementation technologies, UML integrates methods from data modeling, object modeling, business modeling and component modeling. [7][8].

Software engineering also uses UML, a visual language for defining systems, documentation and development. The UML display the limits of a system as well as the requirements in scenarios that show how users interact with it [9]. In order to further the field of software engineering, Papers concerning the application of UML diagrams in system development are frequently written by scholars who are also software engineers and used diagrams to achieve security requirements during the stages of software engineering.

The Use Case Diagram identifies interactions between users and the system, outlining critical scenarios such as authentication, monitoring, and alerting. Activity Diagrams depict the workflow of security processes, detailing the steps involved in motion detection and incident response. Class Diagrams represent the structure of the system by defining objects, their attributes, and relationships. Sequence Diagrams illustrate the temporal interactions among components during specific events, while Component Diagrams provide an overview of system architecture. Finally, Deployment Diagrams show how the system components are distributed across physical devices [10] [11]. This comprehensive approach enables stakeholders to understand the security system's design, facilitates effective communication among developers, and supports system implementation and maintenance. Ultimately, the use of UML diagrams in designing security systems enhances clarity, efficiency, and effectiveness in achieving security objectives [12]. UML is a powerful language for modeling business processes, system behavior, application structures, and software solutions. Structure diagrams and behavioral diagrams are the two primary categories [8] [13].

1. Structural (or Static): display the objects, operations, attributes and relationships to highlight the system's static structure. Both composite structure diagrams and class diagrams are included.
2. Behavioral (or Dynamic): display aspect the system's dynamic behavior by demonstrating object cooperation and alterations to objects state, activity diagrams, state machine diagrams and Sequence diagrams are things contained in this aspect.

Structural and Behavior diagrams are comprised of different types of the UML diagrams as shown in Table (1) and Table (2).

Table 1: Structural Diagrams of UML

Diagram Type	Detail
Class	Class diagrams capture the logical structure of the system, the Classes and objects that make up the model, describing what exists and what attributes and behavior it has.
Composite Structure	Composite Structure diagrams reflect the internal collaboration of Classes, Interfaces and Components (and their properties) to describe a functionality.
Component	Component diagrams illustrate the pieces of software, embedded controllers and such that make up a system, and their organization and dependencies.
Deployment	Deployment diagrams show how and where the system is to be deployed; that is, its execution architecture.
Object	Object diagrams depict object instances of Classes and their relationships at a point in time.
Package	Package diagrams depict the organization of model elements into Packages and the dependencies amongst them.
Profile	Profile diagrams are those created in a «profile» Package, to extend UML elements, connectors and components.

Table 2: Behavioral Diagrams of UML

Diagram Type	Detail
Activity Diagrams	Activity diagrams model the behaviors of a system, and the way in which these behaviors are related in an overall flow of the system.
Use Case Diagrams	Use Case diagrams capture Use Cases and relationships among Actors and the system; they describes the functional requirements of the system, the manner in which external operators interact at the system boundary, and the response of the system.
StateMachine Diagrams	StateMachine diagrams illustrate how an element can move between states, classifying its behavior according to transition triggers and constraining guards.
Timing Diagrams	Timing diagrams define the behavior of different objects within a time-scale, providing a visual representation of objects changing state and interacting over time.
Sequence Diagrams	Sequence diagrams are structured representations of behavior as a series of sequential steps over time. They are used to depict workflow, Message passing and how elements in general cooperate over time to achieve a result.
Communication Diagrams	Communication diagrams show the interactions between elements at run-time, visualizing inter-object relationships.
Interaction Overview Diagrams	Interaction Overview diagrams visualize the cooperation between Interaction diagrams (Timing, Sequence, Communication and other Interaction Overview diagrams) to illustrate a control flow serving an encompassing purpose.

Figure (1) shows the importance of diagrams and their use by researchers in modeling large systems and sometimes critical systems in addition to systems that require security to be achieved and taken into consideration from the early stages of this software.

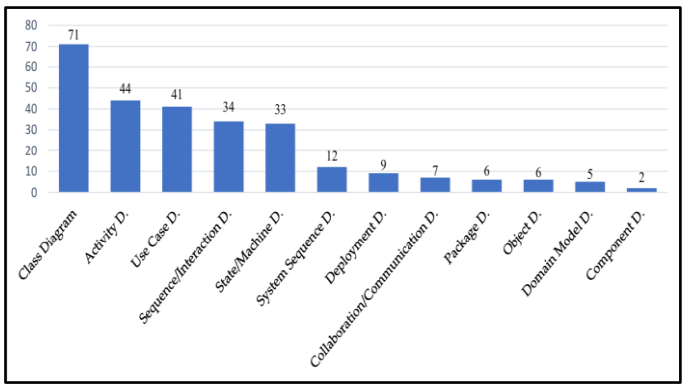

Figure 1: UML diagram usage in publications [10]

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 5, pp 213-222, May-2025

https://doi.org/10.47001/IRJIET/2025.905028

Planning is done via UML diagrams and constructs systems using the Object-Oriented paradigm; they enable for a thorough study and comprehension of the system's operational features, workings, implementation specifics, and architecture since they allow for several system aspect viewpoints [16]. Security requirements are known to exist. At the intersection of software engineering and security, engineering is a new field. The understanding that security must be addressed in the early stages of the procedure for developing software because these stages encompass a wider arranging viewpoint has led to a lot of study in this area in recent years. [17]. In this context, agent-oriented approaches have proven particularly helpful since they facilitate The social framework in which the future system might operate is modeled [18].

The completion of a system safety/security environment analysis requires knowledge, understanding, and modeling of many aspects of the system's operation, data flow, data types, architecture, and implementation details in order to identify potential weak points for the system's safety or security based on the application of the system. All the information required for a safety/security system analysis is provided by the numerous UML diagrams, and many elements of the UML approach can be used for the same objective. [16].

III. LITERATURE REVIEW

In this section reviews the most important research that has dealt with scurity System in UML Diagrams as following:

In 2002, Torsten Lodderstedt, et al [19] presented a modeling language based on “Unified Modeling Language (UML)” enabling the model-driven creation of distributed, safe systems. This method can be used to increase efficiency while developing secure distributed systems and the caliber of the final systems. With additional capability for specifying permission requirements, it is predicated on role-based access control.

Also, in 2002, Jan Jurjens [20] support the challenging process of creating security-critical systems using a methodology based on the Unified Modeling Language's notation. He introduced the UMLsec extension, which enables the expression of security-related information in system specification diagrams. UMLsec is a UML profile defined using standard UML extension procedures.

In 2003, Siv et al, [21] presented Security evaluate UML, the UML profile designed for security assessments that are based on models. Supporting documentation of the outcomes of a security assessment's risk identification and analysis is the primary goal of Security Assessment UML. The profile specifically allows for the creation of threat scenarios that illustrate potential attack scenarios and fault tree-inspired activity diagrams that analyze attack frequency.

And in 2004, Jurjens [22] presented the UMLsec is a UML extension that enables the expression of security-related data inside system specification diagrams. According to standard UML extension methods, UMLsec is a UML profile. By using an official semantics of a condensed portion of UML, the related constraints provide criteria to assess a system design's security features. He explains how automated theorem provers for first-order logic can be used to explicitly verify these restrictions against the specification's dynamic behavior. C code produced from the specifications can likewise be subject to this formal security verification.

In 2006, Zhitang Li, et al [23] presented a structure for modeling security protocol, referencing an accurately specified semantics of behavioral features, the UML extension for security protocol (USP) enables the creation of security procedures in a user-friendly and visible manner. UML State Machines' dynamic semantics is the foundation for the official semantics of a condensed version of UML. Using the Denning-Sacco protocol as an example, USP.

In 2006, Matthew J. Peterson, et al [1] presented UMLpac, an extension of UML that fills the gap between the design of software classes and the security methods needed for those designs. By maintaining a degree of abstraction between the system class diagram and its security features, security packages achieve this objective. This design approach maintains comprehensive security measures for every part of the system while preserving the original system diagram.

In 2007, D.C. Petriu et al [24] focused on the analysis of performance effects of different security solutions modeled as aspects in UML. This paper's contribution is the composition of the aspects at the CSM(Core Scenario Model) level using the primary model. The core model and several shape models in UML+SPT provide the input, which is then processed in the manner described below:

1. Translated to CSM individually.
2. Combined into a one CSM model.
3. Translated into a “Layered Queueing Networks (LQN)” model.
4. Examined. A case study based on the TPC-W (a transactional web benchmark of the Transaction Processing Performance Council) and SSL standards serves as an example of the suggested methodology.

In 2008, Nina Moebius , et al [25] presented method known as Secure MDD, was proposed to model security-critical systems. They used unified modeling language (UML) with a UML profile added to customize their models for

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 5, pp 213-222, May-2025

https://doi.org/10.47001/IRJIET/2025.905028

applications in security from the model; they automatically provide an implementation and a formal specification that can be verified. They provided a model-driven development approach that smoothly combined formal and semi-formal approaches with implementation. They use the Mondex electronic payment system as an example of their methodology.

In 2009, Karine P. Peralta, et al [2] presented a method for defining UML security stereotypes with the goal of assisting developers by highlighting weak model components and enabling the creation of security test cases automatically. They have established several archetypes to depict the most prevalent security problems depend on Weber's classification system and the list of vulnerabilities supplied by the OWASP project.

In 2010, JUN KONG, et al [26] presented a UML-depend structure for visually and systematically modeling and assessing security threats, or potential attacks on security. They used UML state chart model to outline the desired functionalities of a software application and Using sequence diagrams to simulate the dangers to security. An established theoretical paradigm for graph transformation is automatically applied to state chart diagrams. A series of transformations of paired graphs is how method invitation is processed in a security threat's sequence diagram. Simulating the state covert ions that are brought about by method invocations from an initial state to a final one is thus how a security threat is analyzed.

In 2011, VipinSaxena, et al [27] provided a UML model for a credit/debit card-safe online banking system, demonstrating how to apply the model to improve security. UML class, activity, and sequence diagrams are created, and the case study is used for validation.

Also in 2011, Sandra Smith, et al [28] offered a modeling technique that supports the planning, development, and validation of security measures using UML 2 without extensions. With the exception of communications that are shielded by known-strong encryption, the method is predicated on a powerful threat model, which allows an assailant to intercept, modify, and spoof all conversations. They proved that the method enables accurate representation of protocol attributes and automatic testing of protocols to identify possible security vulnerabilities through a number of models of widely researched protocols. The method makes use of UML 2's robust tool support, which enables the models to automatically generate protocol implementations.

In 2014, S. Batool and S. Asghar [29] provided a method for converting extended UML models to common UML models in order for these models to be immediately subjected to the use of current MBT approaches. Because of the differences in modeling notation and new model elements, the "Model Based Testing (MBT)" methods now in use cannot be easily used to expanded UML models. It is also crucial to verify these models. The model's capacity to lower risk, expense, and the likelihood of system weakness in an economical manner is strengthened by the realization and testing of nonfunctional needs such as model-level security, portability, and efficiency. They discovered that by using the suggested technique on case studies, MBT techniques may be used on the created test pathways and consequent state machine diagrams, which may help detect the ability to recognize the dangers associated with attacks involving security limitations.

In 2015, MOHAMED EL-ATTAR, et al [12] proposed a new set of notation that expands on the state-charts notation of the UML. The perceptions of the new notation on its coverage of modeling state-based security elements and its semantic clarity were assessed by an online industry survey. To assess the suggested notation's cognitive efficacy, an empirical study that is subject-based empirical examination employing software phases experts was also carried out. Because the subjects were able to read models made with the new notation considerably more quickly, the main conclusion was that the new notation outperforms the old set of UML state charts in terms of cognitive efficiency.

In 2017, David Alejandro Robles-Ramirez [30] introduced a new UML extension that includes UML notation extensions, UML stereotypes to model common actors, and security issues contained inside a nomenclature. All of this is done to give IoT developers, including those who are not quite conversant with cybersecurity principles, a helpful notation to represent security in IoT systems from the beginning of their design.

In 2018, Lasbahani and Tabyaoui[31] By creating a general approach for integrating security and creating code that considers security requirements into account throughout the system development cycle, they focused their work on non-functional components such as data flow monitoring, the business logic layer, and offering top-notch services. In order to improve the "Platform Independent Model (PIM)" source model elements with guidelines for security integration for the validation and verification of security guidelines, as well as to create the A new UML profile for security integration and code generation for Java platforms has been presented, which maps annotated items to the associated stereotypes. This is done in a practical way while switching from CIM to PIM. Enhancing software application interoperability, productivity, and generalizing integration of security across the board is the goal.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 9, Issue 5, pp 213-222, May-2025

https://doi.org/10.47001/IRJIET/2025.905028

In 2020, Mohammad Alshayeb, et al[5] examined the use of model refactoring to address the security issue in a behavioral model (sequence diagram). They suggested methods for detection and correction, evaluated the suggested methods empirically, and assessed the increase in sequence diagram security. The adaption of a genetic algorithm is used to discover security poor odors, and the model transformation approach is used to repair them. The findings demonstrate that the suggested method can enhance UML Sequence Diagram security and is successful in identifying and eliminating bad smells.

In 2021, PONCIANO JORGE ESCAMILL A-AM B ROSIO, et al [32] suggested a method known as IoTsecM. This proposal is a Model-Based Systems engineering approach that extends UML/SysML for modeling security needs throughout the analysis phase of a waterfall development life cycle. IoTsecM enables the depiction of security needs in UML and SysML, two highly used modeling languages. When designing IoT systems, developers can take security concerns into account from the analysis stage. IoTsecM makes it feasible to build IoT systems with potential risks and the associated security requirements analysis in mind. The security needs found in the system design parts may be represented by IoTsecM, and the suggested IoTsecM profile was used to illustrate all of the countermeasures found.

In 2022, Hind Meziane and Noura Ouerdi [33] centered on IoT system security modeling. By outlining the IoT design and showcasing its constituent parts, the objective is to create a layer-by-layer arbitrage. Stated differently, the study's objective is to identify security modeling within the layers of the Internet of Things. Since there are various IoT level plans and no standard that considers the security of the IoT architecture, each author has his or her own idea and suggestion. Furthermore, modeling languages for IoT security systems are scarce. Selecting the layer that we should be interested in is the primary goal of this investigation. Therefore, "which is the layer whose modeling is relevant?" is the query. The results were definitive and offered the most comprehensive understanding of all the details of every layer of the IoT architecture under study.

In 2024, Tracy Tam, et al [34] suggested a new UML model of class called “Small IT Data (SITD)” to help with the frequently disorganized information collection stage of Initially, a small business's cybersecurity venture. To assist small businesses in implementing technology solutions, the SITD model was created in the UML style. By utilizing generic classes and structures that adapt to changes in the environment and in technology, the SITD model structure remains current. By emphasizing the connections between business strategy tasks and IT infrastructure, the SITD model maintains security decisions in line with the business. They developed a set of design guidelines to meet the cyber security requirements of small businesses. These requirements inform the design of the model's components. The NotPetya event is also used to demonstrate how the SITD model can depict breach information.

Table 3: Structural Diagrams of UML

No	Researchers	Year	Security Methodology / Technique used	UML Modeling	Objective	Outcomes
1.	Torsten Lodderstedt, et al	2002 [19]	A prototypical generator for EJB.	Class diagram	Integrating the specification of access control into application models	Improve productivity during the development of secure distributed systems
2.	Jan Jurjens	2002 [20]	UMLsec	UML diagrams	Developing security-critical systems	Express security relevant information within the diagrams
3.	Siv , et al	2003 [21]	Security Assessment UML,	Sequence, activity diagram	Support documentation of output from risk in a security assessment	Specifying concrete threat scenarios demonstrating the relationship between undesired events, their frequencies, consequences and impacts
4.	Jan Jurjens	2004 [22]	UML sec	UML diagrams	Assist in the challenging process of creating security-critical systems using a formal methodology.	Use UMLsec to model security requirements, threat scenarios, security concepts, security mechanisms, security primitives, underlying physical security, and security management

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Issue 5, pp 213-222, May-2025
https://doi.org/10.47001/IRJIET/2025.905028

5.	Zhitang Li, et al	2006 [23]	Denning-Sacco protocol with USP	Sequencs diagram	Enables the creation of security protocols in a clear and understandable manner.	Standard notions from formal techniques pertaining to security protocols can be expressed using UML's extension mechanisms, which facilitate the construction of security protocols in an understandable and visible manner.
6.	Matthew, et al	2006 [1]	UMLpac , extension of UML	Class diagram	Closes the gap between the security methods needed for software class design and the design itself.	Keeps all of the system's security features intact while preserving the original system diagram.
7.	D.C. Petriu, et al	2007 [24]	AOM	UML diagram	Utilizing the fundamental model to compose the components at the CSM level.	Enhance a system with security solution
8.	Nina Moebius et al	2008 [25]	SecureMDD	UML diagrams	Use a UML profile to augment the unified modeling language (UML) for modeling security-critical applications	Demonstrate the validity of the Java protocol implementation with respect to the abstract protocol level security features and formal model.
9.	Karine P. Peralta, et al	2009 [2]	Security stereotypes	Use case Model	Help developers by marking areas of the model that are vulnerable and enabling the creation of security test cases automatically.	Representing security considerations from the beginning of the design process and helping developers steer clear of flaws.
10	JUN KONG, et al	2010 [26]	aUML-based framework	Sequence, state-chart diagrams	Thoroughly and graphically modeling and assessing security threats, or possible security assaults	UML diagrams are used directly by designers to visually represent system behaviors and security risks.
11	VipinSaxena, et al	2011 [27]	RSA algorithm,	Class, Sequence, activity diagrams	Enhance the security level.	Enhance the security level.
12	Sandra Smith, et al	2011 [28]	UML 2 with not extensions		To bolster the composition and design and verification of security protocols. to simulate the Needham-Schroeder and Yahalom protocols as well as the simple protocols without the need for language extensions	Allowing automatic generation of protocol implementations from the models.
13	S. Batool, at al	2014 [29]	MBT techniques	State machine diagram	To determine the dangers of breaking security restrictions.	Applying MBT approaches to created test pathways and consequent state machine diagrams may help detect the risks related to security constraint violations.
14	mohamed el-attar, et al	2015 [12]	An expansion of the UML statecharts	statechart	To suggest a new set of notation that expands on the notation used for	Because the respondents were able to interpret models made with the new notation considerably more

ISSN (online): 2581-3048
Volume 9, Issue 5, pp 213-222, May-2025
https://doi.org/10.47001/IRJIET/2025.905028

			notation using a new set of symbols		UML statecharts.	quickly, the new notation is cognitively more effective than the original set of UML statecharts.
15	David Alejandro Robles-Ramirez, et al	2017[30]	IoTsec	UML Diagrams	Giving IoT developers, especially those who are not fully conversant with cybersecurity ideas, a helpful notation to model security in IoT systems from their designing stage.	Includes UML notation extensions, UML stereotypes to describe common actors, and security concerns contained inside a nomenclature.
16	A. Lasbahani, at al	2018 [31]	SDSIB	sequence diagram	Boost applications software interoperability, Integration of security and productivity throughout the whole software development life cycle as opposed to only the implementation stage.	To get past non-functional elements at every stage of the software development process. to produce safe apps that adhere to the new LOC design and semantic guidelines. Additionally, during the software development process, it has been feasible to integrate both "functional and non-functional" elements.
17	Mohammad Alshayeb, et al	2020 [5]	-Bad smell detection -GA	Sequence diagram	Examine the security issue in a behavioral model (sequence diagram) by applying model refactoring and assessing security enhancements.	Effective at identifying and fixing bad smells, it can also increase the UML Sequence Diagram's security
18	PONCIANO JORGE, et al	2021 [32]	IoTsecM	UML diagrams	Enables the design of IoT systems to take potential threats and the study of related security requirements into account.	Reflect the security needs found in the components of the system design, helps to understand and consider the security of IoT systems during their design stage before they are implemented in physical objects.
19	Hind Meziane, et al	2022 [33]	IoTsec	Class diagram	To present the components of the IoT architecture and describe it in order to compare them in terms of layers..	To determine which layer of the model is most pertinent to IoT system security.
20	Tracy Tam, at al	2024 [34]	SITD Model	Class diagram	To aid in the frequently disorganized phase of a small business's first cyber-security endeavor that involves acquiring information.	Helps small businesses overcome some of the obstacles they have while utilizing the technologies that are currently available and comprehending security procedures that are centered on cyber security.

IV. CONCLUSION

It is possible to design software to operate securely throughout development. This entails creating the software with security requirements in mind. Typically, security-related information, such as the presence of specific security measures, is marked solely on the high-level design. Nonetheless, security requirements impose limitations on the software's functionality. These limitations are required to meet the security requirements.

This work reviews the research designed on a security system using Unified Modeling Language (UML) diagrams, providing a structured approach to visualizing and documenting system requirements and interactions. The security system encompasses various components such as user interfaces, sensors, cameras, and alert mechanisms. Key UML

ISSN (online): 2581-3048
Volume 9, Issue 5, pp 213-222, May-2025
https://doi.org/10.47001/IRJIET/2025.905028

diagrams, including Use Case, Class, Activity, Sequence, Component, and Deployment diagrams, are employed to illustrate the functionality and architecture of the system. It has been unable to locate a single study that addressed all of these factors and gave UML designers the anticipated responses. Our work's primary contributions are:

1. Examining the state of the art about UML security requirements and determining the primary strategies implemented.
2. Explaining the applications of each strategy for security requirements.
3. Outlining security requirements and utilizing them to talk about how usable each strategy is. The most recent developments in software security requirements definition served as the model for the set of security criteria.

In future work, can Create UML diagrams for each security specification and put them in an open-access repository that developers and designers may use at any time.

REFERENCES

[1] M. J. Peterson, J. B. Bowles and C. M. Eastman, "UMLpac: An Approach for Integrating Security into UML Class Design", Proceedings of the IEEE Southeast Con 2006, Memphis, TN, USA, pp. 267-272, doi:10.1109/second.2006.1629362, 2006.

[2] A.F. Zorzo, F.M.d. Oliveira, “Specifying security aspects in uml models”, Lecture Notes in Computer Science, Springer, 2009.

I. Gorton, “Essential software architecture”, Springer Science & Business Media, 2006.

[3] ISO/IEC 25010:2011: Systems and software engineering - Systems and software Quality Requirements and Evaluation 2011.

[4] M. Alshayeb, H. Mumtaz, S. Mahmood and M. Niazi, "Improving the Security of UML Sequence Diagram Using Genetic Algorithm," in IEEE Access, vol. 8, pp. 62738-62761, doi: 10.1109/ACCESS.2020.2981742, 2020.

[5] B. Bhatt1, M. Nandu2. “An Overview of Structural UML Diagrams”, International Research Journal of Engineering and Technology (IRJET). Volume: 08 Issue: 08. e-ISSN: 2395-0056, p-ISSN: 2395-0072, Aug 2021.

[6] L. B. R. dos Santos, V. A. de Santiago Junior, and L. N. Vijaykumar, “Transformation of UML Behavioral Diagrams to Support Software Model Checking,” Electron. Proc. Theor. Comput. Sci., vol. 147, pp. 133–142, Apr. 2014.

A. Metzner, "Systematic Teaching of UML and Behavioral Diagrams," 2024 36th International Conference on Software Engineering Education and Training (CSEE&T), Würzburg, Germany, 2024, pp. 1-5, doi:10.1109/CSEET62301.2024.10663036.

B. Kitchenham, "Procedures for performing systematic reviews," Keele, UK, Keele University, vol. 33, no. 2004, pp. 1-26, 2004.

[7] H. Koç, A. M. Erdoğan, Y. Barjakly, and S. Peker, “UML diagrams in software engineering research: A systematic literature review,” Proceedings, vol. 74, no. 1, Mar. 2021, Art. no. 13.https://doi.org/10.3390/proceedings2021074013.

[8] P.Forbrig, “Objektorientierte Softwareentwicklungmit UML”, Carl Hanser, 2024.

[9] M. El-Attar, H. Luqman, P. Kárpáti, G. Sindre and A. L. Opdahl, "Extending the UML Statecharts Notation to Model Security Aspects," in IEEE Transactions on Software Engineering, vol. 41, no. 7, pp. 661-690, 1 July 2015, doi:10.1109/TSE.2015.2396526.

[10] F. Ciccozzi, I. Malavolta, B. Selic, "Execution of UML models: a systematic review of research and practice", Software & Systems Modeling, Vol. 18, pp. 2313–2360, 2019.DOI: 10.1007/s10270-018-0675-4.

[11] J. Jürjens, “Secure Systems Development with UML”, Springer-Verlag. 2004.

[12] D. Mouheb, C. Talhi, V. Lima, M. Debbabi, L. Wang, and M. Pourzandi, “Weaving security aspects into UML 2.0 design models”, In Proceedings of the 13th workshop on Aspect-oriented modeling, pages 7–12, Charlottesville, Virginia, USA, 2009, ACM.

[13] F. M. Rachel & P. S. Cugnasca, "Using UML diagrams for system safety and security environment analysis. "International conference on computer aided design, manufacture, and operation in the railway and other advanced mass transit systems; Computers in railways X, vol. 88, pp. 319-328, WIT, 2006.

[14] N. Zannone, “A Requirements Engineering Methodology for Trust, Security, and Privacy”, Ph.D. thesis, University of Trento, 2007.

[15] F. Massacci, J. Mylopoulos, and N. Zannone, “Security requirements engineering:Thesi* modeling language and the secure tropos methodology,” Advances in Intelligent Information Systems, pp. 147–174, 2010.

[16] T. Lodderstedt, D. Basin, and J. Doser, “SecureUML: A UML-based modeling language for model-driven security”, In: International Conference on the Unified Modeling Language, pages 426–441, 2002, https: //doi.org/10.1007/3-540-45800-X_33.

[17] J. Jürjens, “UMLsec: Extending UML for secure systems development. In: UML”, The Unified Modeling Language, 5th International Conference,

ISSN (online): 2581-3048
Volume 9, Issue 5, pp 213-222, May-2025
https://doi.org/10.47001/IRJIET/2025.905028

Dresden, Germany, September 30 - October 4, vol. 2460, pp. 412– 425, 2002.

[18] S. Houmb, K. Hansen, Towards a UML profile for Security Assessment, in: Work. on Critical Systems Development with UML, pp. 815–829, 2003.

[19] J. Jürjens, "Model-based security engineering with UML", International School on Foundations of Security Analysis and Design. Berlin, Heidelberg, p. 42-77, 2004, https://doi.org/10.1007/11554578_2.

[20] Zhitang Li et al, "USP: Modeling security protocol with UML", Network Architectures, Management, and Applications IV, Vol. 6354, SPIE, 2006, https://doi.org/10.1117/12.689087.

[21] D. C. Petriu, C. M. Woodside, D. B. Petriu et al., "Performance analysis of security aspects in UML models", Proceedings of the 6th International Workshop on Software and Performance (WOPS '07), pp. 91-102, February 2007.

[22] N. Moebius, W. Reif, and K. Stenzel, "Modeling security-critical applications with UML in the Secure MDD approach," International Journal On Advances in Software, vol. 1, no. 1, pp. 59–79, 2009.

[23] J. Kong, D. Xu, and X. Zeng, "Uml-Based Modeling and Analysis of Security Threats," International Journal of Software Engineering and Knowledge Engineering, vol. 20, no. 06, pp. 875–897, Sep. 2010.

[24] V. Saxena and Ansari, G.A., Ajay Pratap, "Enhancing Security through UML", International Journal of Computer Sciences, Software Engineering and Electrical Communication Engineering, Vol. 2(1), pp. 31-36, June 2011.

[25] S. Smith, A. Beaulieu and W. G. Phillips, "Modeling and verifying security protocols using UML 2", 2011 IEEE International Systems Conference, Montreal, QC, Canada, 2011, pp. 72-79, doi:10.1109/SYSCON.2011.5929088.

[26] S. Batool and S. Asghar, "Secure State UML: Modeling and Testing Security Concerns of Software Systems Using UML State Machine", Research Journal of Applied Sciences, Engineering and Technology 7(18): 3786- 3790, 2014.

[27] D. A. Robles-Ramirez, P. J. Escamilla-Ambrosio and T. Tryfonas, "IoTsec: UML Extension for Internet of Things Systems Security Modelling", 2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), Cuernavaca, Mexico, pp. 151-156, 2017, doi: 10.1109/ICMEAE.2017.20.

[28] A.Lasbahani, M. Chhiba, A. Tabyaoui, "A Model Transformation Methodology for Security Integration and Code Generation from Sequence Diagram of System's Internal Behavior", International Review on Modelling and Simulations (IREMOS), vol. 11, n. 2, pp. 102-116, apr. 2018, ISSN 2533-1701.

[29] P. J. Escamilla-Ambrosio, D. A. Robles-Ramírez, T. Tryfonas, A. Rodríguez-Mota, G. Gallegos-García and M. Salinas-Rosales, "IoTsecM: A UML/SysML Extension for Internet of Things Security Modeling", in IEEE Access, vol. 9, pp. 154112-154135, 2021, doi: 10.1109/ACCESS.2021.3125979.

[30] H. Meziane and N. Ouerdi, "A Study of ModellingIoT Security Systems with Unified Modelling Language (UML)" International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 13, No. 11, 2022, http://dx.doi.org/10.14569/IJACSA.2022.0131130.

[31] T. Tam, A. Rao, J. Hall, "Structuring the Chaos: Enabling Small Business Cyber-Security Risks & Assets Modelling with a UML Class Model", arXiv preprint arXiv: 2403.14872. 2024.

AUTHORS BIOGRAPHY

ASMAA H. THANOON was born in Mosul, Nineveh, Iraq, in 1977. She received the B.S. and M.S. degrees in software engineering from Mosul University, in 2007 and 2013, respectively, from 2013 to 2025, she was taught software engineering and techniques in the College of Computer Science and Mathematics, University of Mosul. She has many researches in a field software engineering.

Taghreed Riyadh Alreffaee was born in Mosul, Nineveh, Iraq, in 1984. She received the B.S. and M.S. degrees in software engineering from Mosul University, in 2006 and 2017, respectively, from 2018 to 2025, she was taught software engineering and techniques in the College of Computer Science and Mathematics, University of Mosul. She has many researches in a field software engineering.

ISSN (online): 2581-3048
Volume 9, Issue 5, pp 213-222, May-2025
https://doi.org/10.47001/IRJIET/2025.905028

ANFAL A. FADHIL was born in Mosul, Nineveh, Iraq, in 1983. She received the B.S. and M.S. degrees in software engineering from Mosul University, in 2006 and 2012, respectively, from 2012 to 2025, she was taught software engineering and techniques in the College of Computer Science and Mathematics, University of Mosul. She has many researches in a field software engineering.

Citation of this Article:

Asmaa Hadi Albayati, Taghreed Riyadh Alreffaee, & Anfal A. Fadhil. (2025). Security System in UML Diagrams: A Literature Review. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 9(5), 213-222. Article DOI https://doi.org/10.47001/IRJIET/2025.905028

****

