"Mircea cel Batran" Naval Academy Scientific Bulletin, Volume XIX – 2016 – Issue 2
The journal is indexed in: PROQUEST / DOAJ / Crossref / EBSCOhost / INDEX COPERNICUS / DRJI / OAJI / JOURNAL INDEX / I2OR / SCIENCE LIBRARY INDEX / Google Scholar / Academic Keys/ ROAD Open Access / Academic Resources / Scientific Indexing Services / SCIPIO / JIFACTOR

ENHANCING UML WITH SECURITY

Marius Iulian MIHAILESCU[1]
Adrian BETERINGHE[2]
Violeta OPRIS[3]

[1]Department of IT&C, LUMINA – The University of South-East Europe
[2]Department of IT&C, LUMINA – The University of South-East Europe
[3]Military Technical Academy

Abstract: *Unified Modeling Language (UML) is very used in different companies and industries where the process of software analysis plays an important role. Still, the UML has different lacks, such as formal, explicit, support for access control. The security represents an important issue over which we have to stop when designing the access control into the application. In this paper we will discuss about a new approach for expressing security-relevant information that can be mapped in the UML diagrams, such as sequence diagrams, class diagrams and state diagrams. New diagrams that already have been proposed will be shown and presented in a practical manner, such role-based, discretionary and mandatory access controls. The intent of the paper is to give the designers with a set of security and integrity features. Only the necessary features are selected for the application that is designed and furthermore implemented.*

Keywords*: security, UML, integrity, software analysis.*

Introduction

Security is a critical issue in the advancement of programming applications. Definition procedure of access control arrangements, together with other security prerequisites, must be an installing part of the product advancement process, keeping in mind the end goal to guarantee that the best possible level of security in an application is gotten. Numerous meanings of access control exist, however we stop on the one definition which is nearer to the truth: "Constraining access to data framework assets just too approved clients, projects, forms or different frameworks" [1]. Programming improvement process comprises in a deliberate arrangement of undertakings used to make a product framework: necessities catch, examination, outline, coding, and testing. The objective of the paper is at the outline phase of the procedure, focusing on demonstrating of access control.

To dissect the issues of demonstrating security, one must comprehend the most well-known security conspires that are utilized to conceptualize access control approaches: required access control (MAC) [2], optional access control (DAC) [3], and part based access control (RBAC) [4].MAC is appropriate to applications where the assurance of data is foremost (i.e., discharging such data would have desperate national security or money related results). In MAC, every object is tagged with an arrangement level (e.g., top mystery, mystery, classified, and unclassified) that focus on the affectability of their data. Every subject has a leeway level. Security is implemented by guaranteeing that a subject's leeway level dependably overwhelms an item's characterization level. DAC targets applications that are communitarian and element. In DAC, consents are characterized amongst subjects and questions, yet a subject can be conceded the authorization to assign its very own subset authorizations to another client. RBAC gathering's consents into free units called parts, which speak to the part that a client expect in an association.

Different parts, instead of authorizations, are relegated to clients (subjects) when they start an intelligent session with the product framework. The arrangement of benefits that are allowed to a client is characterized by the arrangement of authorizations doled out to its relating part. Security plans, for example, MAC, DAC, and RBAC, determine the essential semantics for access control, yet they don't give a visual dialect to speak to this data.UML [5], the overwhelming programming and framework demonstrating approach, while an undeniable contender to give security, needs express backing for access control. Besides, security is a crosscutting worry that swarms the whole application, which makes it troublesome for programming experts to enough coordinate security into programming [6].Accordingly, when planners wish to join security worries into an application utilizing UML, the subsequent model is liable to have security tangled and scattered all through the whole outline.

Our proposed methodology will address the above issues by stretching out UML with security outlines to speak to MAC, DAC and RBAC

DOI: 10.21279/1454-864X-16-I2-070

arrangements as perspectives. Besides, the proposed approach means to give adaptability to the displaying of access control: as prerequisites shift between applications, planners don't generally require the majority of the elements present in the documentation, yet just a subset of them to suit their application needs. The methodology breaks down MAC, DAC, and RBAC into security highlights, which speak to the negligible components of an entrance control strategy. Planners can choose particular elements and join them (as per principles and breaking points) keeping in mind the end goal to make a security angle displaying framework that is appropriate for their necessities. Since security highlights include a little subset of the data of an entrance control mapping, they ought to be simpler to comprehend by architects. Moreover, they help with following security necessities from models to code, lessening scrambling of access-control definitions over the application, and giving an aggregate perspective of the security approach.

This paper develops the works of in regards to the part cut outline for RBAC [7] with extra charts for clients, appointment, and required access control highlights.

In particular, this work applies composability to permit custom application level security. Area 2 portrays a case that will be utilized to outline the methodology. We will point out the security highlights and the procedure to make custom security viewpoint models. Later we will show some contrasts in the proposed approach and related work.

Case Study: The University System

The contextual analysis introduced beneath speaks to a surely understood case on which we will make our talk keeping in mind within the final goal which consist in accomplishing the objective of the paper.

The university application oversees course, understudy, educator, and open list data.

The security prerequisites are as per the following: educators have relegated an arrangement of courses, they can read and compose the syllabus, and read the code of every course. Instructors can see the selected understudies in every course, get to their names, and allot grades, however they can't find in which courses understudies are enlisted. Understudies can see their evaluations, selected courses, the educators of those courses, read the syllabus and code, however can't see which understudies are enlisted in those courses, or adjust any data in the framework. Record information can be gotten to by anyone; no passage control is required for this data.Figure 1 demonstrates a class chart of the college application. Course monitors the greater part of the courses of a college.

Student Information oversees data about understudies. List deals with the openly accessible data about courses offered at a college.

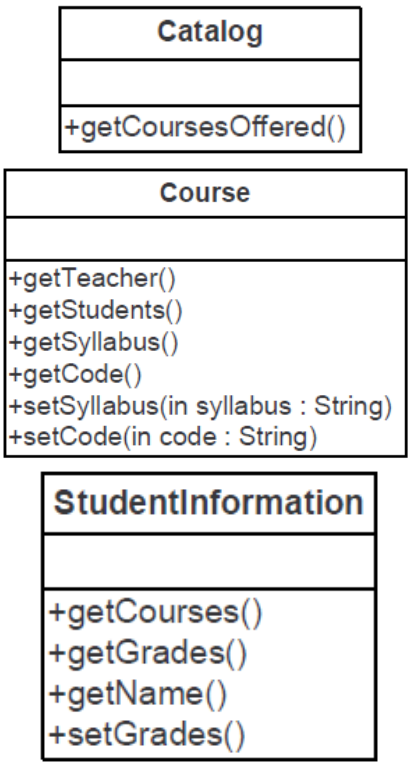

Fig. 1. The university case study class diagram

3. Increasing Security with UML

The center of the methodology is to develop UML with security angle demonstrating abilities. The expansion involves two components: Security Features and Security Diagrams. Security Features are segments that compare to particular components of access control plans (e.g., positive consents, assignment rules, MAC security properties, and so forth). Security Diagrams give the documentation to delineate security highlights as viewpoints isolated from the fundamental configuration of the application.

Figure 2 demonstrates a review of the proposed security augmentations to UML. The Role Slice Diagram (1), which is a piece of past work in [7], is a visual documentation for parts, positive and negative consents (e.g., parts to techniques), and part pecking orders. The User Diagram (2) delineates clients, positive and negative consents (parts to clients), relationship to parts, and limitations over part task. The Delegation Diagram (4) is a documentation for guidelines of designation of the DAC security plot, and incorporates client appointment task (who is

DOI: 10.21279/1454-864X-16-I2-070

permitted to delegate), assignment power (can delegate), and go on assignment power (can appoint the capacity to assign). Macintosh Features (3) give the builds to the three security charts to delineate Mandatory Access Control rules. Every augmentation is connected with an arrangement of Security Features, which are building obstructs that relate to particular components of access control plans (e.g., parts, consents, assignment rules, orders, clearances, and so on.). Originators pick a subset of elements, and perform a creation (6) between their meta-models and the UML meta-model (5) to yield an increased meta-model (7). To make a configuration model for the whole application (counting the greater part of the security and non-security concerns), the composite meta-model (7) is instantiated (8) into a Main Design (9) that is the outline of the non-security concerns and Security Aspects (10) that acclimate the entrance control approach for the application.
The primary subject of this paper is represented by the meaning of the framework (i.e., metamodels and approach consents) required to model security from different angles.

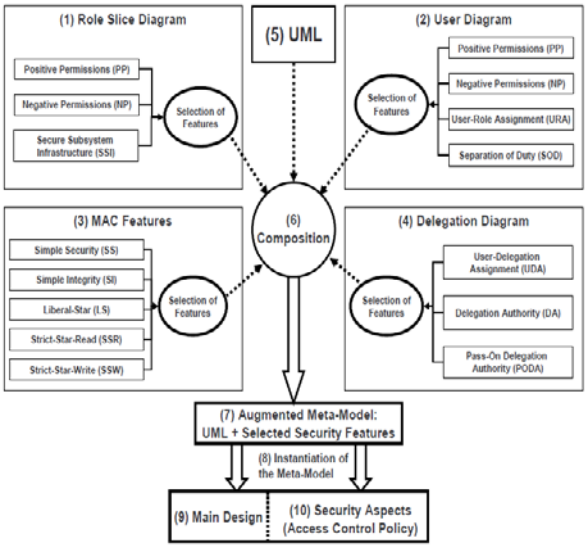

Fig. 2. The proposed approach[16]

3.1. How to model security aspects

To demonstrate a security arrangement, developer must distinguish three key segments: subjects, items, and authorizations. Subjects are the elements that oblige access to the framework. The framework contains an arrangement of items that are the substances that require insurance against subjects. For the proposed approach, class strategies (operations) are the items in the framework that require assurance. Authorizations figure out which operations can get to every subject in the framework. Formally, this is spoken to as takes after:

Subject: A set of themes.

Operation: A set of processes, i.e., the methods of classes.

$Per \subseteq Theme \times Operation$: A set of permissions, where (s, op) *iff* subject *s*is allowed to invoke operation *op*.
To show the necessities of the college application, architects must pick an entrance control structure that speaks to the three sets above, and fulfills the security prerequisites. The college application has two sorts of clients, everyone with various authorizations: educators and understudies. A part based approach is a decent distinct option for gathering clients as indicated by their likenesses.
To dole out consents to parts, developers have two options: allocate positive authorizations unequivocally, or use obligatory access control rules. For this case, accept that originators pick MAC rules, relegating clearances to clients, and groupings to operations, and permitting a subject to get to an operation just if its freedom is more noteworthy than or equivalent to the arrangement of the operation. A few operations that would be permitted by a MAC-based strategy may not be allowed by, so architects can likewise choose to utilize negative authorizations to unequivocally deny them.
Figure 3 demonstrates a part cut chart improved with MAC that speaks to the parts and consents for the college application. The Secure Subsystem, portrayed as a bundle with the generalization <<SubSystemSecuring>>, includes the majority of the operations in the framework that require access control. The protected subsystem additionally characterizes their groupings (unclassified (u), classified (c), secret(s), or top-mystery (ts)), and their entrance mode (read or compose). Parts Teacher and Student show up as bundles with the generalization <<slice_role_diagram>>. They have relegated a leeway and negative authorizations (operations with the generalization <<negation>>).
Parts are associated with the protected subsystem, implying that part consents must be a subset of the operations referenced by the safe subsystem. Figure 4 demonstrates a client graph that delineates clients as bundles with the generalization <<user>>; and, clients' doled out parts as conditions with the generalization <<assignment_of_role>>.

DOI: 10.21279/1454-864X-16-I2-070

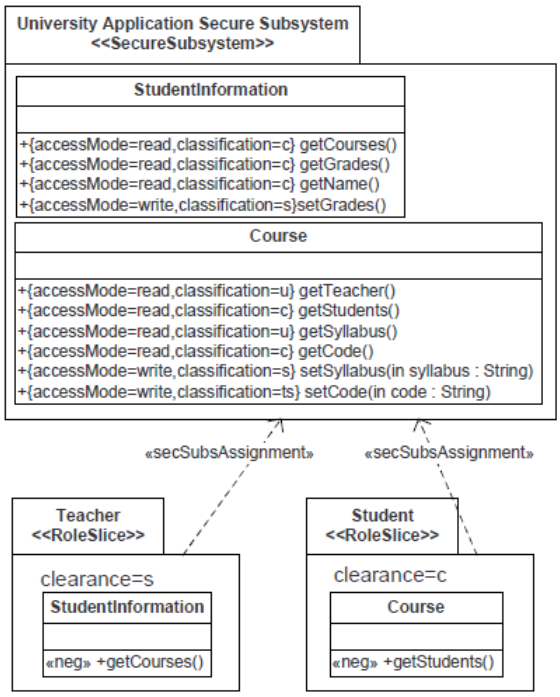

Fig. 3. Role-slice Diagram[16]

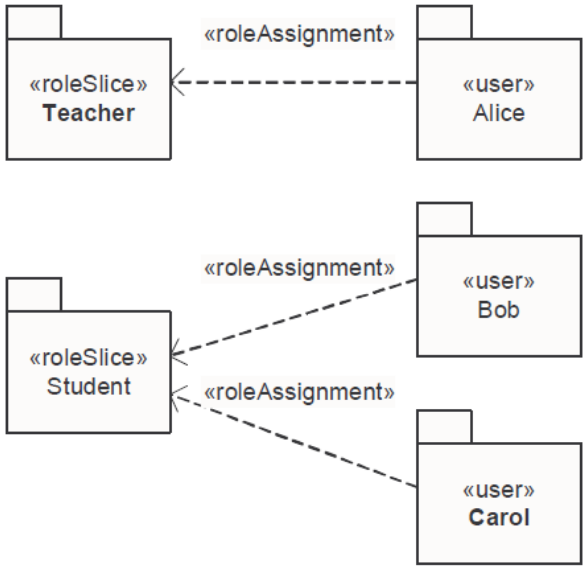

Fig. 4. User Diagram[16]

CONCLUSIONS

Establishing the security aspects on which we have to focus, represent an important phase which must be included in the development process of the software application.

This paper presents a different approach in order to compose and organize features from different securityschemes, to represent the customization of security aspects that are used within specialized diagrams. This includes an extension of UML to include role-based, discretionary, and mandatory access controls, using new UML diagrams for roles, user authorizations, anddelegation, and MAC features that are able to be applied on multiple diagrams. As aresult, software analysts and designers are able to represent access control aspects using UML-based diagrams and an underlying scheme that combines RBAC, MAC and DAC. The unification of these three security schemes provides designers with a broader set of options to define security aspects than each scheme separately. According to the knowledge of other authors, no other approach integrates RBAC, MAC and DAC using a set of security specific UML-based diagrams which are isolated from the main design. Using security features will increase the flexibility to deal with changes in requirements, providing the structure composition of the underlying security characteristics and abilities, which make itpossible to add security characteristics in such way that we will not have any kind ofaffecting process of the non-security aspects of thedesign. The usage of existing UML mechanisms to accomplish this goal (MOFand PackageMerge) will facilitate the integration of the proposed methodwith tools that are based on different standard practices for software development, more preciseUML CASE tools. In the end, we will develop our work to eventually yield animproved secure-software-engineering process with security aspects incorporatedas an integral part of the software design and implementation process.

BIBLIOGRAPHY

[1] Telecom, A.: Glossary 2000. t1.523-2001(2001).
[2] Bell, D., LaPadula, L.: Secure Computer Systems: Mathematical Foundations Model. Technical report, Mitre Corporation (1975).
[3] Liebrand, M., E.H.J.P.C., Ting, T.C.: Role delegation for a distributed, unified RBAC/MAC. In: Proceedings of Sixteenth Annual IFIP WG 11.3 Working Conference on Data and Application Security. (2002).
[4] Ferraiolo, D., Sandhu, R., Gavrila, S., D., K., Chandramouli, R.: Proposed NISTStandard for Role-Based Access Control. ACM Transactions on Information andSystem Security 4 (2001) 224{274.
[5] Object Management Group: UML 2.0 superstructure. Technical report, ObjectManagement Group (2005).

DOI: 10.21279/1454-864X-16-I2-070

[6] De-Win, B., Piessens, F., Joosen, W., Verhanneman, T.: The importance of theseparation-of-concerns principle in secure software engineering (2002).
[7] Pavlich-Mariscal, J., Doan, T., Michel, L., Demurjian, S., Ting, T.: Role Slices:A Notation for RBAC Permission Assignment and Enforcement. In: Proceedingsof 19th Annual IFIP WG 11.3 Working Conference on Data and ApplicationsSecurity. (2005).
[8] Object Management Group: Meta object facility (MOF) core specification. Version2.0. Technical report, Object Management Group (2006)
[9] France, R.B., Ghosh, S., Dinh-Trong, T., Solberg, A.: Model-driven developmentusing uml 2.0: Promises and pitfalls. Computer 39(2) (2006) 59.
[10] JÄurjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Proceedings of the 5th International Conference on The Unified Modeling Language.(2002).
[11] Basin, D., Doser, J., Lodderstedt, T.: Model Driven Security. In: EngineeringTheories of Software Intensive Systems. Springer (2005).
[12] Alghathbar, K., Wijesekera, D.: AuthUML: a three-phased framework to analyze access control speci¯cations in use cases. In: Proceedings of the 2003 ACMWorkshop on Formal Methods in Security Engineering. (2003).
[13] Ray, I., Li, N., Kim, D., France, R.: Using Parameterized UML to Specify andCompose Access Control Models. In: In Proceedings of the 6th IFIP TC-11 WG.
[14] Working Conference on Integrity and Internal Control in Information Systems.(2003).
[15] Doan, T., Michel, L., Demurjian, S., Ting, T.: Stateful Design for Secure Information Systems. In: Proceedings of 3rd International Workshop on Security inInformation Systems (WOSIS05). (2005)
[16] J.Pavlich, L. Michel, and S. Demurjian, Enhacing UML to Model Custom Security Aspects, Department of Computer Science & Engineering, The University of Connecticut.

DOI: 10.21279/1454-864X-16-I2-070