Modeling security aspects in model driven web application development

Dhanya Pramod
Indira Institute of Management
Thathawade, Pune
Maharashtra
91-9422017481
pramoddhanya@yahoo.co.in

ABSTRACT

Web Applications have become the target for hackers in the recent years as there emerged security protection mechanisms and techniques to protect system and network. Our research is mainly focused on web application security aspects and how to introduce it during software development. The research contributions in Model Driven Engineering has paved a way in reducing the complexity in software development. Since this approach enhances the platform Independent modeling and allows transformation to platform specific model, has got wide acceptance. Our work adopt the concepts of Aspect Oriented Modeling to model non functional security requirements while creating domain specific web applications.

This paper focuses on dynamic modeling and incorporating security in the dynamic models. We define UML stereotypes, the extensibility mechanism of UML2.0 to specify the authorization and authentication security aspects of web application. The security aspect models are designed separately and later on weaved with the basic business model . The model transformations are done in semi-automatic way. This approach enables integration of new security models to the MDE as and when they evolve without changing the basic model. UWE(UML based web engineering) principles are used as the MDE approach as it follows OMG standards. The main advantage of including security during MDE is that the applications can be made self defendable to attack from hackers.

Categories and Subject Descriptors

D.2.11 [**Software Architectures**] : Domain-specific architectures, D2.2 **[Software Engineering]:**Design Tools and Techniques-Object Oriented Design methods

General Terms

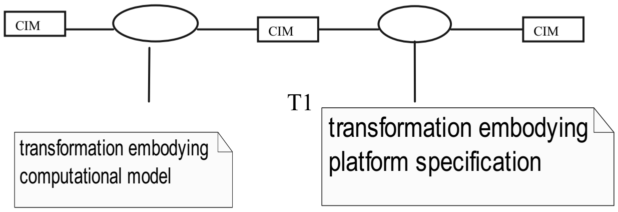
Design, Security, Standardization

Keywords

Authorization, authentication, model transformation, domain

1 INTRODUCTION

The research community has been continuously looking for software engineering techniques which solve the problems of software building and maintenance. CASE(Computer aided Software Engineering) tools have evolved and helped developers to do general purpose graphical modeling of software, but they have not became popular as expected as it was difficult to incorporate code generation and scalability. Model Driven Engineering has been emerged in recent years which reduces the complexity of software development and improves the efficiency and platform independence of software. The Object Management Group(OMG) came up with their Model Driven Architecture(MDA) which realizes MDE. They have introduced the concept of CIM(Computational Independent Model), PIM(Platform Independent Model) and PSM(Platform Specific Model) to convert requirements to software.


CIM
CIM
CIM
T1
transformation embodying computational model
transformation embodying platform specification

figure(1) MDA transformation sequence

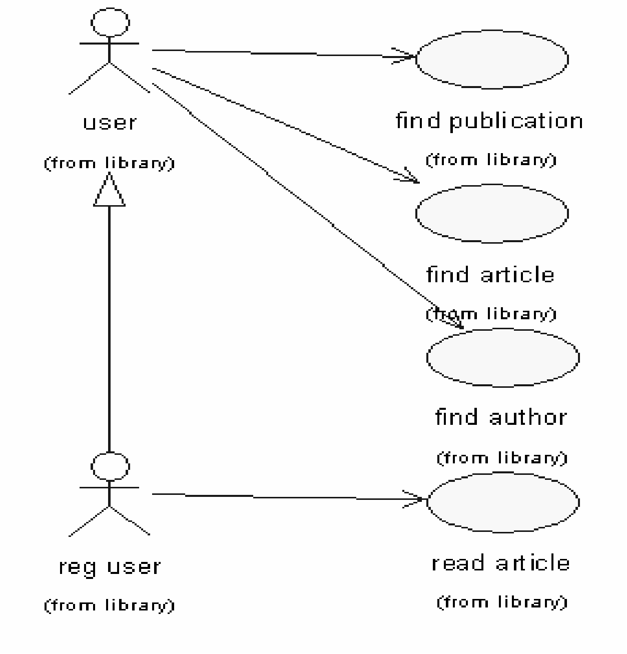
This model is suitable for web application development as there emerge continuously new platforms and languages. MDA provides adaptability and facilitates automatic code generation. It also reduces time and effort of deployment, configuration and quality assurance.

Every application may have some non functional requirements in addition to the functional requirements. AOM[9] deals with weaving the non functional aspects to the basic model and thus increases modularity. It complements the MDA and helps separation of concerns. Basic functionality and aspects can be designed separately and programmer need not know about the aspects. Where ever we need the insertion of aspect marking is done using pointcut in the base model. These join points are weaved with the additional behaviour called aspect advice. The main application of this approach is to model cross-cutting issues that occur in various designs as a separate fragment and apply to existing and newly designed applications. We are trying to adopt this approach to capture the security issues of web applications.

2 MODELING SECURITY ASPECTS

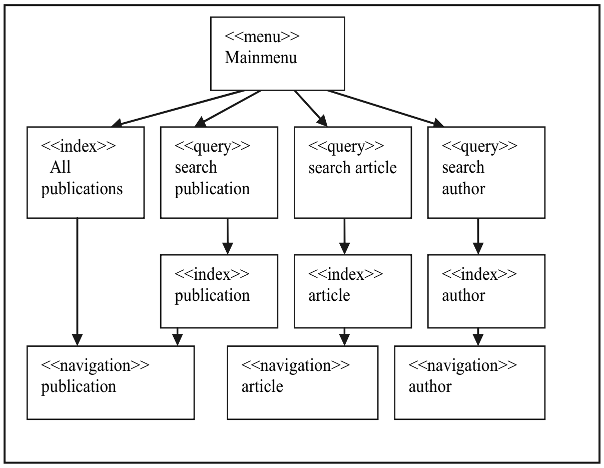
Eventhough the application servers incorporate access control policies, programmatic security is required for fine grained control and developers are made responsible for the incorporation of these concerns. Model driven development weave these concerns even early during analysis and design phase. The generic authorization model is based on subject, credentials, principals and services. In modeling world we have user, role, permission, action, constraint and resource.

The UWE[8] approach uses a UML profile for modeling web systems and provides different views of application such as navigation structure, business processes, content and presentation. Here we take the example of online digital library application to illustrate the access control security concerns such as authorization and authentication. Online digital library publishes various publications(journals) which have articles written by authors. Any user can view the list of publications and articles Details of articles like author names, abstract, references etc are made available to anonymous users but full content can be accessible only by registered users. Usecase diagram of the system is given in fig(2).We identify roles from the usecase diagram ie actors of system. Mapping a user to role is done manually according to the privilege that has to be offered for the user



Figure(2) Usecase diagram for the base model

Navigation model fig(2) depicts the navigation paths used by the modeled application to access the content. Access to the content article is restricted for registered user. We use aspect oriented modeling to incorporate this crosscutting concern.



Figure(3) Navigation diagram of the example

Behavioural aspects can be better modeled in sequence diagram For each object, the operation calls on it are the permissions.

Here we suggests two ways to model authentication aspect. (i)Identify all nonanonymous users entry point and check access control. This way authorize actions. In this case both users will have same presentation view. (ii)Does not show button, link, or actions if the user is not logged in and content is sensitive. Storing the sensitive content in the web application directory path may allow a hacker to access the content by brute force methods to predict the path. Here the security requirement of content protection arises. We follow the first approach as the presentation model of UWE is not much flexible

2.1 Authentication and Authorization Issues

Class diagram depicts the relationship between various entities required for authorizing a user(static model). The major issues related to authorization are (i)Insufficient authorization where an application has increased access control. Modeling the principle of least privilege is necessary to plug this loophole. A better way to ensure this is automatic generation of privileges of each role from the sequence diagram, where the messages sent by an object to another become the rights of the user. (ii) A hacker may predict credential/session and hijack a user. To prevent this attributes and behavior of session, modeled as a class should have some constraints for eg. Large session id, random in nature, expire after a limit, invalidate by client and server during logout (iii)Hacker may force a user's sessionid to an explicit value and cause session fixation. Hence re authentication should be done before giving access to sensitive content. Re authentication is also required when the user change his credential. This repeated process enforces the modeling of authorization as separate aspect to avoid redundancy.

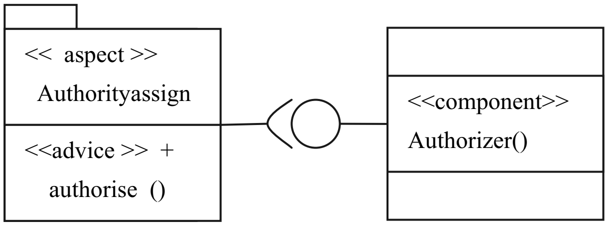
Each security concern is modeled as separate aspect to increase modularity. Even though advisable incorporating certain security concerns may make the testing of application difficult. For eg: enforcing account lockout may not allow multiple request from a single source or multiple login failures.

2.2 Aspect oriented modeling

Security concerns have two related aspects, attacks and countermeasure.. The security aspect is a package inherited from the package metaclass. It contains the different concerns like authorization, authentication, cross site scripting etc. arranged as packages. Packages authorization and authentication are the only scope of this paper.

2.2.1 MODELING AUTHORIZATION

The countermeasure of authorization issues are modeled as separate aspects. The authority assign aspect indicates that the content require authorization to access it. The pointcut is when a user is signed in. The component authorizer is connected to this aspect(figure(4)) and advice authorize() is executed when the weaving is done with the basic model. The aspect authority assign enforces authorization when merging of aspect and base data model is done.

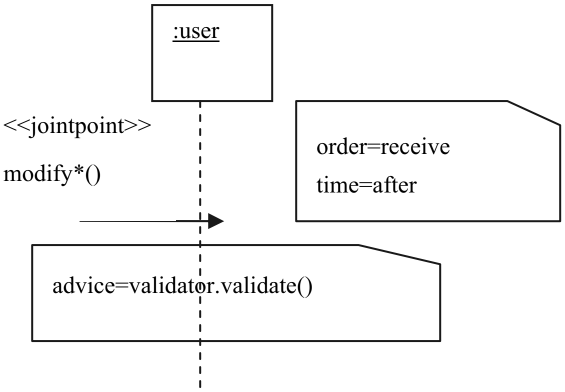


Figure(4) Aspect Authorityassign

The<<leastprivilege>> stereotype is added to the selected methods if the resource is very sensitive for eg: readArticle() of Article and while transforming to navigation model it is replaced by a <<linkaspect>> where pointcut is the link that is connected to the equivalent navigation class and advice is the <<navigation annotation>> which can be used to indicate the user about the restricted access of the link. In order to take care of the validity of passwords or such credentials credential validity aspect is used. The pointcuts are the registration and update user profile sequence diagrams, interception point stereotyped as <<jointpoint>> and the tag value order indicates intercept while send/receive. The tag value time specifies when the advice is executed. So when user call add/modify credential methods the advice validator. validate() is executed after receiving it in user class instance. The weaving behaviour can be observed in sequence diagram fig(5).

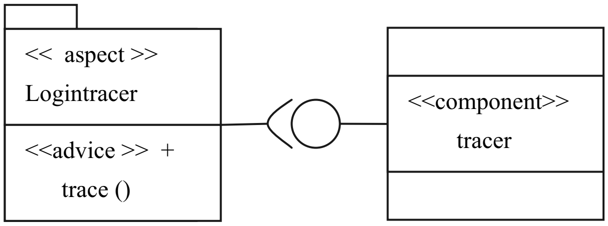
2.2.2 MODELING AUTHENTICATION

Instead of Authentication modeled as a single aspect we have a chain of aspects for increased modularity where related concerns can be added or removed . Here we discuss some of the loopholes that are open to attacks due to incorrect authentication. (i)An attacker may crack password by the use of automated tools/use of brute force attack. Tracing of failed login attempts and multiple accesses to the same url originating from the same source should be done. (ii)Weak password recovery is another loophole and the credential should have some constraint to accept passwords which are difficult to guess.



Figure(5) weaving of credential validity aspect

We define some more aspects for the authentication procedure besides the basic authentication. Whenever the user tries to access the restricted content the credentials of the user is requested. The credentials should be passed through a secured channel and cryptography should be enforced . In connection with this encryption aspect is defined. All messages passed from user to system interface are the pointcuts. To avoid the brute force attack logintracer aspect is defined figure(6). Error message of authentication aspect is the pointcut. Once the credentials are found acceptable the user is provided with a unique session id. Aspect secure session is added to ensure the protection of users session.

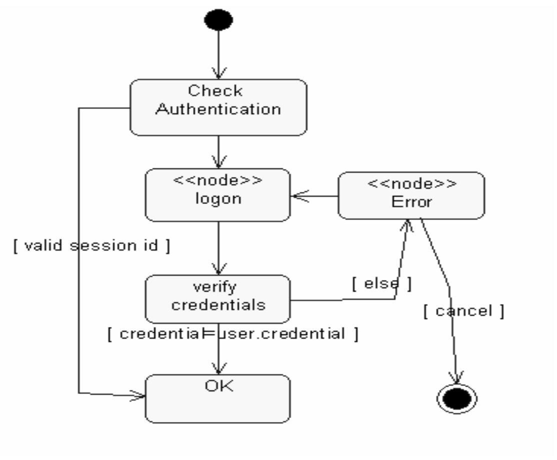


Figure(6) Aspect to trace failed logins

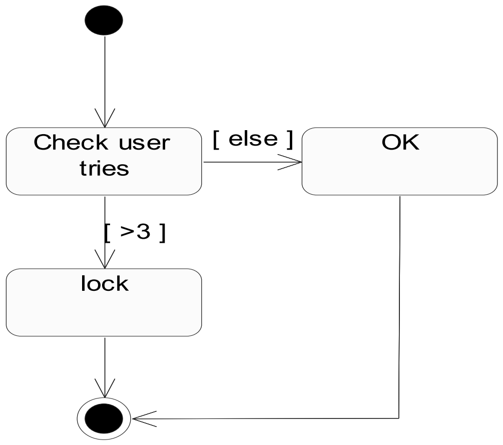
This is marked in the sequence diagram with a pointcut where the user receives a valid logon message. Whenever the user is in secured session every request to content demands re authentication. Aspect reauthenticate is used for this. In connection to this we define a runtime aspect expire after limit. As and when the user logout the sessionInvalidate aspect is incorporated. The aspects are ordered and can be incorporated selectively according to the requirements.

Another diagram which is used to model the behaviour of the system is statechart model. It as used by Koch[8] show the design level details. Following statechart diagrams depicts the basic authentication (figure (7))and login tracer aspect(figure(8)).

The aspects are ordered to satisfy the authorization and authentication requirement in the following manner. Authorityassign, credentialvalidate, authentication, encryption logintracer, securesession, reauthenticate , sessioninvalidate etc.



Figure(7) Basic authentication aspect Statechart



Figure(8) Logintracer statechart

3 MODEL TRANSFORMATION

The UWE approach does not support fully automated model transformation. Semiautomatic model transformations are done to convertions of certain models like content to navigation. The suggested weaving module has to be incorporated according to the above approach to support automation.

4 RELATED WORK

Jurjens[3,4] defines a general approach to security concepts modeling but we focus on modeling web application specific countermeasures of attacks.

Lidia[9] introduce a approach that can be used to weave multiple aspects in to the executable UML model. We do weave multiple aspects of the security domain and thus more specific Koch[8] use design stage weaving of aspects and use state machines to weave them. We focused on both analysis and design phase dynamic models.

5 CONCLUSIONS AND FUTURE WORK

We have tried to analyse major access control problems of web applications and used the notations of UML based UWE MDE to model the domain. Incorporation of the automation of weaver is the future plan. We also plan to incorporate the script injection issues of web application.

Another area of work that is in progress is the visualization of behaviour using executable UML

6 ACKNOWLEDGEMENT

We would like to thank all those who have reviewed work.

7 REFERENCES

[1] Filippo Ricca, Massimiliano Di Penta, Marco Torchiano,, Paolo Tonella, Mariano Ceccato , The Role of Experience and Ability in Comprehension Tasks supported by UML.

[2] The Object Management Group(OMG): Unified Modeling Language: Superstructure, Version 2.0 Final Adopted Specification, OMG, http://www.omg.org(2003)

[3] J.Jurjens. Secure Systems Development with UML, Springer, 2004.

[4] J.Juerjens. UMLsec: Extending UML for Secure Systems Development. In Proc. Of 5th Int. Conf. on the Unified Modeling Language, Lect. Notes in Comp. Sci. 2460 pages 412-425, Springer, 2002.

[5] Peter F. Linington and Pulitha Liyanagama. Incorporating Security Behavior into Business Models using a Model Driven Approach. 11th IEEE International Enterprise Distributed Object Computing Conference(2007)

[6] N. Moreno, P. Fraternali and A. Vallecillo. WebML modielling in UML, IET Software, 2007 pp 67-80

[7] Stereotypes , Softwae Engineering, 2007 , ICSE 2007 May 29 th International Conference, Pages 375-384.

[8] Nora Koch and Andreas Kraus , The Expressive Power of UML-based Web Engineering, 2nd Int. Workshop on Web-oriented Software Technology(IWWOST02), Malaga, Spain, June 2002.

[9] Lidia Fuentes, Pablo Sanchez , Designing and Weaving Aspect-Oriented Executable UML models, Journal Of Object Technology August 2007.

[10] A. Schauerhuber, M. Wimmer, E. Kapsammer, W. Schwinger and W. Retschitzegger, Bridging WebML to model-driven engineering: from document type definitions to meta object facility, IET Softw. 2007.

[11] Hubert Baumeister, Alexander Knapp, Nora Koch, Gefei Zhang, Modeling Adaptivity with Aspects.

[12] Application security-An essential part of your risk management program-IBM whitepaper 2005,pp1-2

[13] Yao-wen Huang, Fang Yu. Securing web application code by static analysis and runtime protection. 13th International