Hybrid Analysis and Design Model for Building Web Information System

Dayanand Ingle[1] and Dr. B.B. Meshram[2]

[1] Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Navi Mumbai, Maharashtra, India

[2] Department of Computer Technology, VJTI, Mumbai, Maharashtra, India

Abstract

The World Wide Web has become a highly adopted platform for Web applications. The traditional structured systems analysis and design as well as object oriented analysis and design (UML) cannot capture all the requirements of the e-Commerce Applications. There are various web-engineering techniques used to capture the requirements of web Applications, but there is no silver bullet to capture all the characteristics of the E-Commerce applications. The notations and diagrams of the particular method cannot produce the good modeling of web applications; therefore we tried to assimilate all the goodies of the Web Engineering techniques. By considering the strengths and weaknesses of various Web engineering techniques, we have formulized a conceptual framework by integrating, investigating various web analysis and design methods such as OOHDM, WSDM, WebML, W2000 (HDM), OOWS, HERA and WebSA for building the Web Information Systems. We have integrated the various features such as notations, requirement modeling, workflow, content modeling, hypertext modeling, presentation modeling, authoring systems and navigational requirements, customization modeling and mapping of analysis and design of Web applications to the implementation.

Keywords: *OOHDM, WSDM, WebML, W2000 (HDM), OOWS, HERA, WebSA.*

1. Introduction

Web-based systems are those systems that use the Internet, intranets, and extranets. The Internet is a worldwide collection of interconnected networks. An intranet is a private network inside a company using web-based applications, but for use only within an organization. An extranet is a private network that allows external access to customers and suppliers using web-based applications. Structured analysis and design is useful tool for gathering the requirements of every application through interviews, questionnaires, records and, observations. These requirements can be captured by EER Diagram, dataflow diagram, flowchart and decision trees in[1]. And can be mapped with the data structure design of the application and software architecture, however the object oriented features cannot be captured by the SSAD which lead to the various object oriented methods like Ivar Jacobson's use case analysis, Raumbagh's OMT and Booch's methods, However these methods are integrated together and the method is known as UML[2] for object oriented analysis and design. A drawback of UML is that it lacks the capability to represent uncertainty. With use case approach, the only solution that can be developed will be one that can be automated with an application. This virtually ignores the relevant business issues that may be all or part of the problem as well.

UML requirements tend to include a lot of technical language since they are describing technical features. Finally, another problem with UML requirements is that they tend to focus on business transaction at a time. Since they are most often derived from Use Cases, the requirements are documented with an eye toward that one transaction and often ignore the business workflow surrounding it. Without a highly structured reuse analysis, it is often possible to end up with highly efficient transactions that contain a lot of business redundancy between them. Thus UML alone is not sufficient for capturing the requirements of the web applications. Web Engineering methodologies provide a number of benefits compared to ad hoc development: Web Engineering methodologies guide through the process of software-creation, provide a complete documentation, take into consideration the users' needs, and ensure quality. Further, by forcing modularity and support re-use, the designed applications are more flexible, robust, and reliable. Through the homogenous structure of engineered Web-applications, complexity decreases and maintenance eases.. In addition to common SE-methodologies, Web Engineering methodologies are specially designed to meet the requirements of the Web. But the particular method is suitable for capturing the certain requirements and characteristics of the applications. It gives rise to various modeling methods, which follows different paradigms, depending on their origin and focus. Modeling methods are categorized into four types[3].

i) **Data Oriented Methods**: These are originated from the field of database systems, and mainly based on ER model. The primary focus of these methods is the modeling of database-driven web applications. For Examples: Relationship management methodology (RMM)- 1995, Hera-2001, Web-Modeling Language (WebML,-1999).

ii) **Hypertext-Oriented Methods**: HOM Focuses on hypertext character of web applications. These methods include Hypertext Design Model (HDM)-1993, later HDM extended to W2000 and HDM-lite-1996, WSDM-1997.

iii) **Object Oriented Methods**: These methods are based on either OMT or UML. This category includes Object Oriented Hypermedia Design Method (OOHDM)-1996, UML based Web Engineering (UWE)-2000, Object oriented web solutions (OOWS)-2001, and Object Oriented Hypermedia (OO-H) -2002.WebSA- 2004.

iv) **Software Oriented Metho**ds: It uses techniques that strongly follow classical software engineering. Examples of this category are Web Application Extension (WAE) 1999 and the enhanced version WAE2- 2001.

We tried to integrate the goodies of each method for the web modeling. We do not pretend that our method will sort out each and every problem, it cannot be silver bullet, but it is a feasible and realizable solution.

The paper is organizes as below: Section 2 deals with literature survey required for the proposed method, Section 3 describes our proposed model, Section 4 deals with case study of Web Based Conference Management System to produce the quality web information systems, Lastly we conclude our result in section 5

2. Literature Survey

Software requirement specification documents provide the details for the requirements, analysis and design of the systems, however for modeling the web applications, we have considered the following characteristics of the web applications.

Content Modeling: Content modeling is aimed at transferring the information and functional requirements determined by requirement engineering to a model. The idea of a content model is new, but it has similarities to both a database design and an object model. The purpose of both of these is to provide a foundation for the logic of the operation. The same applies to a content model [6].

Hypertext Modeling: Also known as navigation modeling – is to specify the navigability through the content of web applications. Navigation of the site is specified thru links. Links can be defined between the units inside a single page, between units placed in different pages, and between pages. The information carried along a link is called navigation context, or simply context. Links that carry context information are called contextual links, whereas links that have no associated context information are called non-contextual links. Context information is typically necessary to ensure the computability of units.

Presentation Modeling: Presentation modeling deals with user interface and look and feel of web application. The main objective of presentation modeling is aimed at designing the structure and behavior of the user interface.

- In presentation modeling the model elements are described on three hierarchical levels A presentation page: describes page presented to the user as a visualization unit.
- A presentation unit: serves to group related user interface elements.
- A presentation element: presentation element represents a node's set of information & can include text, image, audio etc.

Customization Modeling: It is used to represent context information, and the adaptations derived from it. Mostly customization modeling is intermingled with content, hypertext and presentation models[14].

2.1 Web Modeling Methods

In order to propose our hybrid model we have studied the following methods:

OOHDM[1], WSDM[4], WebML[5], [6], W2000 (HDM) [7], OOWS[8], HERA[15], WebSA. These methods are the superset for the other remaining methods.

The Hypertext Design Model (HDM)[16] is a model for the structured design of hypertext-applications. Therefore it describes a design model rather than a process model. In contrast to HDM, OOHDM offers a clearly defined procedure for the development of hypermedia-applications. OOHDM consists of the four steps as shown in figure 1: conceptual design, navigational design, abstract interface design and implementation, which have to be executed according to an iterative and incremental process model.

Fig 1: OOHDM Activities

The methodology of *OOHDM[17]* comprise four different activities:

(1) Requirement Gathering: This step involves gathering stake holder's requirements with the help of use cases, which are represented using User Interaction Diagram.

(2) Conceptual Design: The conceptual model corresponds to a traditional object-oriented model and describes design entities using UML[18] notations.

(3) Navigational Design: In this phase a navigational model is specified, which can be seen as a view (in the sense of databases) of the *conceptual model*. The navigation model consists of a *navigational class schema* and a *navigational context schema*.

(4) Abstract Interface Design: This step deals with the user interface objects. The *Abstract Data View* (*ADV*) design approach is used for describing the user interface.

(5) Implementation: It deals with the mapping of the designed application to implementation of the application.

Advantages: The modularity and reusability of design concepts is relatively high due to the high degree of abstraction found in the resulting models. OOHDM have a well-guided design-phase with expressive visual models and diagrams. They are useful for Web applications with a structured information domain behind.

Disadvantages: Their problem is the weak mapping from design (Visual Models and diagrams} to the application implementation. OOHDM considers several important aspects of Web-applications but is lacking system support that adequately corresponds to the basic principles of the Web. This implies many problems such as the implementation in heterogeneous environments, the integration of distributed objects and artifacts. The major drawback is the weak mapping functionality; this results in poor construction of the application and serious maintenance issues. The generality of the modeling approach tend to lead to a higher complexity. For example the method explicitly supports the use of design patterns but does not support the retrieval of patterns nor does it provide assistance for automating the implementation of reusable design artifacts.

2.1.2 WSDM (Web Site Design Method)

The Web Site Design Method (WSDM)[4] is a user-centered method for web application design[21]. The design process consists of four different phases: Mission Statement Definition, Audience Modeling, Conceptual Design (divided into Task Modeling and Navigation Design) and Implementation design as shown in figure 2. During the Task Modeling phase, all the information and functional requirements obtained during the Audience Modeling phase are elaborated to generate two main models: the Task Model and Object Chunks. In particular, for each requirement a task is defined, and then divided: into elementary tasks, in a hierarchical structure

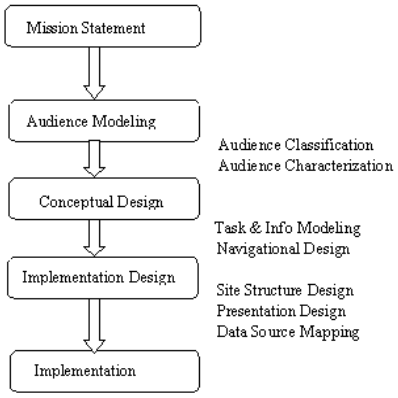

Fig 2: Different Phases of WSDM

Advantages: Navigation track of WSDM is more powerful for navigation modeling, for the detailed structure of the web sites.

Disadvantage

WSDM is a user-centered methodology [4], having a strong usability design phase. Due to its high level of abstraction, its weakness lies in the area of data and structural design, and dynamics.

2.1.3 WebML

WebML (Web Modeling Language) is a visual notation for designing complex data-intensive Web applications[19]. It provides graphical, yet formal, specifications, embodied in a complete design process, which can be assisted by visual design tools, like WebRatio.

The five perspectives of WebML are:

1) Structural Model: It is typical data conceptual model and based on ER-Model (ERM), UML, and ODMG. But preferred model by authors of WebML is UML

2) Derivation Model: It describes how the structure can be extended with derived data, to introduce redundant information**.** In other words it is similar to VIEWS in database modeling. Like VIEW in Oracle or MySQL. For each page there is one abstract Table of data's. But it is merged from other tables. It uses WebML-OQL (WebML-Object Query Language)[20].

3) Composition Model: Describes the allocation of content to application pages.

4) Navigational Model: Shows the navigation between pages using links (context, non-context) Models how the user moves on the web.

5) Presentational Model: Presentation modeling is concerned with the actual look and feel of the pages identified by composition modeling. WebML pages are rendered according to a style sheet[22].

Advantages of WebML:

- WebML guarantees a model-driven approach to Web site development, which is a key factor for defining a novel generation of CASE tools for the construction of complex sites
- WebML has supporting advanced features like multi-device access, personalization, and evolution management.
- Can capture the detailed contents

Disadvantages: WebML provide a set of methods and supporting tools for a systematic design and development of Web applications addresses different concerns using separate models (content, navigation, presentation, business logic, etc.), and provide model compilers that produce most of the logic and Web pages of the application from these models. However, these proposals also have some limitations, especially for exchanging models or representing further modeling concerns, such as architectural styles, technology independence, or distribution in[23].

2.1.4 W2000 (HDM)

W2000 originates from HDM but also borrows principles from UML to support the concept of business processes. A W2000 model comprises some models. Each model has a predefined package, which acts as a root for the hierarchy of other packages and elements that belong to the model[24]. This is implemented through the abstract class element with package and all the W2000 Elements as sub-classes as shown in figure 3. Elements belong to the package in which they are defined, but are rendered in diagrams, which also belong to different packages [9].

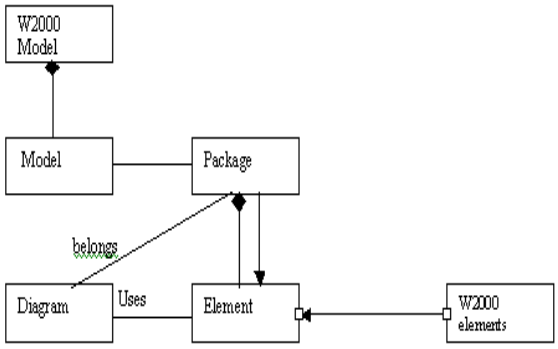

Fig 3: W2000 hierarchy

A complete W2000 model is organized in four models:

i) Information: Defines data used by application and perceived by the user
ii) Navigation: Need to define how the contents are organized for fruition. ``
iii) Services: Specifies the control
iv) Presentation: How data and services are presented to the user.

Advantage: An advantage of W2000 is its ability to decompose hypertext-applications into fine-grained artifacts for reuse while considering ensuing structural relations.

Disadvantage:

- The applicability of W2000 is severely limited by its methodology.
- The developer is subject to a high cognitive load caused by an unsatisfactory process model that neither supports artifact reuse nor allows for modeling artifacts using an object-oriented paradigm. The mapping of a hypertext-application designed with HDM to a Web-application is difficult due to the lack of an appropriate support.

2.1.5 OOWS

Object-Oriented Web-Solutions Modeling (OOWS) is a UML-based Web Engineering method to develop web applications that is strongly based on conceptual modeling techniques. OOWS is the extension of OO methods used for the web development as shown in figure 4. In System Specification Step the conceptual schema is build to represent the applications requirement[25]. The modeling tool used by the method allows the specifizacation of structural and functional requirements of dynamic applications by means of a set of models. These models are as follows

Structural Model: It defines the classes, operations and attributes (structure) and relationship between classes (specialization, association and aggregation) by means of class diagram[26].
Dynamic Model : It describes the different valid object-life sequence for each class of the system using State Diagrams. Also in this model object interactions (communications between objects) are represented by Sequence diagrams[27].
Functional Model :that captures the semantics of state changes, using a textual specification derived from an OO formal specification [10] to define service effects[28].

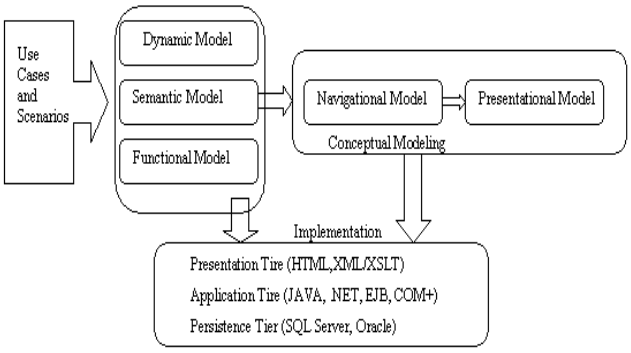

Fig 4 : OOWS Methodology

Advantages: OOWS is best for static as well as dynamic Process modeling

Disadvantages: It has Weak mapping functionality

2.1.6 HERA

Hera is a design methodology for WIS (Web-based Information Systems) with automated presentation generation that is adapted to device capabilities and user preferences. Hera is also used as the name of a model-driven design approach and specification framework focusing on the development of context-dependent or personalized WIS shown in figure 5[6]. The approach distinguishes three main models that specify the generation of hypermedia presentations over available content data: a model for the content, a model for the hypermedia navigation construction, and a model for the presentation construction enable the creation of a hypermedia-based view over the content.

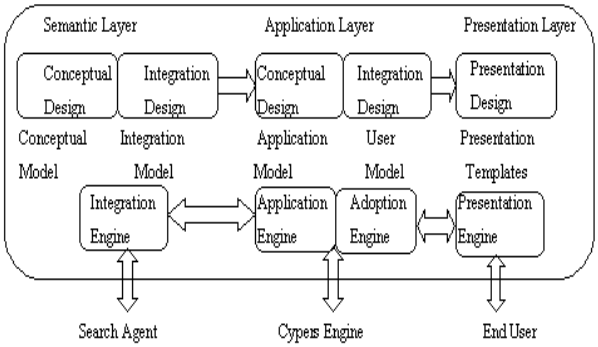

Figure 5: Hera Methodology

Advantage: The integration of WIS designs for hypermedia and web personalization or presentation generation with user adoption makes this characteristic more powerful.

Disadvantage: The applicability of Hera is severely limited by its methodology.

- The developer is subject to a high cognitive load caused by an unsatisfactory process model that neither supports artifact reuse nor allows for modeling artifacts using an object-oriented paradigm.
- The mapping of a hypertext-application designed with Hera to a Web-application is difficult due to the lack of an appropriate support.

2.1.7 WebSA

Web software architecture is a model driven approach for applications, which is based on MDA (Model driven approach) paradigm. In WebSA methodology it proposes a development process made up of a set of UML models used to represent web applications as shown in figure 6.
The view model shows the links among the different views (regarded as a set of artifacts created during the software development process) that make up web application software architecture. In WebSA, the web application model is made up of 8 views, further grouped in viewpoints. A viewpoint is a set of views that share concerns. In the following Fig. 6 we can observe a UML diagram that depicts the set of viewpoints and views in WebSA, as well as the relationships among them.
The requirements viewpoint gathers the information needed to specify the system: in particular the set of use cases scenarios (functional requirements view) and quality scenarios (non-functional requirements view) are captured. Departing from the requirement-engineering phase, the functionality of a web system is defined by means of a Functional Viewpoint. That functionality is captured by means of the corresponding views defined by the web engineering community for web applications. In particular, the conceptual view captures the structure of the information system that lies behind the application. The navigation view specifies the interactions that the user may perform in order to step through the different application scenarios. The presentation view is concerned with the general appearance of the application and the functionality associated with this appearance. Finally, the process view gathers process activities and flows [11]

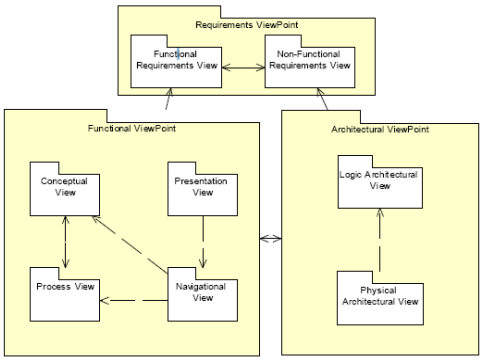

Fig. 6: Web Software Architecture

Also, WebSA includes an architectural viewpoint to explicitly address the architectural issues.

Departing from non-functional requirements a set of architectural patterns can be inferred to gather the logical and physical architecture views. The first one gathers the set of logical components (subsystems, modules and/or software components) and the relationships among them. The last one describes which are the physical components that integrate the final representation (clients, servers, networks, etc…). Note that the interdependency between the functional and the architectural viewpoints expressed with the double arrow in Fig 6. Implies that on some occasions it will be advisable to take architectural decisions based on functional features and vice versa. The existing relationships among the different WebSA views are formalized in a common metamodel that permits the establishment of traceability between the elements in the different views. WebSA, aiming at being compliant with MDA, defines a conservative extension of the UML metamodel in the context of a UML profile.

An overview of the WebSA Approach: WebSA is a proposal whose main target is to cover all the phases of the Web application development focusing on software architecture. It contributes to cover the gap currently existing between traditional Web design models and the final implementation. In order to achieve this, it defines a set of architectural models to specify the architectural viewpoint, which complements current Web engineering methodologies such as. Furthermore, WebSA also establishes an instance of the *MDA Development Process,* which allows for the integration of the different viewpoints of a Web application by means of transformations between models [2].

WebSA Architectural Models: The WebSA approach proposes three architectural models:

Subsystem Model (SM): determines the subsystems that make up our application. It is mainly based on the classical architectural style defined in – the so called "layers architecture" – where a layer is a subsystem encapsulating a certain level of abstraction. Furthermore, it makes use of the set of architectural patterns defined in that determine which is the best layer distribution for our system.

Configuration Model (**CM**): defines an architectural style based on a structural view of the Web application by means of a set of Web components and their connectors, where each component represents the role or the task performed by one or more common components identified in the family of Web Applications.

An overview of the WebSA Approach: WebSA is a proposal whose main target is to cover all the phases of the Web application development focusing on software architecture. It contributes to cover the gap currently existing between traditional Web design models and the final implementation. In order to achieve this, it defines a set of architectural models to specify the architectural viewpoint, which complements current Web engineering methodologies such as. Furthermore, WebSA also establishes an instance of the *MDA Development Process,* which allows for the integration of the different viewpoints of a Web application by means of transformations between models[2].

Integration Model (**IM**): merges the functional and the architectural views into a common set of concrete components and modules that will make up the Web application. This model is inferred from the mapping of the components, which are defined in the configuration model, the subsystem model and the models of the functional view. The formalization of these models is obtained by means of a MOF-compliant repository metamodel and a set of OCL constraints (both part of the OMG proposed standards) that together specify (1) which is the semantics associated with each model element, (2) which are the valid configurations and (3) which constraints apply.

WebSA Development Process: The WebSA Development Process is based on the *MDA development process*, which includes the same phases as the traditional life cycle (Analysis, Design, and Implementation). However, unlike in the traditional life cycle, the artifacts that result from each phase in the MDA development process must be a computable model. These models represent the different abstraction levels in the system specification and are, namely:

(1) Platform Independent Models (PIMs) defined during the analysis phase and the conceptual design,

(2) Platform Specific Models (PSMs) defined in the low-level design,

(3) Code.

WebSA Architectural Models: The WebSA approach proposes three architectural models:

Subsystem Model (SM): determines the subsystems that make up our application. It is mainly based on the classical architectural style defined in – the so called "layers architecture" – where a layer is a subsystem encapsulating a certain level of abstraction. Furthermore, it makes use of the set of architectural patterns defined in that determine which is the best layer distribution for our system.

Configuration Model (**CM**): defines an architectural style based on a structural view of the Web application by means of a set of Web components and their connectors, where each component represents the role or the task performed by one or more common components identified in the family of Web Applications.

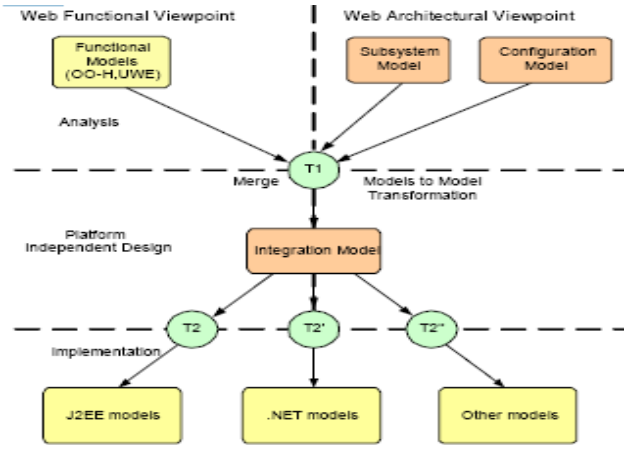

Fig. 7: WebSA Development Process

In order to meet these requirements, the WebSA development process establishes a correspondence between the Web-related artifacts and the MDA artifacts. Also, and as a main contribution, WebSA defines a transformation policy driven by the architectural viewpoint, that is, an "architectural-centric" process. Fig. also shows how in the analysis phase the Web application specification is divided vertically into two viewpoints. The Web functional models provided by approaches such as OO-H or UWE , while the Subsystem Model (SM) and the Configuration Model (CM) define the software architecture of the Web Application gives the functional-perspective. In the analysis phase, the architectural models are based on two different architectural styles to define the Web application. As it is defined in , "an architectural style is independent from its realization, and does not directly refer to a concrete application problem it is intended to solve". In this way, these models fix the application architecture orthogonally to its functionality, therefore allowing for their reuse in different Web applications. The PIM-to-PIM transformation (T1 in Fig.7) from analysis models to platform independent design models provides a set of artifacts in which the conceptual elements of the analysis phase are mapped to design elements where the information about functionality and architecture is integrated. The model obtained is called Integration Model (IM), which merges in a single
architectural model the information gathered in the functional viewpoint with the information provided by the Configuration and Subsystem Models. It is important to note that the Integration model, being still platform independent, is the basis on which several transformations, one for each target platform (see e.g. T2, T2' and T2'' in Figure 7.), can be defined. The output of these PIM-to-PSM transformations is the specification of the Web application for a given platform. The inclusion of an architectural view in this process plays a pre-eminent role for the completion of the specification of the final Web application, and drives the refinement process from analysis to implementation

Advantages: Summarizing, the advantages of extending the architecture for MDA models can be named as follows:

- Possibility of capturing the non-functional requirements in order to improve the quality of the resulting Web Applications.
- The reuse of the architecture models for different systems.
- A more rigorous mapping between the domain model and the different component view models.
- A better quality of the generated code, because it permits the definition of a generation mechanism where the functional and architectural parts are combined.

Disadvantages: Disadvantages related to scalability, platform-independence or security. Architecture models are fundamental in an MDA process, which consists in building and transforming platform-independent models and platform-specific models of the Web application

3. Proposed Model

All the goodies of various WEB methodologies and selection criteria are integrated together and model is known as hybrid model. The goodies of every methodology are given below:

We propose that each web application should be divided into following parts

I) Hybrid BBM-DRI Model for Web Engineering
II) Security Model
III) Deployment Model
IV) Communication Model

I) Hybrid BBM-DRI Model For Web Engineering

We propose the following activities to initialize the work and these activities can be repeated again and again for the correctness of the workflow:

1. Decide the statement of the problem
2. Define the static and dynamic characteristics of the applications
3. Decide the user view and administration view
4. Construct the workflow of the systems.

Modeling Paradigm: hypertext + database paradigm from the concern methodology such WSDM and WebML can

considered as in some Web applications data must be stored into database for the administration or decision making and hypertext for accessing the sides by the users.

Requirement Modeling: Structured Systems analysis methods and its software requirement specification document can be used for capturing the basic requirements of the business. OOHDM + W2000 are rich in requirement modeling.

Content Modeling: WebML can capture the detailed contents. The most outstanding features of WebML are (1) the consideration of workflows where more than one actor is involved (and therefore how the corresponding hyper views may be synchronized) (2) the possibility to control certain types of activity flows by the assignment of objects.

Hypertext Modeling: HERA framework implies a stepwise approach to get from data retrieval to presentation generation at the level of instances. The integration of WIS designs for hypermedia and web personalization or presentation generation with user adoption makes this characteristics more powerful, but the data retrieval is query based, so multimedia data retrieval strategy must be incorporated into it as proposed in our creation.

Navigation Modeling: WebML has explicitly defined a class framework that supports the defined structures. UWE, WebML and OO-H define a new conceptual process diagram. And both structure and navigation through processes. OOHDM extends both its structure and navigational model to support the new concepts. Navigation track of WSDM and OOHDM's Navigation diagram are more powerful for navigation modeling, for the detailed structure of the web sides we propose the inbound and outbound links to the pages of the web sides. However we propose the navigation mapping with the help of web structure diagram to show the links with various web sides and search engines.

Functional Modeling: OOWS is best for static as well as dynamic Process modeling and data modeling is integrated with it. The Concur Task Tree notion provides WSDM with capabilities such as task decomposition, task synchronization, and deactivation of tasks, information exchange or certain types of activity suspension and transactions; processes cooperativeness is supported in this model.

Other Issues Modeling: Temporal relationships are expressed by means of operators between tasks, and for each of them an object chunk, modeling the information and/or functionality required, is defined in an extended ORM (Object Role Modeling) notation. Persistence of processes and support for multiple processes running at a time are advanced features only considered in OOHDM.

Uncertainty is not captured by any methods. We propose that Uncertainty modeling be represented by statistical modeling,

II) Security Model: Applying Security to Product Life Cycle

Now days there is a necessity of secure software development life cycle especially for web applications. Different parts of the project apply to the different phases of the product development life cycle. The sequence of project mirrors the typical phases of the life cycle. Figure 8 shows how the security implementation corresponds to the phases of a classic product development life cycle. Threat modeling and security assessment apply when you build new Web applications or when you review existing applications. The software life cycle was in use to develop the good software. Now a day's requirement is software development should incorporate the security features. For fixing any problem or for security implementation, if you do it early in the project life cycle will reduce the cost accordingly. Security aspects are related to product development life cycle. IBM reported that the cost to fix an error found after product release was 4 to 5 times as much as one uncovered during design, and up to 100 times more than one identified in the maintenance phase. Research by Stake demonstrated that on average an organization caught only a quarter of its software security holes and had typically seven significant bugs within its enterprise software. Their findings verified that fixing the same defects during the testing phase cost around seven times less than after deployment. They concluded building security into software engineering at the design stage would net a 21% ROSI (Return on IT security Investment); waiting until the implementation stage would reduce that to 15% and at the testing stage, the ROSI would fall to 12%. Integrating security early into the application development lifecycle produces more secure, robust applications at a lower cost.

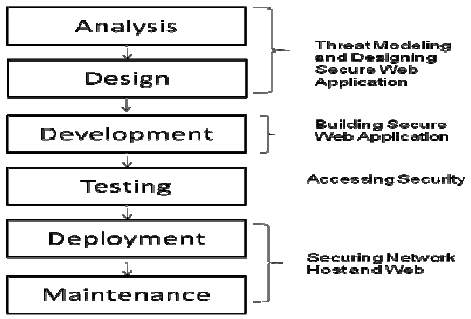

Fig 8: Improving Web Application Security: Threats and Countermeasures as it relates to product lifecycle

To make your application hack-resilient, you need a holistic and systematic approach to securing your network, host, and application. The responsibility spans phases and roles across the product life cycle in[12]. Following table 1 shows the threats in different phases of software development life cycle.

Table 1 Threats in Software Development Life Cycle

Threats in Software Development Life Cycle		
Phases	**Threats**	
Analysis	• Business goals e.g. allowing 24hrs banking via web can yield DOS • System Boundary assessment: Every legitimate system entry or exit point is a threat, as well as other possibly illegitimate access points to a system. • analysis on abuse of privileges by insiders can yield a lot of threat and vulnerability information	
Design	Input Validation	BOF, XSS,SQLIA, Canonicalization
	Authentication	Network eavesdropping, brute force attacks, dictionary attacks, cookies reply, credential theft
	Authorization	Elevation of privilege , disclosure of confidential data, data tampering , luring attacks
	Configuration mgmt	Unauthorized access to administration interfaces ; unauthorized access to configuration store, retrieval of clear text configuration data; lack of individual accountability , over privileged process and service accounts
	Sensitive mgmt	Access sensitive data in storage , network eavesdropping , data tampering
	Session mgmt	Session hijacking, session replay, man in the middle
	Cryptography	Poor key generation or key mgmt, weak or custom encryption
	Parameter manipulation	Query string manipulation , form field manipulation, cookie manipulation , HTTP header manipulation
	Exception mgmt	Information disclosure, denial of service
	Auditing & logging	User denies performing an operation; attacker exploits an application without trace, attacker covers his or her tracks
Development	Input Validation	BOF, XSS,SQLIA, Canonicalization
	Authentication	Network eavesdropping, brute force attacks, dictionary attacks, cookies reply, credential theft
	Authorization	Elevation of privilege , disclosure of confidential data, data tampering , luring attacks
	Configuration mgmt	Unauthorized access to administration interfaces ; unauthorized access to configuration store, retrieval of clear text configuration data; lack of individual accountability , over privileged process and service accounts
	Sensitive mgmt	Access sensitive data in storage , network eavesdropping , data tampering
	Session mgmt	Session hijacking, session replay, man in the middle
	Cryptography	Poor key generation or key mgmt, weak or custom encryption
	Parameter manipulation	Query string manipulation , form field manipulation, cookie manipulation , HTTP header manipulation
	Exception mgmt	Information disclosure, denial of service
Testing	Improper test data: Most published literatures introduce techniques for generating test cases from UML models.	
Deployment	Network Threats	Information gathering, Sniffing or eavesdropping, Spoofing, Session hijacking, Denial of service
	Host Threats	Viruses , Trojan horse and worms, Foot printing, Password cracking , Denial of service
		Arbitrary code execution, Unauthorized access
Maintenance	Improper update of database	

III) Deployment Model:

In web application security is required at many places right from its designing till its deployment. As the web application passes through many tiers there also security is required.

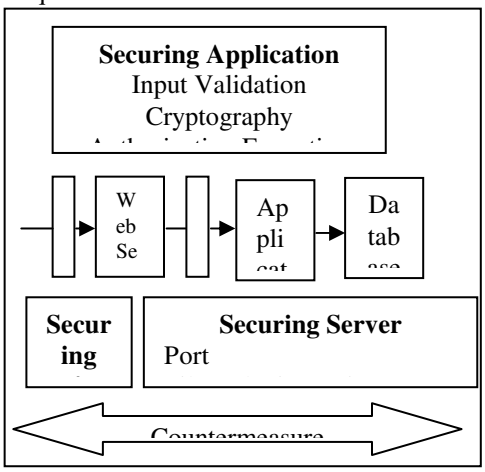

Fig. 9 Scope of the security in web application

The security across the three physical tiers shown in Figure 9.It covers the Web server, remote application server, and database server. At each tier, security is addressed at the network layer, host layer, and application layer. Figure 9 also shows the configuration categories that are uses to organize the various security configuration settings that apply to the host and network, and the application vulnerability categories, which are used to structure application consideration.

III- a) The Holistic Approach

Web application security must be addressed across application tiers and at multiple layers. An attacker can exploit weaknesses at any layer. For this reason, the author has considered a holistic approach to application security and applies it at all three layers. This holistic approach to security is shown in Figure 10. The network security is security of host plus application security. The host security is application security plus Runtime Services and

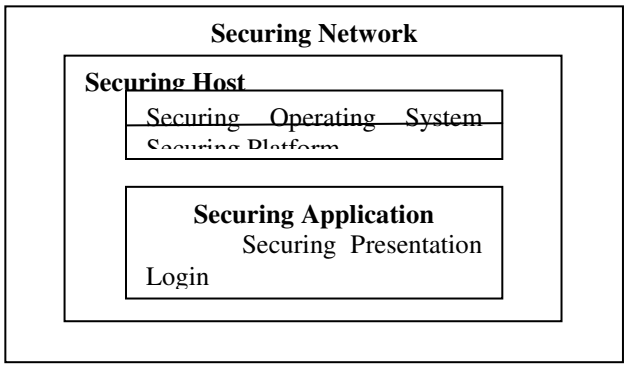

Fig. 10 Holistic approach.

Components, Operating System, Platform Services and Components.The three core elements of a secure network are the router, firewall, and switch. The guide covers all three elements. Table 2 provides a brief description of each element. The network security covers LAN, WAN, MAN and Internet security. Security can be applied to each layer of TCP/IP protocol. Network security is very important for the execution of web program securely.

Table 2 Description of network security tools

Element	Description
Router	Routers are your outermost network ring. They direct packets to the ports and protocols that you have prepared your applications to work with. Insecure TCP/IP protocols are blocked at this ring.
Firewall	The firewall blocks those protocols and ports that the application does not use. Additionally, firewalls enforce secure network traffic by providing application specific Filtering to block malicious communications.
Switch	Switches are used to separate network segments. They are frequently overlooked or over trusted.

common categories that span multiple technologies and components in a layered architecture. These categories are the focus for discussion through the designing, building, and security assessment. Application vulnerability categories Input invalidation, Authentication, Authorisation, configuration management, sensitive data, session management, cryptography, parameter manipulation and exception management.

IV) Communication Model

Web applications present designers and developers with many challenges. The stateless nature of HTTP means that tracking per-user session state becomes the responsibility of the application. As a precursor to this, the application must be able to identify the user by using some form of authentication. Given that all subsequent authorization decisions are based on the user's identity, it

III-b) Securing Your Host

The host includes the operating system and .NET Framework, together with associated services and components. Whether the host is a Web server running IIS, an application server running Enterprise Services, or a database server running SQL Server, there are some methodology that is common across the various server roles and types.

III-c) Securing Your Application

There are some sets of application vulnerability categories to help you design and build secure Web applications and evaluate the security of existing applications. These are

is essential that the authentication process is secure and that the session handling mechanism used to track authenticated users is equally well protected. Designing secure authentication and session management mechanisms are just a couple of the issues facing Web application designers and developers. Other challenges occur because Input and Output data passes over public networks.Preventing parameter manipulation and the disclosure of sensitive data are other top issues. Some of the top issues that must be addressed with secure design practices are shown in Figure 11. Traditionally, security has been considered a network issue, where the firewall is the primary defence (the fortress model) or something that system administrators handle by locking down the host computers. Application architects and developers have traditionally treated security as an afterthought or as a feature to be considered as time permits — usually after performance considerations are addressed. Improving Web Application Security: Threats and Countermeasures the problem with the firewall, or fortress model, is that attacks can pass through network defences directly to the application. A typical firewall helps to restrict traffic to HTTP, but the HTTP traffic can contain commands that exploit application vulnerabilities. Relying entirely on locking down your hosts is another unsuccessful approach. While several threats can be effectively countered at the host level, application attacks represent a serious and increasing security issue. Another area where security problems occur is deployment. A familiar scenario is when an application fails when it is deployed in a locked-down production environment, which forces the administrator to loosen security settings. This often leads to new security vulnerabilities. In addition, a lack of security policy or application requirements that are inconsistent with policy can compromise security. One of the goals of this guide is

to help bridge this gap between development and operations. Random security is not enough. To make your application hack-resilient, you need a holistic and systematic approach to securing your network, host, and application. The responsibility spans phases and roles across the product life cycle. Security is not a destination; it is a journey.

Table 3 Web Application Vulnerability Due to Bad Design

Category	Threats
Input Validation	Buffer Overflow, SQL injection , Cross Site Scripting , canonicalization,
Authorization	Elevation of privilege; disclosure of confidential data; data tampering; luring attacks
Configuration management	Unauthorized access to administration interface; unauthorized access to configuration stores; retrieval of clear text configuration data; lack of individual accountability; over privileged process and service accounts
Sensitive data	Access sensitive data in storage; network eavesdropping ; data tampering
Session management	Session hijacking; session replay; man in the middle
Cryptography	Poor key generation or key management ; weak or custom encryption
Parameter manipulation	Query string manipulation; from field manipulation ; cookie manipulation ; HTTP header manipulation
Exception management	Information disclosure; denial of service
Auditing and logging	User denies performing an operation; attacker exploits an application without trace ; attacker covers his or her tracks

Most security issues come to light only after completion of the development. As a result, security is often managed in an ad-hoc fashion, as an afterthought. Findings indicate that significant cost savings and other advantages are achieved when security analysis and secure engineering practices are introduced early in the development cycle. As the security is incorporated in each and every phase of software development it will reduce all kinds of attacks.

It helps to build hack-resilient applications. A hack-resilient application is one that reduces the likelihood of a successful attack and mitigates the extent of damage if an attack occurs. A hack-resilient application resides on a secure host (server) in a secure network and is developed using secure design and development guidelines.

4. Case Study:

4.1 Workflow of Web Based Conference Management System.

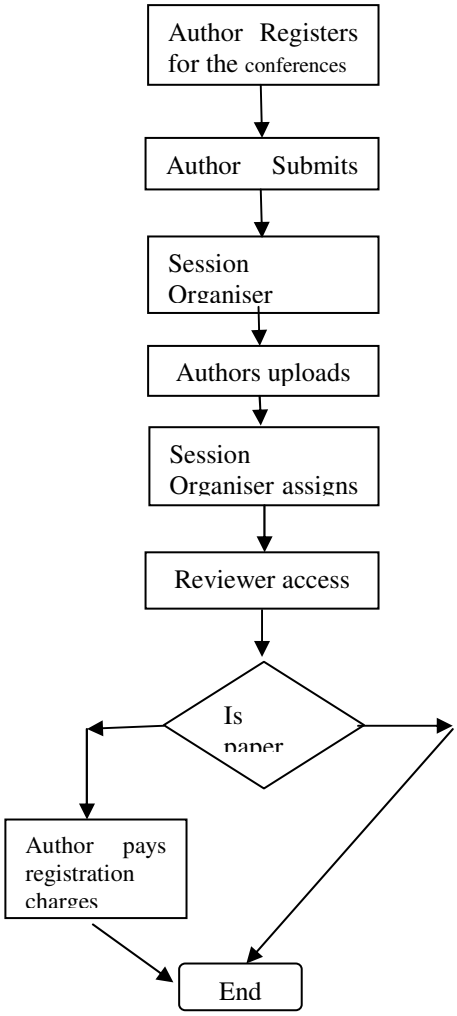

Fig. 11. Workflow diagram for Web Based Conference Management System

4.2 Use Case Diagram

Register: This function is performed by the user. He performs this function when he is logging in for the first time.

Login: This is a function performed by the user.

Make payment: This function is performed by the attendee in order to attend the conference. He may be choose to avail secondary services if he wishes.

Submit paper: This is a function performed by the author when he wants to submit a paper.

Browse review results: It allows the author to go through all the reviews that have been posted.

Bid for papers: A function performed by the reviewer when he wants to review a paper of his choice.

Review paper: Elementary function of the reviewer. He may choose to pass on a recommendation to accept or reject the project.

Makes payment: The function performed by the chairman to avail the conference management system services.

Conference setup: This is done by the chairman in order to setup the conference.
Phase management: The chairman performs this action to decide which phase has to start or stop.

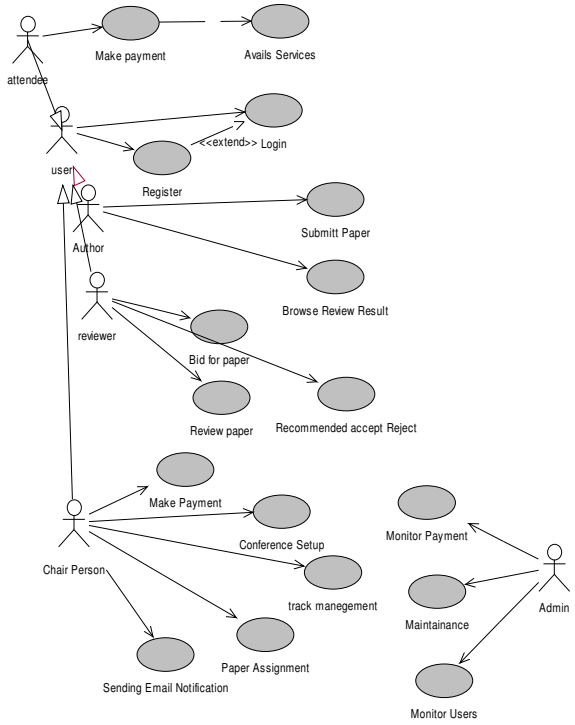

Fig. 12 Use Case Diagram for Web Based Conference Management System

Paper Assignment: This is a function performed by the chairman to assign the various papers coming in to the reviewers.
Sending emails/notifications: Function performed by the chairman to inform all the related personnel about the progress.
Monitor payment: A function performed by the administrator to supervise the monetary transactions.
Monitor users: Similar to transactions, the administrator also supervises the various users using the product.

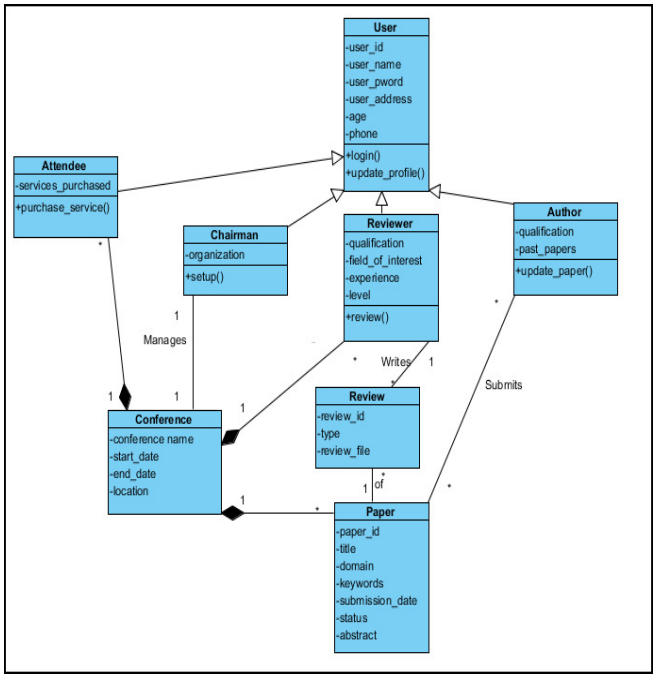

Fig. 13 Class Diagram for Web Based Conference Management System

4.4 Presentation Model

Main Page

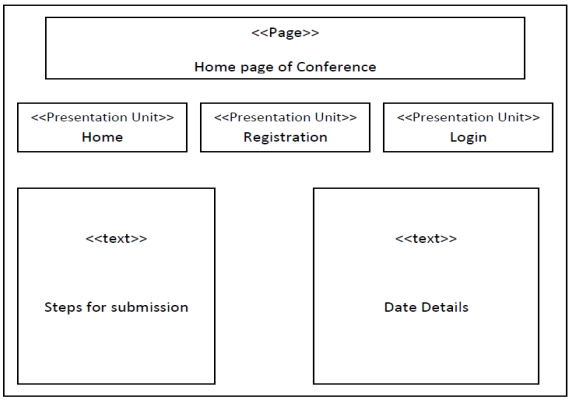

Fig. 14 Presentation Model for Main Page of Web Based Conference Mgmt System

Registration

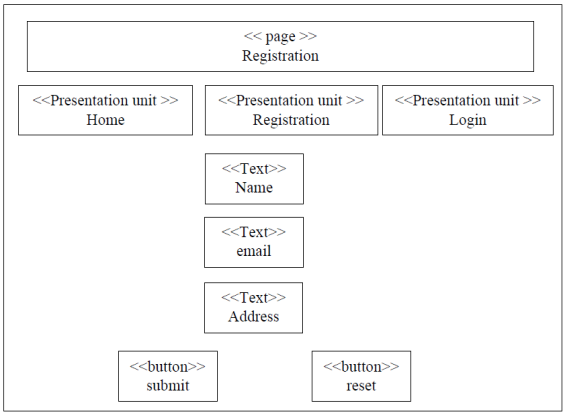

Fig.15 Presentation Model for Registration Page of Web Based Conference Mgmt

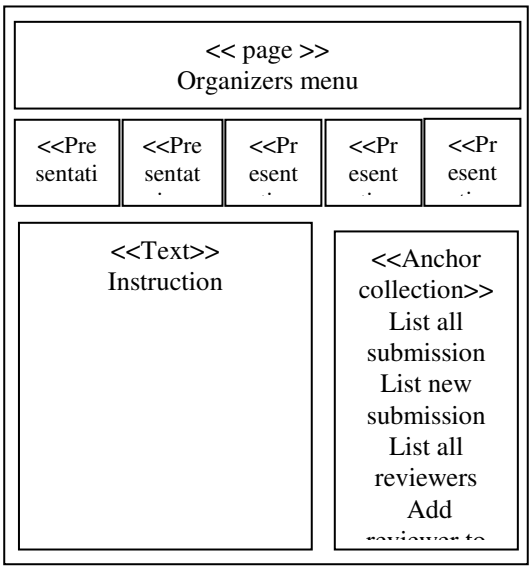

Fig 16 Presentation Model for Registration Page of Web Based Conference Mgmt

Organizer Menu

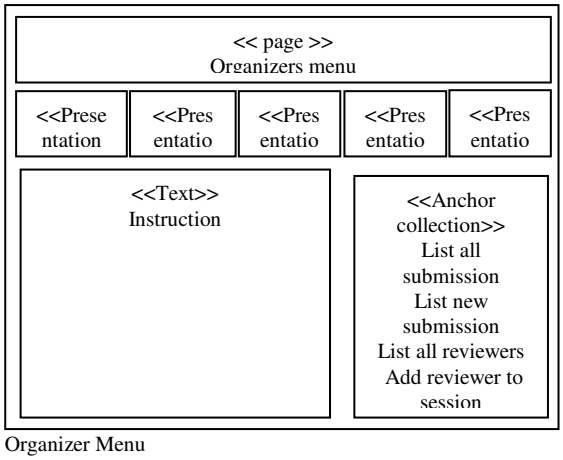

Fig. 17 Organizer Menu

Reviewer Menu

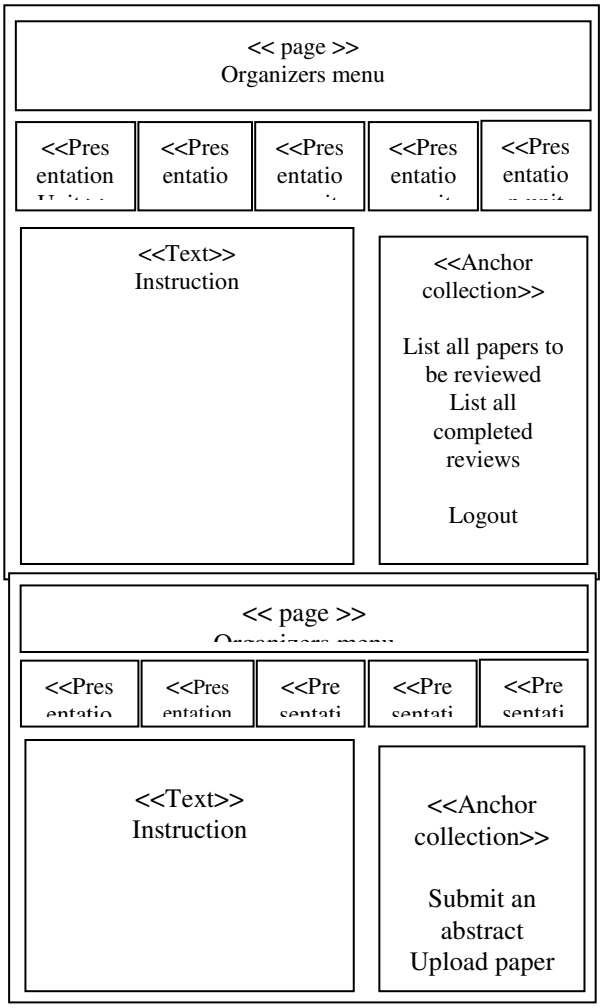

Fig. 18 Reviewer Menu

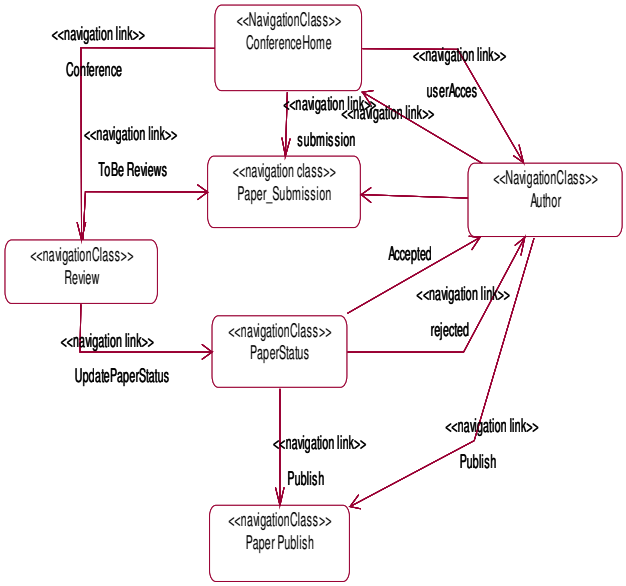

Figure. 19 Hypertext Structural Model of Reviewing System

4.5 Access Model

The hypertext structure model built so far alone is not sufficient to describe how nodes can be reached by navigation. To allow users to navigate to nodes the users need navigation and orientation aids. These are formulated in the form of access structures refining the hypertext structure model.

Figure shows a simplified access model of the PC chair's view specified in the hypertext structure model in our reviewing system. Note that a link's default multiplicity is 1. The PC chair has access to all papers, reviews, and users. To access a specific paper, a unique number is used. Alternatively, the PC chair can search for a paper by title. UWE uses UML stereotypes, i.e., _menu_ (e.g., "Conference"), _index_ (e.g., "ReviewingStatus"), <<query>>, (e.g., "SearchPaperByTitle"), and <<guided tour>>, to specify the menu, index, query, and guided tour access structures.

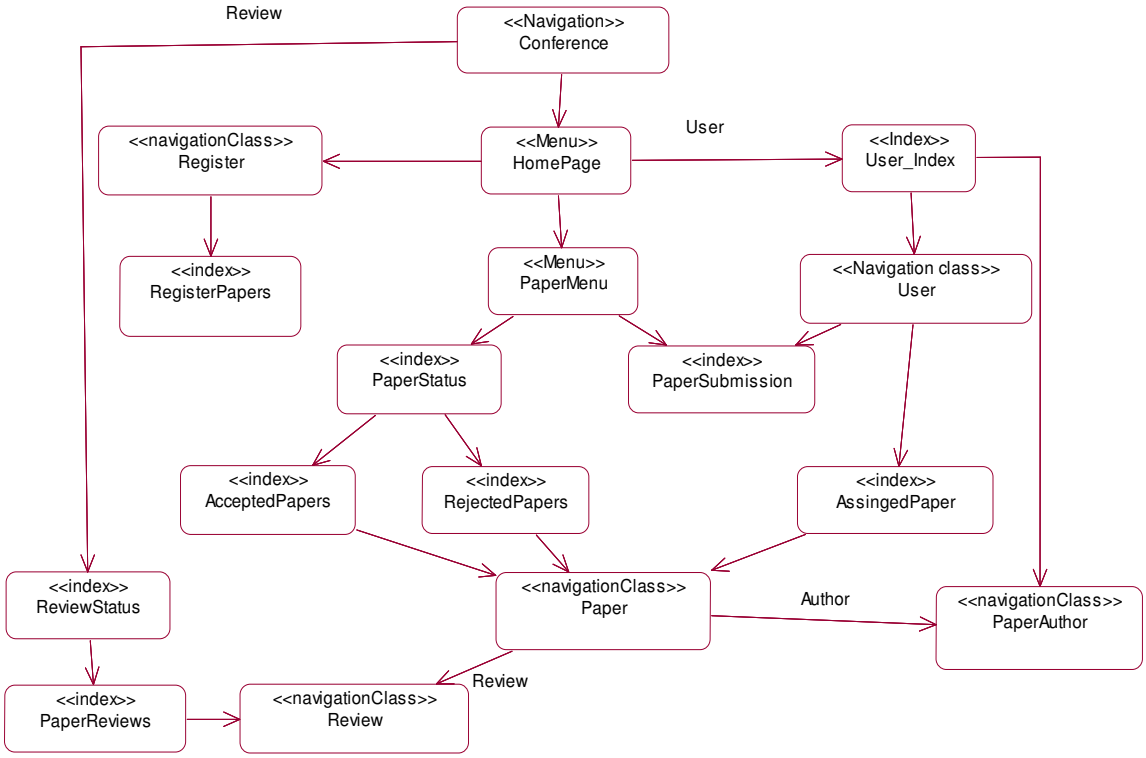

Fig. 20 Simplified Access Model of the Hypertext Structure Model

4.6 Dynamic Model

4.6.1 Sequence Diagram

A **Sequence diagram** is a kind of interaction diagram in UML that shows how processes operate with another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously and as horizontal arrows, the messages exchanged between them, in order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

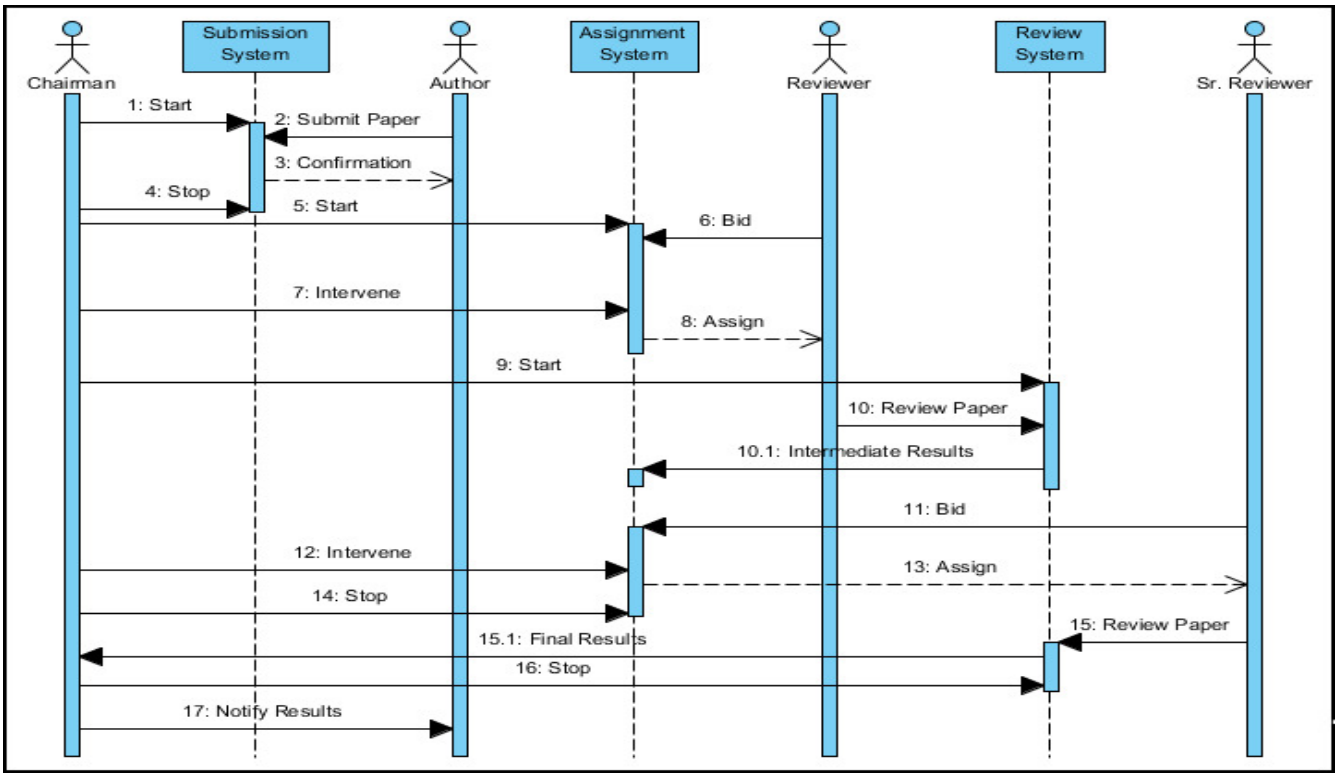

Fig 21 sequence diagram for Web Based Conference Management System

4.6. 2Activity Diagram

Activity diagrams show the flow of activities through the system. The diagrams are read from top to bottom and have branches and forks. We have used branches which describe what activities will take place based on a set of conditions.

Activity Diagram for Conference Setup

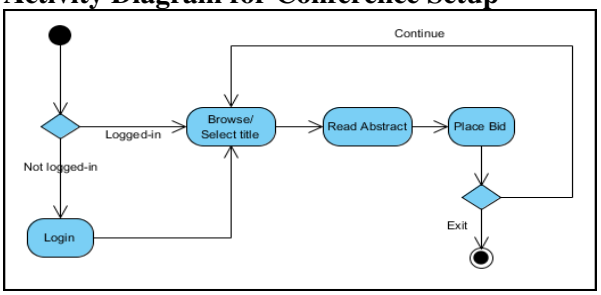

Figure 22 Activity Diagram for Conference Setup

Activity Diagram for Paper

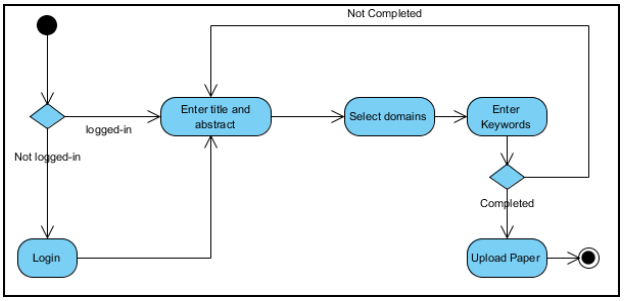

Figure 23 Activity Diagram for Paper Submission

Activity Diagram for Paper Bidding

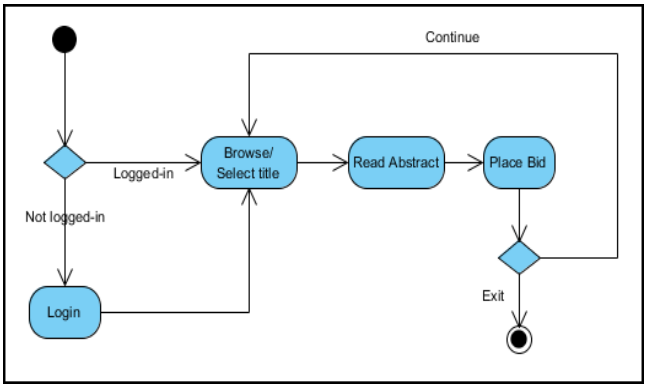

Fig 24 Activity Diagram for Paper bidding

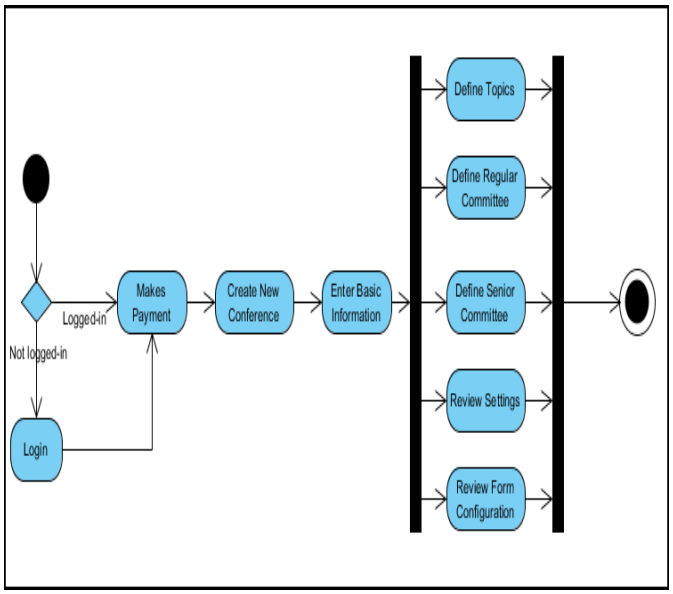

Fig. 25 Activity Diagram for Paper Reviewing

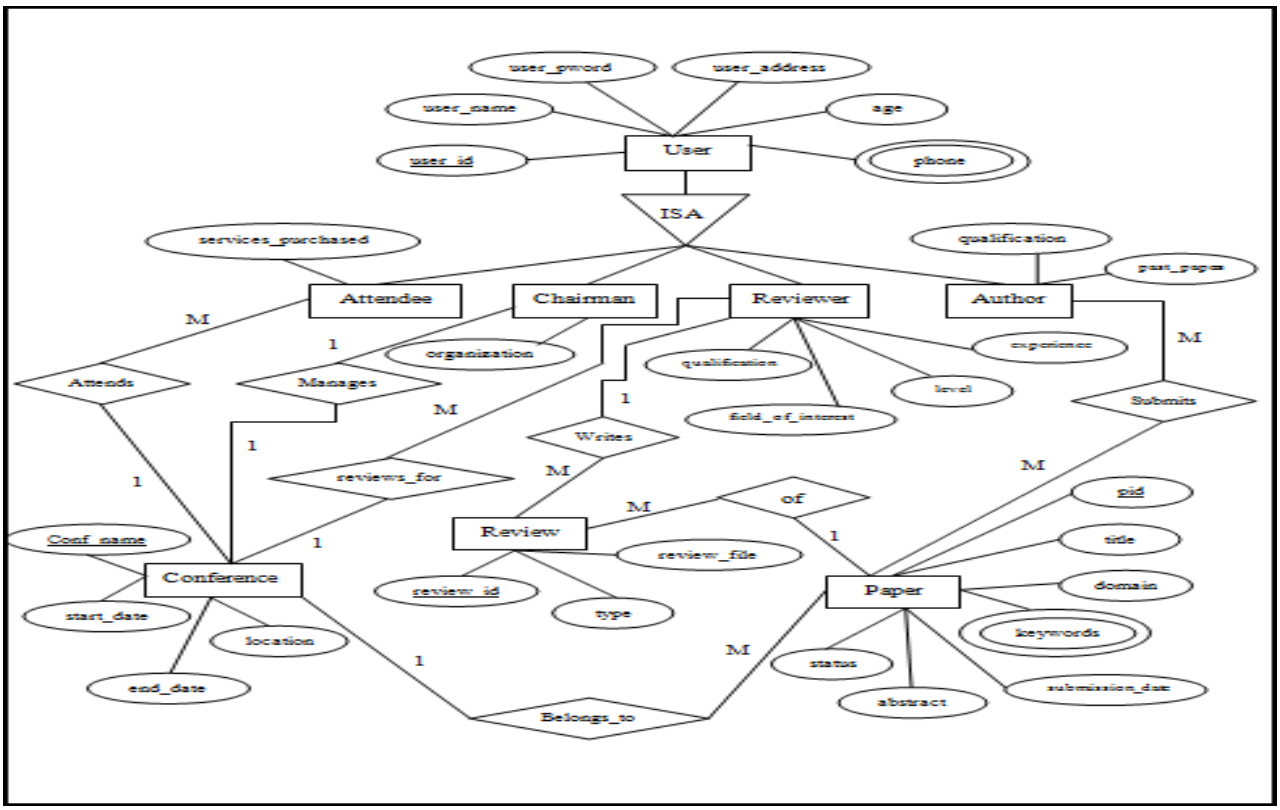

Fig 26 ER diagram for Web Based conference Management System

4.3.2 Component Diagram

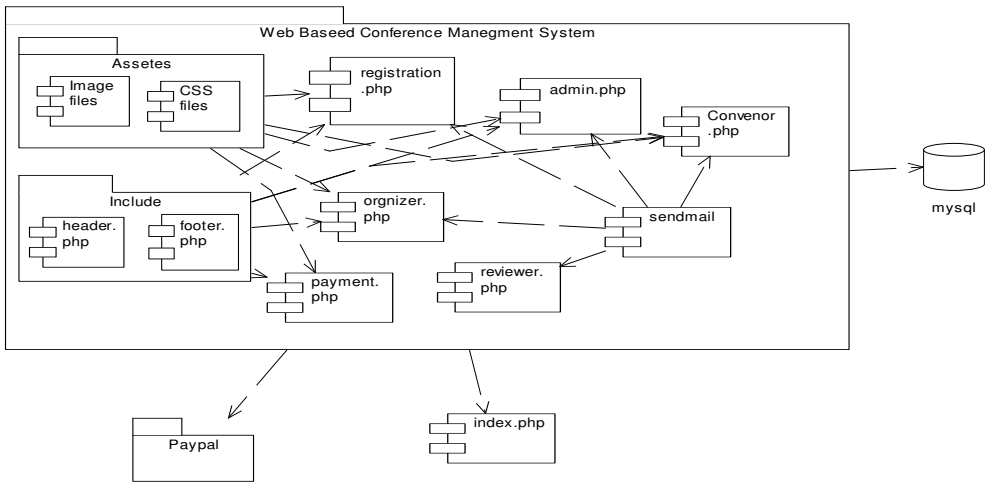

Fig. 27 Component Diagram for Web Based Conference Mgmt System

4.3.3 Deployment Diagram for Web Based Conference Mgmt System

A deployment diagram in the Unified Modelling Language models the physical deployment of artefacts on nodes. The nodes appear as boxes, and the artefacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers. Basically there are three layers one is clients, second is application server and third is database server.

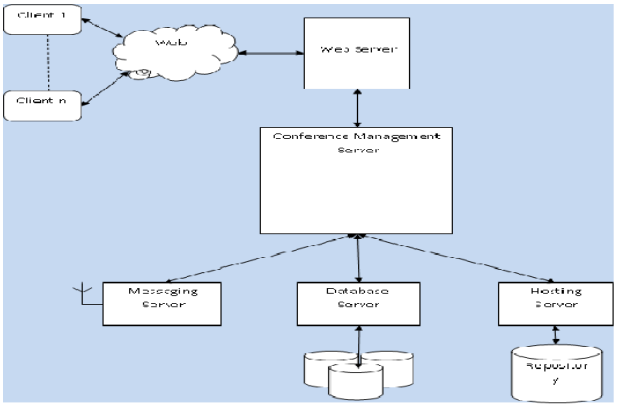

Fig. 28 Deployment Diagram

4.3.4 Security Model: Web Based Conference Management System

The *analysis phase* answers the questions of *who* will use the system, *what* the system will do, and *where* and *when* it will be used. An analysis strategy is developed to guide the project team's efforts. Such a strategy usually includes an analysis of the current system and its problems, and then ways to design a new system (called the to-be system).

Threats Identified in Development of Web Based Conference Management System

Table 4 Threats in Software Development Life Cycle

Threats in software development life cycle		
Phases	Threats	
Analysis	Business goals e.g. allowing 24hrs payment of fees via web can yield DOS Privileges can be given to convenor and organiser account only so that anyone else can not misuse those privileges.	
Design	Input Validation	During designing of online form do proper validation of name , email, comment section, abstract section , file upload etc. Other wise this sections can be used by attacker for input validation
	Authentication	Only authorised user should get access to website otherwise hacker can perform any of the following activity Network eavesdropping, brute force attacks, dictionary attacks, cookies reply, credential theft .
	Authorization	Identify all roles (admin, convenor, organiser, reviewer) properly then give privileges according to the roles.
	Sensitive mgmt	Access to sensitive data like payment details, papers should be given to convenor and organisers only , other wise their is possibility of network eavesdropping , data tampering
	Session mgmt	Once the user login till that user is going to logout his session should get maintain properly.
	Cryptography	For sorting password in database we need encrypted password so keys for encryption and decryption should be strong so that password will get revel.
	Parameter manipulation	Parameters passed by clients like login name, password, email should be manipulated properly by removing suspicious special characters.
	Exception mgmt	Error should get handled properly , it should not reveal any program information

Implementation	Input Validation	Validate all text boxes, remove special characters like < , >, from otherwise SQLIA,XSS and BOF is possible
	Authentication	Instead of using only login name and password use 3D password otherwise hacker can get access to your database.
	Authorization	Identify all roles (admin, convenor, organiser, reviewer) properly then give privileges according to the roles. Make proper use of grant command.
	Sensitive mgmt	Access to sensitive data like payment details, papers should be given to convenor and organisers only , otherwise there is possibility of network eavesdropping , data tampering
	Session mgmt	Once the user login till that user is going to logout his session should get maintain properly using Http Session Objects or any other session tracking technique. Otherwise attacker hijack any session.
	Cryptography	For sorting password in database we need encrypted password so keys for encryption and decryption should be strong so that password will get revel. Use proper DES or AES systems.
	Parameter manipulation	Parameters passed by clients like login name, password, email should be manipulated properly by removing suspicious special characters and spaces should get replace with hexadecimal symbols.
	Exception mgmt	Error should get handled properly , it should not reveal any program information
Testing	Improper test data: Most published literatures introduce techniques for generating test cases from UML models, such as sequence diagrams or activity diagrams. Any diagram-based test method is based on path traversing. A run driven by one test case may not detect the modelled threats, so various runs taking different paths may be necessary to find a path which can activate the threat behavior. Killing criteria should get defined properly. Design test data such a way that it will cover paths.	
Deployment	Network Threats : All network guards like firewall ,application firewall, honey-pot and IDS should be updated otherwise following threats are present Information gathering , Sniffing or eavesdropping , spoofing, Session hijacking, Denial of service Server Threats : Server on which your going to deploy "Web Based Conference Mgmt System" should be secure otherwise following threats are possible Viruses , Trojan horse and worms Foot printing Password cracking Denial of service Arbitrary code execution Unauthorized access	
Maintenance	All tables used in "Web Based Conference Management System" should be updated properly .	

5. Conclusion

The features of the various web-engineering methods are studied and their advantages and disadvantages are studied. Depending on the capability of the particular diagrams and notations to capture the requirements of the web engineering applications, the various characteristics are selected for the analysis and design of the web engineering applications. At least 15 complex applications are implemented by using this model and the testing criteria is applied on it. The time required for the testing is minimized and the customer accepted the web applications without any issues. This is feasible model and can capture all the requirements of the web applications. Then We proposed the secure life cycle, deployment Model and security model accordingly the case system is implemented and now it is not possible to do the attacks on the application, however we can not guess the future, the attackers can break it , but we provided the prevention and control on it.

References

[1] Daniel Schwabe, Gustavo Rossi, "An Object Oriented Approach To Web Based Application Design Theory and Practice Of Object Systems", 4(4), Wiley And Sons, New York, Issn 1074-3224, 1998.

[2] L. Baresi, F. Garzotto, and P. Paolini. Extending, "UML for modeling web applications", in Proceedings of 34th Annual Hawaii International Conference on System Sciences (HICSS'34), Maui, Hawaii, January 2001. IEEE Press.

[3] Gerti Kappel, Birgit Proll, Siegfried ReichWerner Retschitzegger, Web Engineering, The discipline of Systematic Development of Web Applications by, John Wiley & Sons, Ltd, 2006.

[4] Peter Plessers, Sven Casteleyn, Olga De Troyer, "Semantic Web Development with WSDM", CiteSeer ,scientific literature digital library.1998

[5] Stefano Ceri, Piero Fraternali, Aldo Bongio, "Web Modeling Language (WebML): a modeling language for designing Web sites", Dipartimento di Elettronica e informazione, Politecnico di Milano Piazza L. da Vinci, 32 - 20133 Milano, Italy, 2000

[6] M. José Escalona, Nora Koch, "Requirements Engineering

For Web Applications – A Comparative Study", Journal of Web Engineering, Vol.2, No.3,pp.193-212, 2004.

[7] Nathalie Moreno, Piero Fraternalli, Antonio Vallecillo, "A UML 2.0 Profile for WebML Modeling", ICWE'06 Workshops, July 1014, 2006,

[8] Richard Vdovjak, Flavius Frasincar, Geert-Jan Houben, And Peter Barna, Engineering Semantic Web Information Systems in Hera, Journal of Web Engineering, Vol. 0, No. 0, 2003.

[9]T. Isakowitz, E. A. Stohr, and P. ala subramanian, "RMM: A Methodology for Structured Hypermedia Design", Communications,1995

[10] Geert-Jan Houben "HERA: Automatically Generating Hypermedia Front-Ends for Ad Hoc Data from Heterogeneous and Legacy Information Systems", In Engineering Federated Information Systems, pages 81 – 88. Aka and IOS Press.of the ACM, Vol. 38, No. 8, August 1995, pp. 34--44.

[11] M. Gaedke, G. Graef (2000): Development and Evolution of Web-Applications using the Web Composition Process Model. International Workshop on Web Engineering at the 9th International Worldwide Web Conference (WWW9), Amsterdam, The Netherlands, May 15, 2000.

[12] Sabina Jeschke and Helmut Vieritz(2007), Accessibility and Model-Based Web Application Development for eLearning Environments, Berlin University of Technology Department of Mathematics and Natural Sciences Straße des 17. Juni 136D-10623 Berlin, Germany Wieland Schwinger, *Nora Koch(2006*), "Modeling Web Applications", Gerti Kappel c03.tex V2 -, Pg. 39

[13] Schwabe, D.; Rossi, G.; Esmeraldo, L.; Lyardet F(2001), "Engineering Web Applications for Reuse", IEEE Multimedia 8 (1), IEEE Computer Society, , ISSN: 1070-986X. pp 20-31

[14] Gustavo Rossi, Daniel Schwabe, Model Based Web Application Development, In Mendes, E. and Mosley, N. (eds.) Web Engineering, Springer2006

[15] M. Gaedke, G. Graef (2000): Development and Evolution of Web-Applications using the Web Composition Process Model. International Workshop on Web Engineering at the 9th International Worldwide Web Conference (WWW9), Amsterdam, The Netherlands, May 15, 2000.

[16] Joan Fons, Vicente Pelechano, Manoli Albert, and ´ Oscar PastorJoan Fons, Vicente Pelechano, Manoli Albert, and ´ Oscar Pastor, Development of Web Applications from Web Enhanced Conceptual Schemas, I. -Y. Song et al. (Eds.): ER 2003, LNCS 2813, pp. 232–245, 2003.

[17] Olga De Troyer, Sven Casteleyn Vrije Universiteit Brussel WISE Research group, Modeling Complex Processes for Web Applications using WSDM

[18] De Troyer, O. and Casteleyn, S., 2001b. The Conference Review System with WSDM. IWWOST 2001, Valencia, Spain(http://www.dsic.upv.es/~west2001/iwwost01/)

[19]Sabina Jeschke and Helmut Vieritz, Accessibility and Model-Based Web Application Development for eLearning Environments, Berlin University of Technology Department of Mathematics and Natural Sciences Straße des 17. Juni 136D-10623 Berlin, Germany, 2007.

[20]Wieland Schwinger, Nora Koch, "Modeling Web Applications", Gerti Kappel c03.tex V2 -, Pg. 39, 2006.

[21] Politecnico de Milano(2001), "Web Models S.r,l", WebML User Guide Version 3.0.

[22] Antonio Vallecillo1, Nora Koch2, Cristina Cachero4, Sara Comai3, Piero Fraternali3, Irene Garrigós4, Jaime Gómez4, Gerti Kappel5, Alexander Knapp2, Maristella Matera3, Santiago Meliá4, Nathalie Moreno1, Birgit Pröll6, Thomas eiter6, Werner Retschitzegger 6, José E. Rivera1, Andrea Schauerhuber5, Wieland Schwinger6, Manuel Wimmer5, Gefei hang(2007), MDWEnet: "A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods" , Workshops of 7th Intl. Conf. on Web Engineering. M. Brambilla, E. Mendes (Eds.)

[23] J. Fons1, F.J. García, V. Pelechano, and O. Pastor, User Profiling Capabilities in OOWS, J.M. Cueva Lovelle et al. (Eds.): ICWE 2003, LNCS 2722, pp. 486–496, 2003.

[24]Oscar Pastor, Joan Fons, Vicente Pelechano, Silvia Abrahao, Conceptual Modeling of Web Applications: The OOWS Approach, In Mendes, E. and Mosley, N. (eds.) Web Engineering, Springer, 2006

[25] S. Muruguesan and Y. Desphande. Web Engineering. Software Engineering and Web Application Development. Springer LNCS- Hot Topics, 2001

[26] O. Pastor, J. Gomez, E. Insfran, and V. Pelechano. The OO-Method Approach for Information Systems Modeling: From Object-Oriented Conceptual Modeling to Automated Programming. Information Systems, 26(7): 507–534, 2001.

[27] Richard Vdovjak, Flavius Frasincar, GeertJan Houben, and Peter Barna, Engineering Semantic Web Information Systems in Hera, *WWW2003*, May 20–24, 2003., Hungary. ACM xxx

[28]Joseph D. Schulz, Product Line Solutions Manager, Requirements-based Unified Modeling Language (UML),A Borland White Paper, September, 2003

About the authors

Mr. D.R. Ingle (ISTE LM'2004) is Associate Professor of Computer Engineering Department at Bharati Vidyapeeth College of Engineering, Navi Mumbai, Maharashtra state, India received bachelor degree, and Master degree in computer engineering. He has participated in more than 10 refresher courses to meet the needs of current technology. He has contributed more than 20 research papers at national, International Journals. He is life member of Indian Society of Technical Education. His area of interest are in Databases, intelligent Systems, and Web Engineering.

Dr. B.B.Meshram (CSI LM'95, IE '95) is Professor and head of Computer Technology Department at VJTI, Matunga, Mumbai, Maharashtra state, India. He received bachelor degree, Master degree and doctoral degree in computer engineering. He has participated in more than 30 refresher courses to meet the needs of current technology. He has chair more than 15 AICTE STTP Programs and conferences. He has received the appreciation for lecture at Manchester and Cardip University, UK. He has contributed more than 200 research papers at national, International Journals. He is life member of computer society of India and Institute of Engineers. His current research interests are in Databases, data warehousing, data mining, intelligent Systems, Web Engineering and network security.