I.J. Modern Education and Computer Science, **2015, 2, 48-55**
Published Online February 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2015.02.07

Experimental Analysis of SPF Based Secure Web Application

Nitish Pathak
Research Scholar and Corresponding Author, UTU, Dehradun (INDIA)
Email: nitish_pathak2004@yahoo.com

Dr. Girish Sharma and Dr. B. M. Singh
Departments of CS&E, BPIBS, Govt. of NCT, Delhi (INDIA)
Department of CS&E, College of Engineering Roorkee, Roorkee (INDIA)
Email: gkps123@gmail.com and bmsingh1981@gmail.com

Abstract—In this paper we will propose model driven software development and Security Performance Framework (SPF) Model to maintain the balance between security and performance for web applications.

We propose that all security in a Trusted Operating System is not necessary. Some non-essential security checks can be skipped to increase system performance. These non essential security checks can be identified in any web application.

For implementation of this Security Performance framework based trusted operating system, we propose object oriented based Code generation through forward engineering. This involves generating source code of web application from one or more Object oriented Rational Rose model. The novel integration of security engineering with model-driven software expansion approach has varied advantages.

To maintain security in various applications like E-commerce, Banking, Marketplace services, Advertising, Auctions, Comparison shopping, Mobile commerce Payment, Ticketing, Online insurance policy management, we have to use high secured operating systems. In this regard a number of trusted operating systems like Argus, Trusted Solaris, and Virtual Vault have been developed by various companies to handle the increasing need of security. Due to high security reason these operating systems are being used in defense. But still these secure operating systems have limited scope in commercial sector due to lower performance; actually this security will come at a cost. This paper analyzes UML-based software development solutions for SPF to manage the security, performance and modeling for web applications.

Index Terms—UML, Code generation, Model driven software development, OO Rose Mode-Object Oriented Rose Model, TOS -Trusted Operating System.

I. Introduction

Model-driven software development involves software system modeling and automatically code generation from these models. A system and software unit can be modeled at diverse levels of concept or from various viewpoints. The grammar of each model and replica is defined by a Meta model. Systems for Model-driven software development can be viewed as a novel generation of visual programming languages [1].

The Meta model defines the syntax of the UML modeling languages, a model plays the role of the source code, and the code generator replaces the compiler. Using this approach, it is possible to generate automatically huge amounts of source code and other artifacts, e.g. deployment descriptors. Files can also be created based on comparatively concise models. This improves the productivity and output of the development process as well as the quality and value of the resulting systems. It is also a great step towards the platform independent design of object oriented web application systems [2][3].

UML is well suited for designing secure systems. Therefore, Secure UML allows still developers without a strong security environment to develop secure and safe systems. With our prototype and trial product, we developed case studies for a product purchase system based web application. We also developed a scheduling application and code generator as discussed in this paper. UML has rapidly become the language of choice for developers who wish to visualize and model the system under development. The UML includes use cases to facilitate visualization of user requirements, class diagrams to visualize the design of the software and sequence diagrams to visualize the behavior of the objects in the system [4].

In the projected process, a web based e-commerce system is totally specified using object-oriented conceptual modeling techniques [5][6].Security and web application protection plays an innermost role in the development and functioning of several large-scale distributed software systems. On the other hand, an analysis of today’s software design and development processes reveals that the careful engineering of security into the overall web system design is frequently neglected [2][7].

Security features are usually built into web application in an ad-hoc approach or are only integrated afterward during the system administration phase. There are several reasons for this. First, security and protection is a “horizontal” feature of software development that affects almost each component of an application and its incorporation into the software design and development process is not well understood.

Second, there is a lack of tools supporting security engineering. Third, the addition of security into a web application system by hand is complicated. Errors frequently arise due to the lack of experience and knowledge of the individual developers [2][3]. These developers are usually not security experts and they require actual guidelines for constructing secure on line web applications. In general, the post-hoc, low-level incorporation of security and protection has a negative impact on the quality and the worth of resulting web applications [5].

Due to gravity and importance of security in web applications; we should use trusted operating systems for the same [2][3].

High secure trusted operating systems recommend a number of safety mechanisms that can assist, defend information, make a system hard to break into, and detain attacks far better than conventional operating systems[8] [2].

Actually, this security will come at a charge, since it can humiliate the performance of an operating system. This performance defeat is one of the reasons why TOS have not become popular in web applications. UML can be applied in several areas like embedded systems, product purchase web applications, and social networking sites commercial applications etc. A number of UML tools generate program language code from UML in various object oriented programming languages. UML can be used for modeling and designing the entire system independent of platform/ language. Trusted operating systems can be designed by using UML based concepts [1][9].

While Trusted Operating Systems present an unbelievable quantity of security, observations about computing workloads recommend that only a few parts of the operating system security are really essential[6].

Web servers are the most excellent example. For various web servers, the greater part of the information on the server is freely legible and accessible on the Internet. So, if a Trusted Operating System is used on a web server, some security used to protect the privacy of the server's information is not needed. Any security used to protect and defend the confidentiality and privacy of web server data can be considered a waste and misuse of computational resources [1][10]. The security needed in online web servers is the security to guard the integrity of data, not the privacy of data. Based on this surveillance, this research article proposes the Security Performance Flexibility Model (SPF) of TOS to Web applications [6].

This paper will comprise four sections. Section I comprise introduction of SPF model, software design and development and its significance in Web applications, Section II comprise the related work for SPF based secure Web applications, Section III comprise the architecture of system SPF and object oriented development, Section IV comprise the description and object oriented experimental analysis of the problem and the last section comprise the references.

II. Related Work

According to SPF observation we feel that not all computing workloads need all the security in secure Operating Systems. SPF permits computer system administrators to selectively disable parts of the security protection in secure Operating Systems [9]. Whenever we disable some needless parts of the Trusted Operating System security, performance and efficiency of the computer system can potentially be increased [1].

System administrators can use the SPF framework to balance the security and performance needs in their particular web computing environment [11]. First we have to identify which part can be disabled to achieve the maximum performance in specific web application. We should identify various operations in product purchase web application to disable the undesirable operations to improve the performance in all respect. Based on Questionnaire and survey related to web applications like e-commerce sites, job portals, we can find trends, Hit rate, Max operations performed by users, Frequency of read, write, upload, share, comment, messaging operations, What is the chance that an event will occur?, Which patterns are significant?

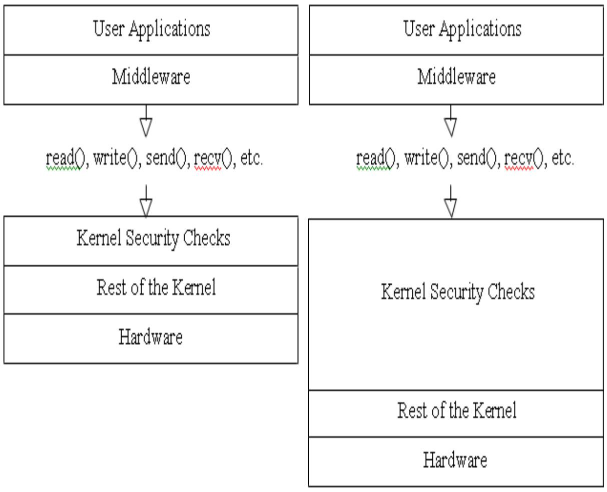
After identifying the answer of above question we can identify the frequency of operations performed by web users and according to our desired requirement we can disable some security checks in trusted operating systems to improve performance of web applications [5][12]. As we know all security checks are not important in all web applications. Selection of these criteria will vary from one web application to another application.

III. Architecture

The design of Trusted Operating Systems is demonstrated under the Figure 1. As a prompt to the reader, the design of an online product purchase system is not the same as the execution. A mixture of implementations for trusted safety mechanisms can be followed for the web applications.

In secure operating systems, there is a larger amount of security and safety placed into the operating system. Figure 1.b indicates this security and safety with a much thicker coating of kernel security checks [13][10].What is inside the kernel security check layer of a Trusted Operating System depends on the design & implementation of web application.

We can see in figure 1.b, the kernel security layer in trusted operating system may comprise DAC, MAC, Least Privilege, auditing, or any more number of extra security features [6][14].



(a)Normal Operating Systems (b) Trusted or Secure Operating Systems

Fig.1. Security Framework of TOS

In figure 1.b the main point is that the kernel safety checks are a large amount in nature than conventional operating systems. This large layer of security and safety causes Trusted Operating Systems to bear performance degradation [8]. All system calls to the kernel of TOS must go through this thick layer of security checks before they can do any useful and valuable work [1][10].

IV. Problem Description And Solution

Design in Figure 1.b shows, the supplementary security checks in the kernel will cause Trusted Operating Systems to be slower than conventional operating systems [6]. If we implement the same trusted operating system for web applications like e-commerce sites, job portals, then we get lower performance but more security features as we are using the trusted operating system for the same. The main problem is to balance security and performance of secure operating system for web applications.

The incorporation of security engineering into a model-driven software development process has the following advantages. To start with, security and safety requirements can be formulated and integrated into software system design at a high level of abstraction for various web applications. In this way, it becomes probable to design and develop security aware web applications that are designed with the objective of preventing violations of a safety and security policy. For example, a database query can be designed so that users can only retrieve and access those data records that they are allowed to access. In addition, the model information can be used to notice and to accurate design errors or to confirm the correctness and rightness of the mapping between requirements and their understanding in a design process [10][11].

Unfortunately, many systems are constructed without the use of modeling and visualization artifacts, due to constraints imposed by deadlines, or a shortage of manpower. In this research paper we will suggest Object-oriented analysis based designing using forward engineering. This approach is accepted technical process to analyzing, designing a web application, TOS or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to promote better stakeholder communication and software product quality[6][1].

With a variety of diagrams from UML we can demonstrate clear views of system. If we model any problem before its progress there are less chances that our development sketch will go wrong, thus decreasing development costs. Modeling decreases the risk and danger of mistakes [15].

For better performance for specific computer system workloads, this research paper suggests the security performance flexibility (SPF) model for Trusted Operating Systems. The combination of SPF and forward engineering can help developers, designers, and analyst as well as web applications user. The architectural thought behind the SPF structure is illustrated in Figure 2. For example, a system manager can use SPF to turn off all read security and safety checks in a web server [16]. By turning off all the read security checks of a web server, it is probable the web server's throughput and efficiency can be increased and improved [1][17].

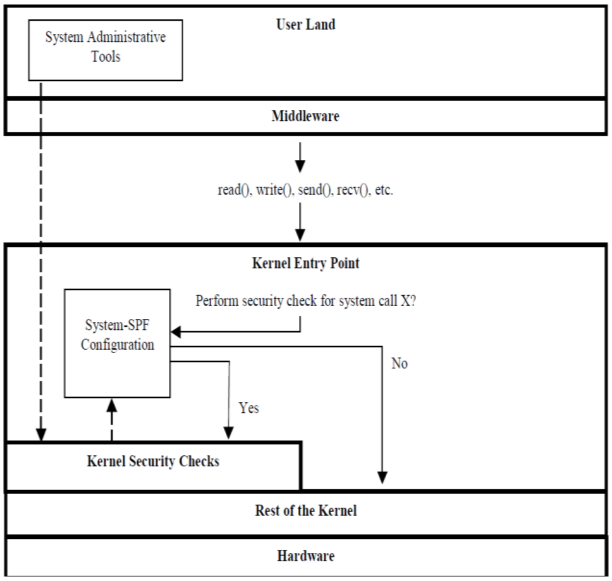


Fig.2. System-SPF Architecture for web applications

Taking the decision to adopt UML in the development process of web applications does not mean that all UML diagrams must be used. It is up to the modeller to decide, which diagrams to use in the development process. The decision is usually based on the nature of the system; the domain, the functionalities, as well as the preferences of the modeller, since some UML diagrams might in some cases replace each other [9].

We can implement the SPF and code generation in following case of web applications. Following requirements are for a product purchase system:

- The administrator runs inventory reports. Every time inventory reports are run, inventory data is loaded from disk.

- The administrator updates the inventory stock. Every time inventory stock is updated, inventory data is loaded from disk. Also, every time inventory stock is updated, inventory data is saved to a disk.
- A (general) make-a-sale (hint: meant to be a verb-noun phrase) can be of two specialized kinds: (1) make-a phone order sale; and (2) make-a walk-in sale.
- A sales clerk records the make-a-walk-in sales.
- A telephone operator, a specialized kind of a sales clerk, handles and records all make-a phone orders.
- Whenever there is a make-a-sale, the inventory stock is updated.
- A sale may need to verify a credit card if the purchase is a credit card purchase.
- A sale may need to verify a check if the purchase is a check purchase.

First we have to design the use case diagram because Use cases are an influential method for the elicitation and documentation of blackbox functional requirements [17]. Use cases are printed in natural language, use cases are effortless to understand and provide an outstanding way for communicating with customers and users. Use cases diagram provide the groundwork on which to specify end-to-end timing requirements for real-time web applications. Use cases can assist control the complexity of huge projects by decomposing the complexity into most significant functions (i.e., use cases) and by specifying applications from the users' perspective [9][18].

The generic Use case diagram of above case study is as follows:–

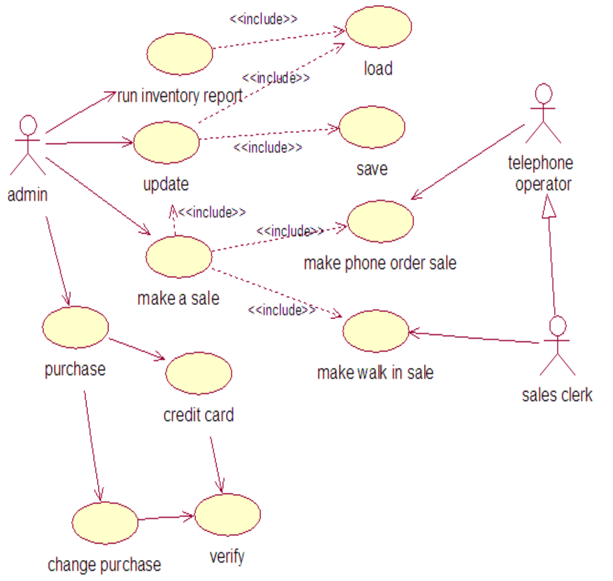


Fig.3. Use case diagram for a product purchase based web Application

For the above web application we can design and develop the class diagram with the help of above use case diagram and can perform forward engineering for the same application with the help of IBM Rational Rose Software [2][15]. We know a class diagram is "a diagram that shows a set of classes, interfaces, and collaborations and their relationships; class diagrams address the static design view of a system; a diagram that shows a collection of declarative (static) behavior"

Class diagrams are helpful in the development process from the very beginning, where they can be used for identifying system requirements and its entities [12]. For example, they construct the initial system architecture in the analysis phase, while in the design phase; they are refined and extended to represent the complete system specifications. In UML, a class is represented as a box with three vertical sections. The top section shows the name of the class [23]. The middle section displays the variables belonging to the class, with symbols representing the visibility (public, protected, or private) and properties (constant or static). The bottom section contains the member functions of the class [19]. Each method has a name, signature, and properties. On the basis of code and design we can implement the SPF model for trusted operating system i.e. well suited for various web applications [20].

The generic class diagram of online product purchase web application is as follows –

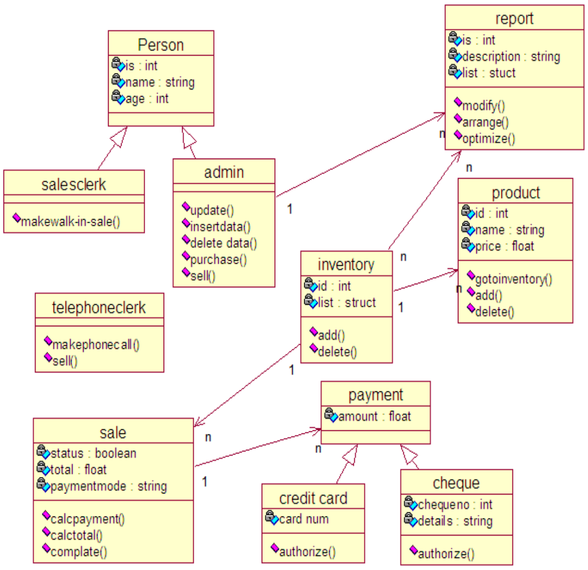


Fig.4. Generic Class Diagram for a product purchase web application

FORWARD ENGINEERING FOR PRODUCT PURCHACE BASED WEB APPLICATION

Forward engineering in Rose is component-centered. Class wise code i.e. component-based programming is given below, this code can help the designers, developers during the development process of web applications [21][22].The class wise corresponding sample code segment in C++ is as follows-

Code:

```
#include "admin.h"
//##ModelId=4F7AB819004E
admin::update ()
{
}
```

```
//##ModelId=4F7AB81B037A
admin::insertdata ()
{
}
//##ModelId=4F7AB81E0203
Admin::delete data ()
{
}
//##ModelId=4F7AB82101F4
admin::purchase ()
{
}
//##ModelId=4F7AB825037A
admin::sell ()
{
}
#ifndef ADMIN_H_HEADER_INCLUDED_B0853EE5
#define ADMIN_H_HEADER_INCLUDED_B0853EE5
//##ModelId=4F7AB8140222
class admin : public Person
{
  public:
    //##ModelId=4F7AB819004E
    update ();
    //##ModelId=4F7AB81B037A
    insertdata();
    //##ModelId=4F7AB81E0203
    delete data();
    //##ModelId=4F7AB82101F4
    purchase ();
    //##ModelId=4F7AB825037A
    sell ();
};
#endif /* ADMIN_H_HEADER_INCLUDED_B0853EE5 */
#include "cheque.h"
//##ModelId=4F7AB95302CE
cheque::authorize ()
{
}
#ifndef CHEQUE_H_HEADER_INCLUDED_B0857ACB
#define CHEQUE_H_HEADER_INCLUDED_B0857ACB
//##ModelId=4F7AB941003E
class cheque : public payment
{
  public:
    //##ModelId=4F7AB95302CE
    authorize ();
  private:
    //##ModelId=4F7AB947035B
    int chequeno;
    //##ModelId=4F7AB94C03C8
    string details;
};
#endif /* CHEQUE_H_HEADER_INCLUDED_B0857ACB */
#include "credit card.h"
//##ModelId=4F7AB93B0203
credit card::authorize()
{
}
#ifndef
CREDIT_CARD_H_HEADER_INCLUDED_B0850B00
#define
CREDIT_CARD_H_HEADER_INCLUDED_B0850B00
//##ModelId=4F7AB9300167
class credit card : public payment
{
  public:
    //##ModelId=4F7AB93B0203
    authorize();
  private:
    //##ModelId=4F7AB937000F
    card num;
};
#endif /*
CREDIT_CARD_H_HEADER_INCLUDED_B0850B00 */
#include "inventory.h"
//##ModelId=4F7AB8E50251
inventory::add()
{
}
//##ModelId=4F7AB8E6033C
inventory::delete()
{
}
#ifndef INVENTORY_H_HEADER_INCLUDED_B0856825
#define INVENTORY_H_HEADER_INCLUDED_B0856825
//##ModelId=4F7AB8D6037A
class inventory
{
  public:
    //##ModelId=4F7AB8E50251
    add();
    //##ModelId=4F7AB8E6033C
    delete();
  private:
    //##ModelId=4F7AB8DB0290
    int id;
    //##ModelId=4F7AB8DF0148
    struct list;
};
#endif /* INVENTORY_H_HEADER_INCLUDED_B0856825
*/
#ifndef PAYMENT_H_HEADER_INCLUDED_B0850957
#define PAYMENT_H_HEADER_INCLUDED_B0850957
//##ModelId=4F7AB922033C
class payment
{
    //##ModelId=4F7AB927032C
    float amount;
};
#endif /* PAYMENT_H_HEADER_INCLUDED_B0850957 */
#ifndef PERSON_H_HEADER_INCLUDED_B0851005
#define PERSON_H_HEADER_INCLUDED_B0851005
//##ModelId=4F7AB7FC02CE
class Person
{
   //##ModelId=4F7AB8040222
    int is;
    //##ModelId=4F7AB807007D
    string name;
    //##ModelId=4F7AB8090203
    int age;
};
#endif /* PERSON_H_HEADER_INCLUDED_B0851005 */
#include "product.h"
//##ModelId=4F7AB879032C
product::gotoinventory()
{
}
//##ModelId=4F7AB883009C
product::add()
{
}
//##ModelId=4F7AB884003E
```

```
product::delete()
{
}
#ifndef PRODUCT_H_HEADER_INCLUDED_B08533FA
#define PRODUCT_H_HEADER_INCLUDED_B08533FA
//##ModelId=4F7AB8630251
class product
{
  public:
    //##ModelId=4F7AB879032C
    gotoinventory();
    //##ModelId=4F7AB883009C
    add();
    //##ModelId=4F7AB884003E
    delete();
  private:
    //##ModelId=4F7AB8680261
    int id;
    //##ModelId=4F7AB86D008C
    string name;
    //##ModelId=4F7AB87102DE
    float price;
};
#endif /* PRODUCT_H_HEADER_INCLUDED_B08533FA
*/
#include "report.h"
//##ModelId=4F7AB8530251
report::modify()
{
}
//##ModelId=4F7AB8560261
report::arrange()
{
}
//##ModelId=4F7AB8590186
report::optimize()
{
}
#ifndef REPORT_H_HEADER_INCLUDED_B0851331
#define REPORT_H_HEADER_INCLUDED_B0851331
//##ModelId=4F7AB83D00CB
class report
{
  public:
    //##ModelId=4F7AB8530251
    modify();

    //##ModelId=4F7AB8560261
    arrange();
    //##ModelId=4F7AB8590186
    optimize();
  private:
    //##ModelId=4F7AB8420167
    int is;
    //##ModelId=4F7AB844034B
    string description;
    //##ModelId=4F7AB84A0148
    stuct list;
};
#endif /* REPORT_H_HEADER_INCLUDED_B0851331 */
#include "sale.h"
//##ModelId=4F7AB913035B
sale::calcpayment()
{
}
//##ModelId=4F7AB917031C
sale::calctotal()
{
}
//##ModelId=4F7AB91B0213
sale::complate()
{
}
#ifndef SALE_H_HEADER_INCLUDED_B085120D
#define SALE_H_HEADER_INCLUDED_B085120D
#include "payment.h"
//##ModelId=4F7AB8F500AB
class sale : public payment
{
  public:
    //##ModelId=4F7AB913035B
    calcpayment();
    //##ModelId=4F7AB917031C
    calctotal();
    //##ModelId=4F7AB91B0213
    complate();
  private:
    //##ModelId=4F7AB8F90167
    boolean status;
    //##ModelId=4F7AB8FD02DE
   float total;
    //##ModelId=4F7AB9010242
    string paymentmode;
};
#endif /* SALE_H_HEADER_INCLUDED_B085120D */
#include "salesclerk.h"
//##ModelId=4F7AB83302DE
salesclerk::makewalk-in-sale()
{
}
#ifndef SALESCLERK_H_HEADER_INCLUDED_B085310E
#define SALESCLERK_H_HEADER_INCLUDED_B085310E
#include "Person.h"
//##ModelId=4F7AB82A0280
class salesclerk : public Person
{
  public:
    //##ModelId=4F7AB83302DE
    makewalk-in-sale();
};
#endif /*
SALESCLERK_H_HEADER_INCLUDED_B085310E */
#include "telephoneclerk.h"
//##ModelId=4F7AB8CF01F4
telephoneclerk::makephonecall()
{
}
//##ModelId=4F7AB8D201A5
telephoneclerk::sell()
{
}
#ifndef
TELEPHONECLERK_H_HEADER_INCLUDED_B0852E98
#define
TELEPHONECLERK_H_HEADER_INCLUDED_B0852E98
//##ModelId=4F7AB8C6007D
class telephoneclerk
{
  public:
    //##ModelId=4F7AB8CF01F4
    makephonecall();
    //##ModelId=4F7AB8D201A5
    sell();
};
```

```
#endif /*
TELEPHONECLERK_H_HEADER_INCLUDED_B0852E98
*/
```

Here we have converted the object oriented model into the target language (e.g. C++, Visual Basic, COBOL, and Java).in above example we generated code in C++. In this approach main benefit is higher flexibility that the developer is not afraid to repeat the process several times and is therefore more adventurous.

V. Conclusion And Future Scope

In this paper, we have analyzed the integration of security engineering into a model-driven software development approach. We specifically focused on UML and component-based programming for forward engineering for web applications in trusted operating systems.

In this research, we have presented a SPF based software development process for web applications. This is based on object-oriented and SPF based conceptual modeling techniques applied to the development of online web applications. Then, we clarify how these primitives can be mapped in a web solution for automating the software development process through forward engineering using UML.

Presently our design focuses on static and fixed design models, which are comparatively close to the implementation. It is significance considering whether the efficiency and effectiveness of the development process of secure web applications can be improved by annotating models at a higher level of abstraction (e.g. analysis, Design etc) or by annotating dynamic models, e.g. state machines.

Furthermore, a number of critical questions concerning the development process are still open, e.g. how are roles and permissions identified?

Future research work will focus on modeling security requirements and design information using dynamic UML models. Additionally, the development process for secure web systems starting with the preliminary analysis up to the complete secure web system design will be investigated. In this perspective, we will study and examine the possibility of propagating security requirements between analysis and design models and ways to prove the compatibility of requirements and design information given at different levels during web based software development.

References

[1] Booch G., Jacobson I., Rumbaugh J. The UML Language Users Guide. Addison-Wesley, ACM Press, 1999.

[2] J. J ürjens. Towards development of secure systems using UMLsec. In H. Hussmann, editor, Fundamental Approaches to Software Engineering, 4th International Conference, Proceedings, LNCS, pages 187–200. Springer, 2001.

[3] P. Epstein and R. Sandhu. Towards a UML based approach to role engineering. In Proceedings of the fourth ACM Workshop on Role-based access control, pages 135–143. ACM Press, 1999.

[4] http://en.wikipedia.org/wiki/Applications_of_UML.

[5] Roy A, Karforma S, A Survey on E-Governance Security, International Journal of Computer Engineering and Computer Applications (IJCECA). Fall Edition 2011, Vol 08 Issue No. 01, Pp: 50-62, ISSN 0974-4983.

[6] Baresi L., Garzotto F., Paolini P. From Web Sites to Web Applications: New Issues for Conceptual Modeling. ER'2000 Workshop on Conceptual Modeling and the Web, LNCS 1921. Springer-Verlag, 2000, pp. 89-100.

[7] K. Koskimies, T. M ̈annist ̈o, T. Syst ̈a, and J. Tuomi, "Automated Support for Modeling OO Software", IEEE Software, 15, 1, January/February 1998, pp. 87–94.

[8] Mark Aldrich. Secured Systems and Ada: A Trusted System Software Architecture. Proceedings of TRI-Ada, 2001.

[9] P. Epstein and R. Sandhu. Towards a UML based approach to role engineering. In Proceedings of the fourth ACM workshop on Role-based access control, pages 135–143. ACM Press, 1999.

[10] G ómez J., Cachero C., Pastor O. Extending a Conceptual Modeling Approach to Web Application Design. In Proc. Conference on Advanced Information Systems Engineering (CAiSE), LNCS 1789, Springer- Verlag, 2000, pp. 79-93.

[11] J. J ürjens. Towards development of secure systems using UMLsec. In H. Hussmann, editor, Fundamental Approaches to Software Engineering, 4th International Conference, Proceedings, LNCS, pages 187–200. Springer, 2001.

[12] G ómez J., Cachero C., Pastor O. Extending a Conceptual Modeling Approach to Web Application Design. In Proc. Conference on Advanced Information Systems Engineering (CAiSE), LNCS 1789, Springer- Verlag, 2000, pp. 79-93.

[13] http://www.ijric.org/volumes/Vol8/Vol8No8.pdf.

[14] http://cairo.cs.uiuc.edu/publications/papers/thesis-achu.pdf.

[15] James B.D. Joshi, Walid G. Aref, Arif Ghafoor, and Eugene H. Spafford. Security Models for Web-Based Applications, 2001.

[16] Christian Friberg and Achim Held. Support for Discretionary Role-Based Access Control in ACL-oriented Operating Systems. Proceedings of the Second ACM Role-Based Access Control Workshop, 2005.

[17] http://www.ksc.com/articles/usecases.htm.

[18] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, and Gunnar Övergaard, Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, Wokingham, England, 1992. David Ferraiolo and Richard Kuhn. Role-Based Access Control. Proceedings of the 15th National Computer Security Conference, 2004.

[19] Power, J. F. & Malloy, B. A. (2000), an approach for modeling the name lookup problem in the C + + programming language, in `ACM Symposium on Applied Computing', Como, Italy.

[20] Tamar Richner and St ́ephane Ducasse. Recovering high-level views of object-oriented applications from static and dynamic information. In Hongji Yang and Lee White, editors, Proceedings ICSM'99 (International Conference on Software Maintenance). IEEE, September 1999.

[21] Vishal Jain, Dr. Mayank "Singh Ontology Based Information Retrieval in Semantic Web: A Survey, I.J. Information Technology and Computer Science", 2013/10, 62-69, http://www.mecs-press.org.

[22] Nan Nie, Suzhi Zhang, Quality Test Template toward Multi-user Access Control of Internet-Based System, International Journal of Information Engineering and Electronic Business (IJIEEB). Vol.3, No.3, June 2011, PP.30-36, ISSN: 2074-9031 (Online) http://www.mecs-press.org.

[23] Nitish Pathak, Dr. Girish Sharma and Dr. B. M. Singh " Forward Engineering based Implementation of TOS in Social Networking" International Journal of Computer Applications, USA, Volume 102 – No.11, September 2014,PP.33-38, ISSN: 0975 – 8887.

Authors' Profiles

Dr. Girish Sharma is currently working as Associate Professor& Head of BPIBS, GNCT OF DELHI. He has More than 19 Years of academic / research and administrative experiences in reputed organizations like Hindu College, Sonepat, ABES Engineering College, Ghaziabad, and College of Engineering Roorkee. He completed his PhD (2005) in Mathematics (Inter Disciplinary) at Department of Mathematics, Gurukul Kangri University, Haridwar. He completed his M.Tech in Computer Technology and Application (2003) from Delhi College of Engineering (DCE), Delhi University, Completed his MCA and M.Sc (Physics) in 1993 and 1996 respectively from Gurukul kangri University, Haridwar. His current research interests include different aspects of Distributed systems, trusted operating Systems; Object oriented based Forward Engineering and Data Mining.

He is a life time member of various professional bodies like IEEE, CSI and ISTE etc.

He has published more than 55 research papers in various reputed national / international journals.

Dr.Brijmohan Singh is currently working as Associate Professor& Head of computer science and engineering department at COER, Roorkee. He has More than 12 Years of academic/research and administrative experiences in reputed organizations. He completed his PhD (2012) in Computer science and Engineering from Uttrakhand Technical University, Dehradun, UK. He completed his Master of Technology (Computer Science) with First Class (76.8%, Grade 'A') from Rajasthan Vidhya peeth Deemed University, Udaipur Rajasthan, in the session 2005 to 2007. He did his Bachelor of Engineering (Computer Science & Engineering) with First Class (74.89%) from College of Engineering Roorkee. He is a life time member of various professional bodies like IEEE, CSI and ISTE etc.

His current research interests include different aspects of Digital Image Processing, Mobile Computing, Artificial Intelligence, Pattern Recognition; Multimedia Systems etc.He has published more than 30 research papers in various reputed national / international journals.

Nitish Pathak received his M.Tech in Computer Science and Engineering (2010) from SIET, Meerut. He did his MCA in 2005 from UPTU, Lucknow and completed M.Phil. (Computer Science) in 2007 from Madurai Kamraj University, Madurai, Tamilnadu, India. Now he is Assistant Professor of Computer Science department in BVICAM, GGSIPU, Delhi (INDIA).He has More than Nine Years of extensive Corporate as well as Professional Teaching Experience at Graduate and Post Graduate level. He got Directorate Award (Three times on teacher's Day) for best teaching performance since 2007 to 2009, at ABES Engineering College, Ghaziabad, India.

His current research interests include different aspects of trusted operating Systems; Object oriented based Forward Engineering and Data Mining. He is a life time member of various professional bodies like IEEE, CSI and ISTE etc.

He has published more than 15 research papers in various reputed national / international journals.

How to cite this paper: Nitish Pathak, Girish Sharma, B. M. Singh,"Experimental Analysis of SPF Based Secure Web Application", IJMECS, vol.7, no.2, pp.48-55, 2015.DOI: 10.5815/ijmecs.2015.02.07