



Journal of  
**Software  
Engineering**

ISSN 1819-4311



Academic  
Journals Inc.

[www.academicjournals.com](http://www.academicjournals.com)

## Uml-based Web Engineering Framework for Modeling Web Application

<sup>1</sup>Hamdy K. Elminir, <sup>2</sup>Mohamed Abu Elsoud and <sup>3</sup>A.M. El-Halawany

<sup>1</sup>Misr Academy for Engineering and Technology, <sup>2</sup>Department of Computer Science, Faculty of Computers and Information Sciences, Mansoura University, Egypt

<sup>3</sup>Digital Chains Company Mansoura Branch, Egypt

*Corresponding Author: Hamdy K. Elminir, Misr Academy for Engineering and Technology, Egypt*

### ABSTRACT

Unified modeling language-based web engineering (UWE) is a software engineering approach for the Web domain aiming to cover the whole life-cycle of Web application development. UWE uses both UML and UML extension mechanisms to develop a web application, UML extension mechanisms define specific stereotype to model system requirements model, conceptual model, navigation model and presentation model. Each one of these models has its modeling elements. Transformation rules that are used for mapping between different models are rarely investigated. Besides the absence of modeling elements to model server side process, client side process, web service which is a method of communication between different web applications and threading concept which is the smallest unit of processing that can be scheduled. A UWE framework is proposed transformation rules for mapping between different models and provides additional modeling elements. UWE based framework consists of four phases (requirements modeling phase, conceptual modeling phase, navigation modeling phase and presentation modeling phase) with each of which having its model and modeling elements. Proposed framework includes a set of transformation rules for the mapping process between different phases.

**Key words:** Engineering, modeling, web application, framework, transformation rule, client side process

### INTRODUCTION

Software engineering support the process of developing web applications based on standard steps. Developing web applications needs team members with different culture, background knowledge and tasks starting from system analyst, passing by designers and ending up at coders (Baumeister *et al.*, 1999). In this context, UWE can provide models that support different team members to do their own tasks (Rossi *et al.*, 2007).

The first step towards developing a web application is the identification of system requirements which corresponds to requirements model in UWE as mentioned by Escalona and Koch (2007). In this phase, use-case diagrams and activity diagrams are used to provide a detailed description about system (Booch *et al.*, 2005). Afterwards, design phase which includes content, navigation and presentation models. Content model is applied by using standard UML class diagram (Booch *et al.*, 2005).

Navigational model consists of navigation space model and navigation structure model using UML extension mechanisms, both models are designed based on both requirements analysis and content model.

Finally an abstract view of user interface is specified in presentation model using UML extension mechanisms (Koch *et al.*, 2000).

Modeling web applications interacts with web services doesn't investigate UWE models.

Web application implemented using server side programming language such as asp.net or php and client side languages like JavaScript, there is no way to clear the difference between server and client side processes (Baumeister *et al.*, 1999).

Besides, in navigation space model, there's always the ability to determine which object can be visited with no capability to determine who can visit this object as mentioned by Hennicker and Koch (2000). Furthermore, typical system lacks transformation rules that provide mapping between models.

Additionally, a set of transformation rules is provided for mapping between navigational space and navigational structure model which in turn determines how objects can be visited as mentioned by Koch and Kraus (2002). Finally, a new Meta-model is proposed to support transformation rules. Presentation model consists mainly of static and the dynamic presentation model. Additionally, transformation rules for mapping from navigational model to presentation model are used.

This study, proposes a UWE-based framework to support the process of modelling web applications by adding new modelling elements to activity diagrams that aim to clear the boundary between server side and client side activities in requirements phase. Moving to design phase, proposed framework uses a rendering model to be responsible for the final layout of web content as well as supporting content reusability. Afterwards, in navigational model, navigational space model redefined by adding elements to model actor-object accessibility.

## **FRAME WORK**

Proposed framework based on UWE for supports web applications development starting from system requirement analysis phase moving on to design phase as shown in Fig. 1. UWE Requirement analysis phase contains two steps use cases diagrams and activity diagrams (Koch *et al.*, 2006). Proposed framework cover the limitation of threading concept modeling by adding new type of use case to model tasks that are performed by system itself.

Activity diagram add new three types of activities they are server side, client side and outer system to reveals the relevancy between server and client side activities additionally, uses outer system activity to cover the limitations of modeling web services. UWE design phase contains three models conceptual, navigational and presentation model. Typical UWE conceptual model includes content modeling using standard UML class diagrams (Rossi *et al.*, 2007). In order to improve conceptual modeling, an earlier work provided the idea of separating content and presentation adding a new rendering model. However, this model was proposed without any elements that are used to model it (Patricia *et al.*, 2000). Whereas, proposed framework divides conceptual model into two models content model and rendering model each of this is modeled using standard UML class diagram. Besides, a set of transformation rules that's responsible for mapping between content model and rendering model.

Navigation Space Model in proposed framework describes which object can be visited by which user. Here a modeling element actor is used to determine which actor of our web application can visit this object. Also, transformation rules are provided to transform from conceptual model to navigation space model as well as transformation from navigation space model to navigation structure.

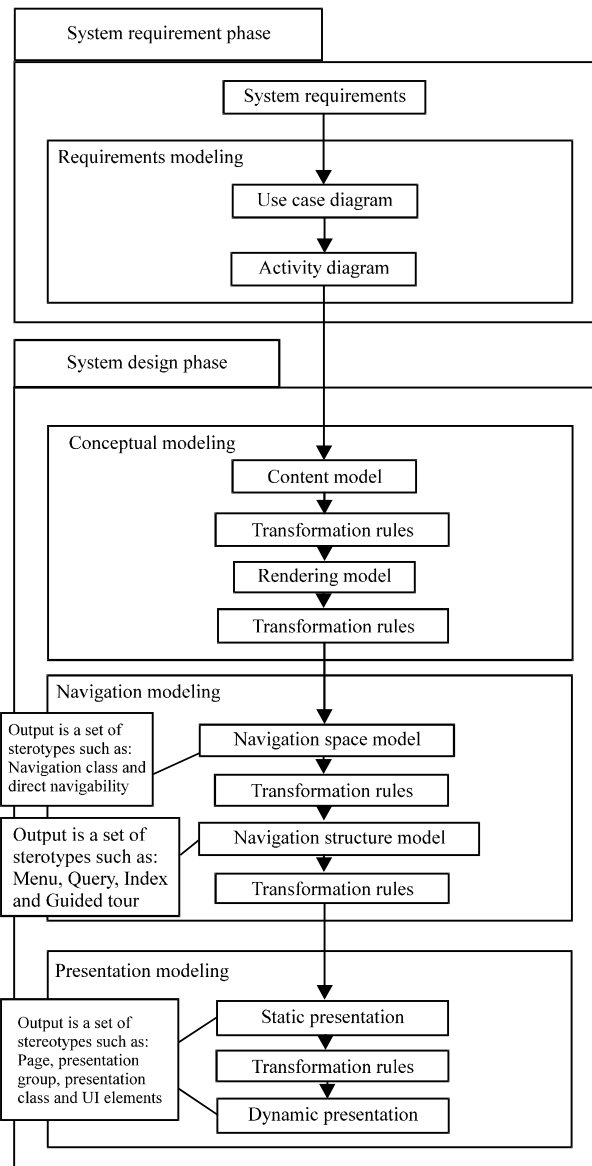


Fig. 1: The proposed framework

Presentation model defines where and how navigation objects will be presented to user by distinguishes between two types of presentation static and dynamic presentation defined by Koch and Kraus (2002). Proposed framework changes the Meta-model of presentation model elements to support both the transformation from navigation to presentation as well as the transformation from static to dynamic presentation.

**System requirements phase:** It consists of two steps, the first step is to provide rough description of system functionality using use cases diagram. The second step provides a more detailed description of use cases using activity diagrams that depict the responsibilities and actions of stakeholders.

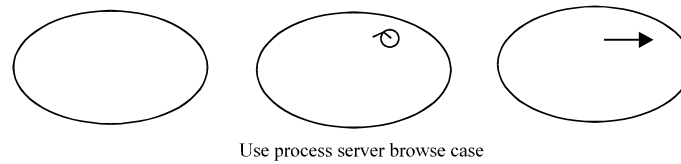


Fig. 2: Three types of use case

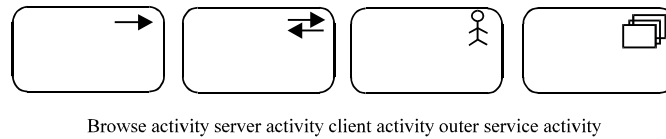


Fig. 3: Four types of activities

Proposed framework distinguishes between three types of use cases: user process, server process and browse, as shown in Fig. 2.

User process use cases are used to describe business tasks that the end user will perform on the system and it is modeled in the same way as the traditional software. Browse use cases are used to model user behavior during browsing the web application contents and modeled in the same way as UWE. Server process use cases describes business tasks that the system itself performs and it may be triggered on a specific time or invoked by a web services without any user intervention.

Proposed framework, activity diagram includes four types of activities: browse, client side, server side and outer side activity as shown in Fig. 3.

Browse activity is used to describe user navigation through web system typical as it exists in UWE (Rossi *et al.*, 2007). Client side activity describes tasks performed by user at client side. Server side activity describes tasks that take place at server side. Outer side activity describes tasks that need outer party (i.e., web service) to be completed.

**System design phase:** System design phase includes three models conceptual, navigation and presentation model as stated by Koch and Kraus (2002). Conceptual modeling aims to model system content using standard UML class diagram (Rossi *et al.*, 2007). After modeling web content and drawing system class diagrams, transformation rules are applied on class diagrams to obtain rendering model (Rossi *et al.*, 2007). Thus, obtaining a new class diagram which represents rendering model that's responsible for displaying system objects. Navigation modeling includes both navigation space model and navigation structure model. Navigation space model is obtained by applying transformation rules on conceptual model. Afterwards, transformation rules are in turn applied to navigation space model to obtain navigation structure model. Presentation model involves both static and dynamic presentation (Koch *et al.*, 2000). Static presentation is obtained by applying transformation rules on both navigation space and navigation structure models. Whereas, dynamic presentation is obtained by applying transformation rules on static presentation.

**Conceptual model:** Conceptual model contains two models; content model and rendering model which are both represented using standard UML class diagram (Rossi *et al.*, 2007). Rendering model was introduced by Rossi *et al.* (2007) to separate between actual content and presentation or user view of content. This separation supports the concept of content reusability. Proposed

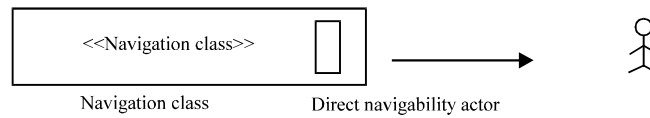


Fig. 4: Navigation space modeling elements

framework provides transformation rules to transform from content model to rendering model and here are our transformation rules:

- All Classes in content model are mapped to rendering model
- For each class in the Rendering Model, find its corresponding use case in the use case diagram and make it as method in that class, where the name of the method in rendering class is the name of the corresponding use case
- For each class in the Rendering Model, find its corresponding browse activity in the activity diagram and make it as method in that class, where the name of the method in the rendering class is the name of browse activity
- Remove redundant methods that have the same functionality in the system
- Remove classes that don't have any methods
- Association between classes is determined by the activity diagram scenario

**Navigation model:** Proposed framework let Navigation space model determines which object can be visited by which actor. Here the use case modeling element actor is reused in navigation space model to determine actors that were allowed to access these objects in order to provide more security issues and to determine the roles of each actor in the system. The modeling elements of the navigation space model as shown in Fig. 4 are the navigational class, actor and direct navigability (Koch *et al.*, 2000).

Navigational class represents a conceptual class whose instances are visited by that user during navigation.

Actor represents the user that can visit that instance of class during navigation.

Direct navigability represents association from source navigation class to target navigation class.

Proposed framework provides the transformation rules to transform from the conceptual Model to the Navigation Space Model and these rules are:

- All classes in the Rendering Model are mapped to the navigational class in the Navigation Space Model
- All classes that exist in the Content Model and does not exist in the Rendering Model are irrelevant in the navigation process
- For each class determine who can visit this class through reviewing the corresponding use case related to that class
- All associations between classes mapped from the Rendering Model and avoiding the Navigation path of greater than one

Navigation Structure Model determines how objects are reached and is built on the basis of the Navigation Space model by applying the transformation rules and here are the rules:

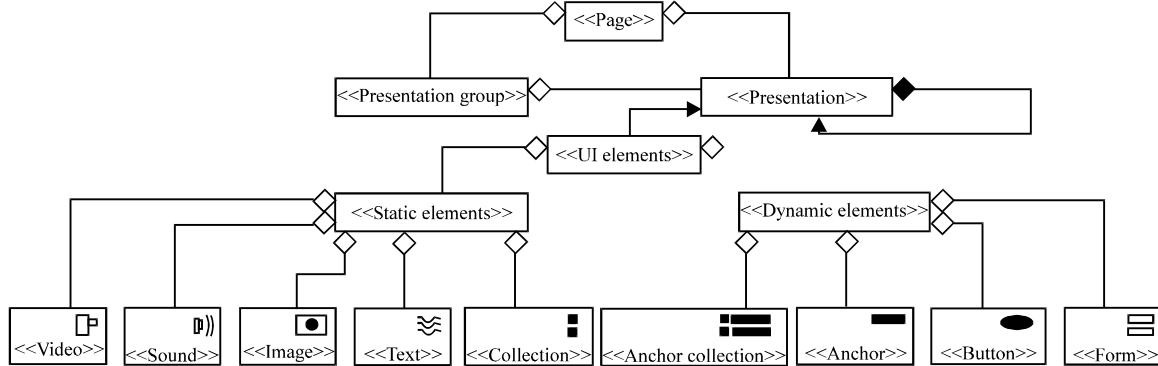


Fig. 5: Meta -model for the modelling elements of presentation model

- Navigational classes are mapped from the Navigation Space to the Navigation Structure
- Classes that have a composition or aggregation relation with other classes are added to the menu with the related class
- Menu item must associate with one of (guided tour , index)
- For each item in the menu corresponding to the navigational class in the Navigation Space Model, check its relation if it is one-to-one, then the association is a Guided Tour and if it is multiplicity, then it is associated to index
- Guided tour and index are associated to the navigational class

**Presentation model:** Presentation model consists of two steps: the static presentation and the dynamic presentation. Presentation modeling is useful for communication between web site designer and customer in the early phase. Also it is independent of actual techniques used to implement the web site, thus allow stakeholders to discuss appropriateness of the presentation before actually implementation (Rossi *et al.*, 2007). UWE defines Static presentation as an abstract user interface design models only the structured organization of presentation given by interface objects such as text, images and menus. Static presentation doesn't model the layout characteristics in terms of fonts and color such as decision that is taken during the development of the user interface prototype or during the implementation phase.

A new Meta-model is proposed for the modeling elements of presentation model as shown in Fig. 5. Proposed framework work with the typical UWE presentation model elements but yet changes the Meta-model where user interface elements are inherited from presentation class then user interface elements are then categorized as either static or dynamic elements. This categorization is helpful when transforming from static to dynamic presentation.

The modeling elements of the presentation model are: pages, presentation group, presentation class, image, text, form, button, anchor, anchor collection, sound and video (Rossi *et al.*, 2007).

Page each instance of this class is a container of all the abstract UI elements and can contain the presentation group and the presentation class.

Presentation Group can itself contain the presentation group and the presentation class.

Presentation Class allows partitioning a user interface view into groups of user interface elements.

UI elements are categorized to static elements (collection, text, image, sound and video) and dynamic elements (form, button, anchor and anchor collection).

Dynamic presentation shows us the available choice for navigation from one page to another. In dynamic presentation we use the concept of package, where each package contains the presentation classes that exist in the static presentation.

Our framework provides the transformation rules to transform from the Navigation Model to the static presentation model and here are the rules:

- Construct a presentation page for each navigation class and for each index occurring in the Navigation Structure Model
- Each navigation class and index is mapped to a presentation class and is included in its page
- Choose the navigation class as a root for navigation
- Each menu is represented as presentation class in static presentation and its elements are represented as anchors
- For each index class in the Navigation Structure, relevant navigation class is mapped as anchor collection
- For each guided tour, relevant navigation class is mapped as a presentation class
- Presentation groups come from two or more presentation classes and the presentation class comes from index or menu item

After transformation from navigation model to static presentation model we apply transformation rules on static presentation to obtain dynamic presentation and here are the transformation rules:

- For each page that has a presentation class that contains a dynamic element, map that page to a presentation package
- Each presentation class exists in the page is mapped to a presentation class that exists in the presentation package
- Remove the presentation class that don't contain dynamic element from presentation package
- From the Navigation Structure and the corresponding presentation class, we can derive an association between presentation classes in the dynamic modeling
- Each presentation class is associated with a presentation Package

### **EXAMPLE APPLICATION**

As an example to illustrate the proposed framework presented in this study, some models of the Web application serves clients of Egyptian bourse. This application presents information about a company that's registered in the Egyptian bourse, this information is company budget, contact information, company news, general public meeting dates and number of stocks.

The web application has three types of users: administrators, Visitors and clients. Administrators are responsible for managing web site by adding, updating or deleting companies' information and have the authority to log in to web site control panel as well as having the ability to monitor users and block user accounts if necessary. Visitors have the ability to register in the web site and choose company to belong to as well as being able to see information about companies that exists in the Egyptian bourse. Clients have the ability to log in to the web site and see any information about companies with search capabilities using different queries i.e. company name. Clients also have the ability to update their personal information and logout of the system. Information about companies such as name and number of stocks is stored in Egyptian bourse database where information about



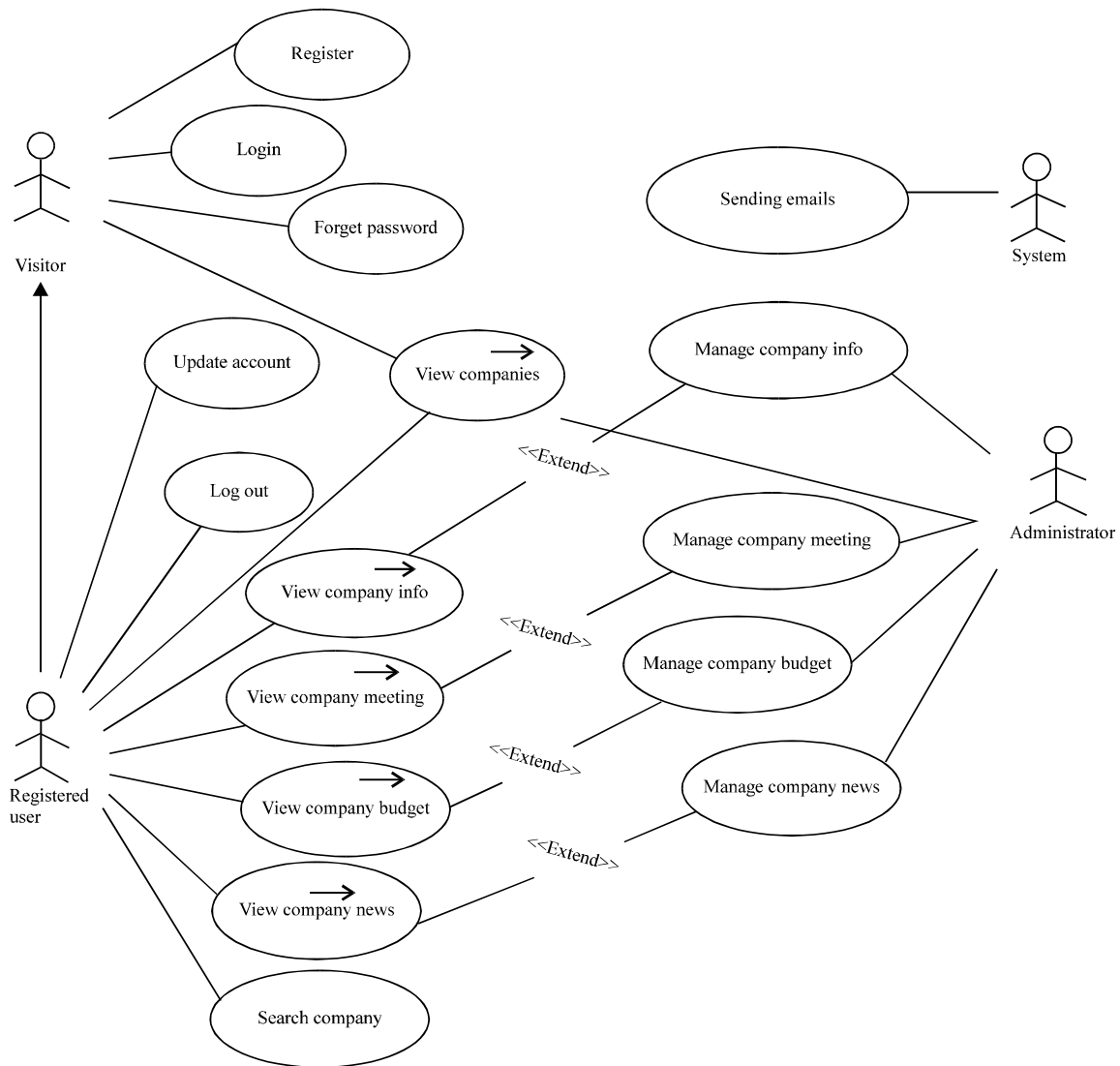


Fig. 6: UWE use case for example application

companies like news or budget are added from control panel and stored in our web site database. System includes sending emails service that automatically sends emails to company's client's schedules of General Public meetings.

**Modeling requirements phase:** Example application contains four actors they are: visitor, registered users (clients), administrator and system it's self. As shown in Fig. 6 visitor can use register, login, forget password and view companies. Registered users can use logout, update account, view companies, view company information, view Company meeting, view company news and search about company. Administrators can use manage company information, manage company meeting, manage company budget and manage company news. System can use sending email use case and it is a server process use case that uses by the system it's self.

According to proposed framework next step after finishing use case diagram is to model activity diagram. Figure 7 shows UWE Activity diagram of login use case. Here the visitor begin to enter

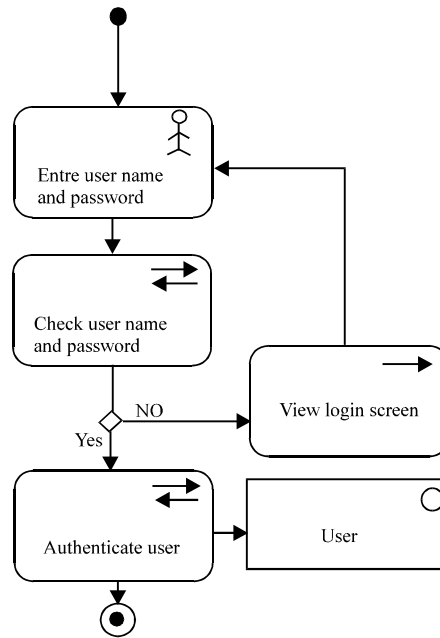


Fig. 7: UWE activity diagram detailing login use case

his name and password at client said to log in to the system after complete this activity the system verify user name and password if the user name and password was true then the system authenticate the visitor at server side and then finish login. If the user name and password was false then the system browse to the login screen and the visitor begin to enter his user name and password again.

Figure 8 shows UWE activity diagram detailing register use case.

Here the system brings companies that registered in the Egyptian bourse from outer system boundary and then visitor begin to fulfill the registration form at client side. After finishing fulfills the registration form we have three activities that happen at the same time which are: check email if exists, check mobile if exists and check user name if exists. In the check mail activity if the email exists then we go to fulfill registration form again to change email address else we go to register activity. In the check mobile activity if the mobile exists then we go to fulfill registration form again to change mobile else we go to register activity. In the check user name activity if the user name exists then we go to fulfill registration form again to change user name else we go to register activity. In the register activity we register visitor information and the visitor become user of the system.

**Modeling design phase:** Figure 9 shows UWE Class diagram for modeling content. Example application has five main objects which are: Company, meeting which is general public meeting of the company, news of the company, budget of the company and users of our web application. The five main objects of the system are obtained from system activity diagram. As shown in Fig. 9 the object user is obtained as a result of authentication activity.

After finishing content model apply transformation rules on content model to obtain rendering model which responsible for rendering view of our web system objects exists in content in our web application.

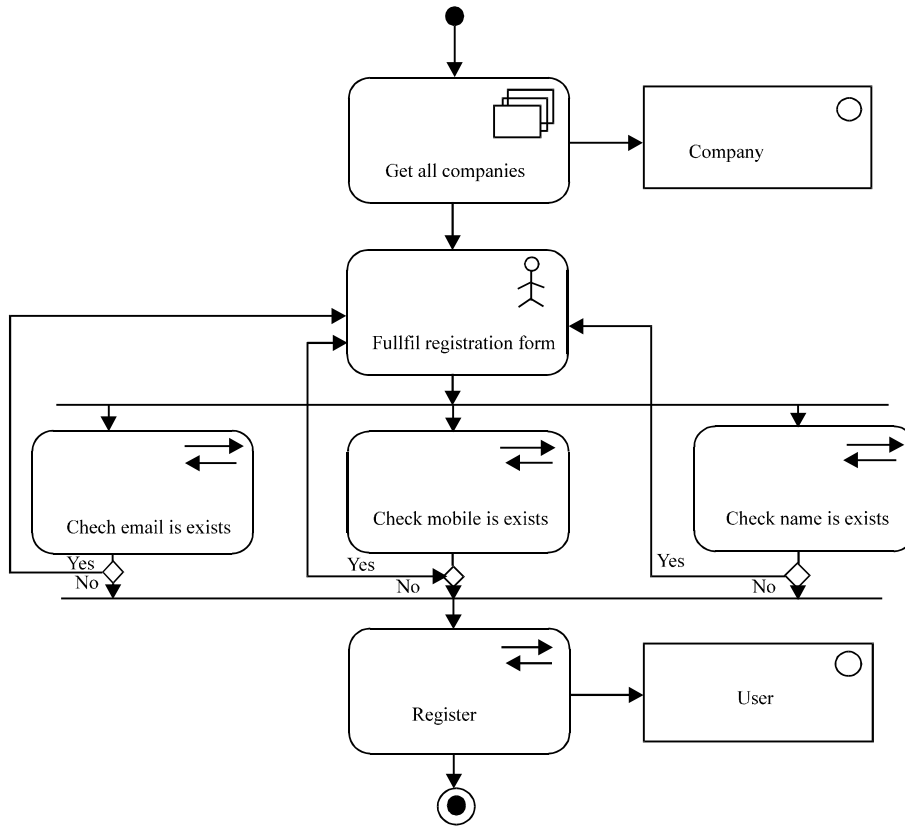


Fig. 8: UWE activity diagram detailing register use case

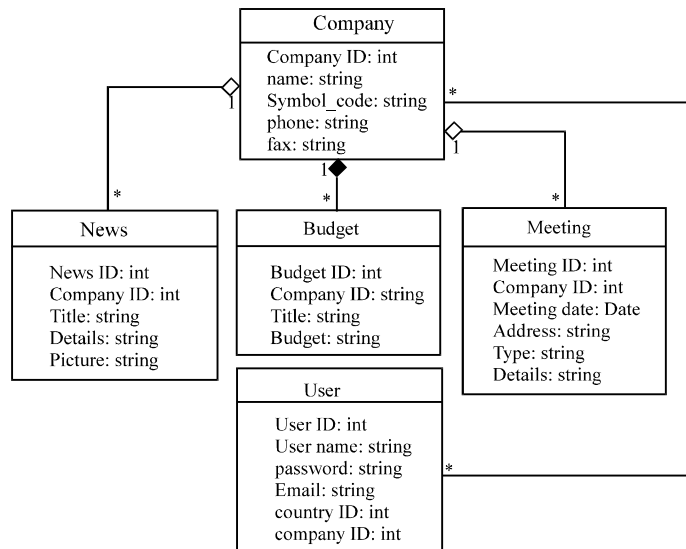


Fig. 9: UWE class diagram for modeling content

Figure 10 shows UWE class diagram for rendering model.

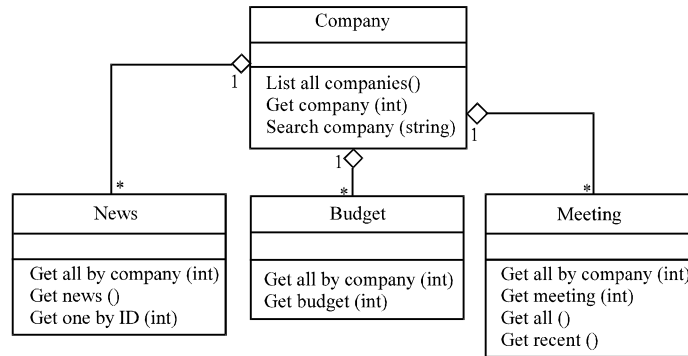


Fig. 10: UWE class diagram for rendering model

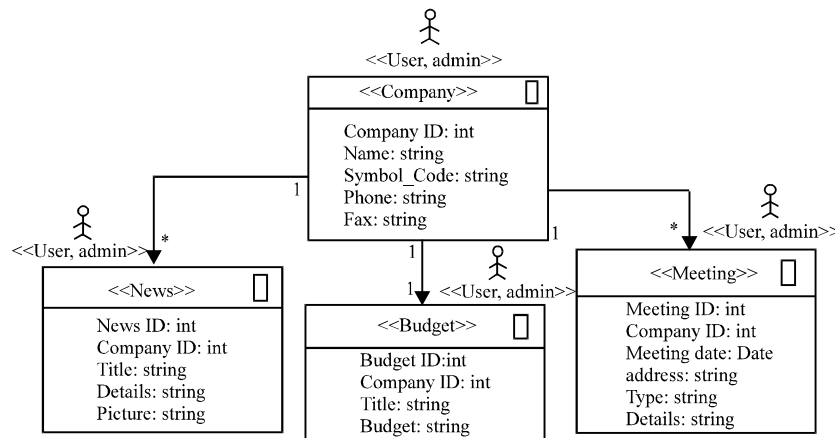


Fig. 11: UWE navigation space model

Rendering model is obtained by applying transformation rules stated before. Class diagram for rendering model shows all Classes that exist in content model. Each class of this diagram associated methods which responsible for displaying objects of that class in our web system.

To obtain navigation space model we apply transformation rules against conceptual model. Figure 11 shows UWE navigation space model which contains four navigational class and association between them using direct navigability also, the actor that determine which actor of the system can access this object.

To obtain navigation structure model we apply transformation rules against navigation space model. Figure 12 shows UWE navigation structure has four navigational class company, news, meeting and budget. Company navigation class is associated with menu which can make navigation to news navigation class through news index, meeting navigation class through meeting index and budget navigation class through company guided tour.

Static presentation model is obtained by apply transformation rules against navigation model. According to transformation rules which say Construct a presentation page for each navigation class and for each index occurring in the Navigation Structure Model so Fig. 13-16 are obtained.

Figure 13 shows static presentation for company index page which contains company index presentation class.

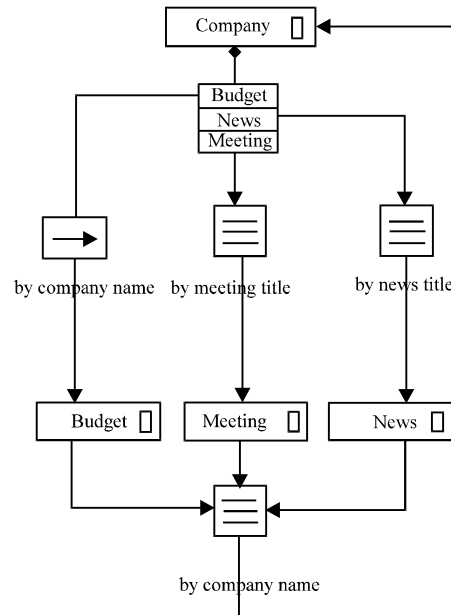


Fig. 12: UWE navigation structure model

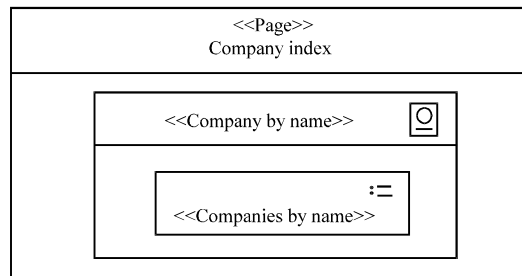


Fig. 13: Static presentation for company index page

Figure 14 shows static presentation for company page which contains company presentation class and presentation group which contains three presentation classes they are: company index, news index and meeting index navigation class.

Figure 15 shows static presentation for news index page which contains news index presentation class and navigation group which contains two presentation classes: company index and meeting index navigation class.

Figure 16 shows static presentation for news page which contains news Presentation class and presentation group which contains two presentation classes: company index and news index presentation class.

In dynamic presentation the concept of package is used where each package contains presentation class that exists in static presentation. Figure 17 is obtained by applying transformation rules against static presentation.

Figure 17 presents dynamic presentation for our web application which contains four packages they are: company index page, company page, news index page and news page.

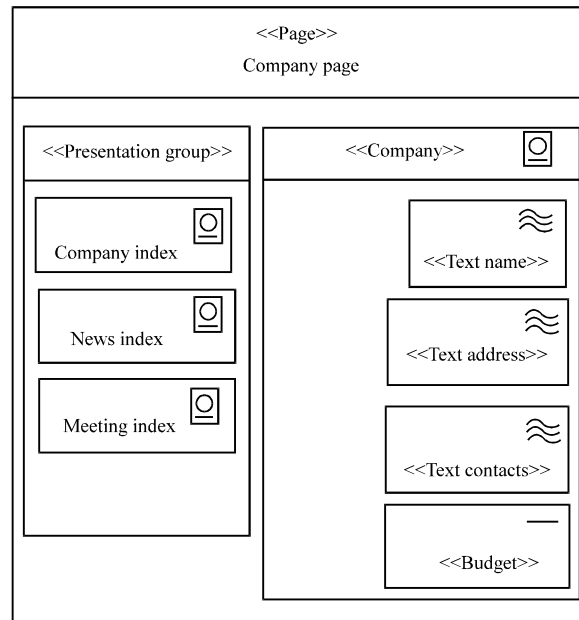


Fig. 14: Static presentation for company page

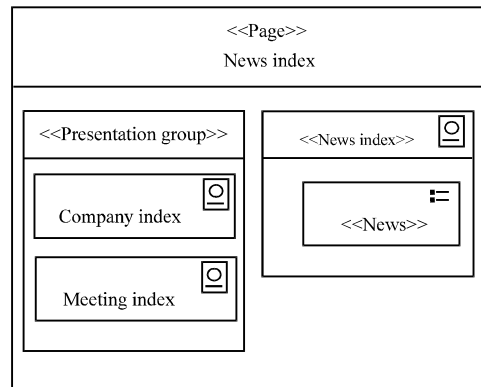


Fig. 15: Static presentation for news page index

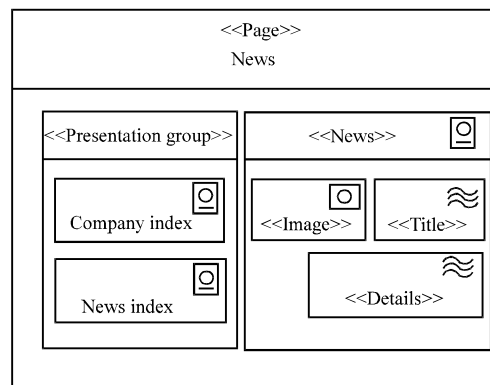


Fig. 16: Static presentation for news page

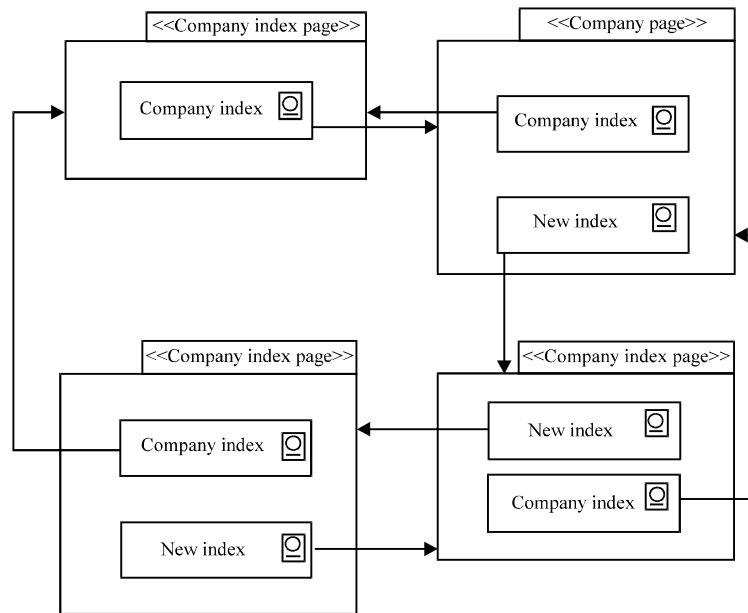


Fig. 17: Dynamic presentation for web applications

## CONCLUSION AND FUTURE WORK

This study proposes a framework to model web applications using UML-based web engineering. The framework consists mainly of two phases' system requirements phase and System design phase.

In system requirements phase, new elements are added to represent threading concept and web services in web environment. Moreover, three types of activities are added to activity diagram to show the difference between activities that take place at server side, client side and outside system boundary. In the design phase, there are three processes namely: conceptual, navigation and presentation model. The conceptual model has content model and rendering model also, provides transformation rules to map between content and rendering models. In navigation model the navigation space model is redefined then provides transformation rules to transform from conceptual model to navigation space model, as well as rules to transform from navigation space model to navigation structure model. In presentation model a new Meta-model is proposed to support transformation rules to transform from navigation model to static presentation model and from that to dynamic presentation model.

Finally an example application to illustrate the proposed framework is presented.

Future work includes designing a case tool that can automatically apply transformation rules for mapping between different stages as well as generating drawings that represent rendering, navigation space, navigation structure, static presentation and dynamic presentation models using an input of content model.

## REFERENCES

- Baumeister, H., N. Koch and L. Mandel, 1999. Towards a UML Extension for hypermedia design. Proceedings of the 2nd International Conference on The Unified Modeling Language: Beyond the Standard, October 28-30, 1999, France, pp: 614-629.

- Booch, G., R. James and J. Ivar, 2005. The Unified Modeling Language User Guide. 2nd Edn., Addison Wesley Professional, New York, ISBN: 0-321-26797-4.
- Escalona, M.J. and N. Koch, 2007. Metamodelling the requirements of web systems. *Web Inform. Syst. Technol.*, 1: 267-280.
- Hennicker, R. and N. Koch, 2000. A UML-based methodology for hypermedia design. Proceedings of the 3rd International Conference Unified Modeling Language, (UML'00), Springer Verlag, New York, pp: 410-424.
- Koch, N. and A. Kraus, 2002. The expressive power of UML-based web engineering. Proceedings of the 2nd International Workshop on Web-Oriented Software Technology, (IWWOST'02), Malaga, Spain, pp: 105-119.
- Koch, N., G. Zhang and M.J. Escalona, 2006. Model transformations from requirements to web system design. Proceedings of the 6th International Conference Web Engineering July 11-14, 2006, ACM, Palo Alto, California, USA., pp: 281-288.
- Koch, N., H. Baumeister, R. Hennicker and L. Mandel, 2000. Extending UML for modeling navigation and presentation in web applications. Proceedings of the Workshop Modeling Web Applications in the UML, (UML'00), England, pp: 1-5.
- Patricia, V., S. Daniel and S.C. de Sieckenius, 2000. A diagrammatic tool for representing user interaction in UML. Proceedings of the 3rd International Conference Unified Modeling Language. (UML'00), Springer, Berlin, pp: 133-147.
- Rossi, G., O. Pastor, D. Schwabe and L. Olsina, 2007. *Web Engineering: Modelling and Implementing Web Applications*. 1st Edn., Springer, New York, ISBN: 184628922X, Pages: 464.