Enriching Web Requirements Analysis by Adding Style Factor into UML

Lecturer Dr.Zainab M. Hussain* Mohammed QasimShawkat**

Abstract

As requirement engineering is considered to be the most important phase in software development process because of the high cost impact of repairing requirement phase errors, therefore web requirement engineering is known to be one of the most popular research areas today. By observing most of the conducted researches it can be concluded that they focus on the application of user experience from content, navigational and presentational point of view.

UML (Unified Modeling Language) is a general purpose visual modeling language for specifying, constructing and documenting systems, which can be used in all major application domains and implementation platforms.
In this paper a new dimension in the UML user experience modeling will be presented which represents the style factor or by other words the web theme. Every company, business or corporation has its own styling factors which are essential for the success of the institute's mission, and therefore should be referred to as 'requirements'. This new dimension will be translated into a new web style diagram with all the associated relationships and constrains. The web style diagram will enable the customer (institute requesting the web site or application) to ensure that the resulting web site or application will meet their style identity.

Keywords: *Web Engineering, Requirement Engineering, UML, User Experience.*

*Al-Mansour University College
** Zain Mobile Communication Company

1. Introduction:

Requirements models should be the result of an intensive communication with the customerand provide the representation of the business decisions related to the applicationto be developed. The more accurate the models produced in this early phase of the development, the less error-prone are the models and the code generated in the following steps [1].

Web Engineering is gaining increasing interest recently due to the enormous number of web sites applied, in addition to the general trend towards web applications instead of traditional computer applications. Requirements play a key role in the development of Web applications. Typical consequences of poor requirements are low user acceptance, planning failures, or inadequate software architectures. Requirements Engineering (RE) deals with the principles, methods, and tools to identify, describe, validate, and manage requirements in system development. Today, numerous RE methods and tools are available. However, these approaches are often not applied by practitioners and RE is often performed in an ad-hoc manner, particularly in Web engineering. Although the complexity of today's Web applications require a more systematic approach, the maturity of the RE process is often insufficient [2].

A model-based approach development of Web applications can be performed using UML techniques and UML notation. Web analysis and design usually takes into account a clear separation of contents, navigation and presentation, i.e. it is performed in the following steps: conceptual, navigational and presentational design. Part of the activities of the conceptual, navigational and presentational design is the constructions of models and their graphical representation. These models consist of UML standard model elements or specific model elements – stereotypes – defined according to the UML extension mechanisms [3].

In this paper a new dimension in the UML user experience modeling will be presented which represents the style factor or by other words the web theme. This new dimension will be translated into a new web style diagram with all the associated relationships and constrains. The rest of this paper is organized as: section 2 describe the UML based web engineering, section 3 describe the User Experience (UX) Modeling, and section 4 describe the UML Extension Mechanisms, while the proposed Web Style Diagram is presented in section 5, with its properties in section 6 and its implementation using two company's web site examples in section 7. Finally section 8 illustrates the conclusions.

2. UML Based Web Engineering

The development of Web systems is subject to continuous changes in user and technology requirements. Models built so far in any stage of the development process have to be easily adaptable to these changes. To cope efficiently with the required flexibility, UML based Web Engineering (**UWE**) advocates a strict separation of concerns in the early phases of the development and implements a model- driven development process, i.e., a process based on the construction of models and model transformations. The ultimate challenge is to support a development process that allows fully automated generation of Web systems.Similarly to other Web Engineering methods, the UWE process is driven by the separate modeling of

concerns describing a Web system. Models are built at the different stages of requirements engineering, analysis, design, and implementation of the development process and are used to represent different views of the same Web application corresponding to the different concerns (content, navigation structure, and presentation) [5].

For each view, an appropriate type of UML diagram is selected and a set of stereotypes, tag definitions and constraints is provided [5]. The *content* model is used to specify the concepts that are relevant to the application domain and the relationships between these concepts. The *hypertext* or *navigation* structure is modeled separately from the content, although it is derived from the content model. The *navigation model* represents the navigation paths of the Web system being modeled. The *presentation model* takes into account presentation and user–machine communication tasks.UWE proposes at least one type of UML diagram for the visualization of each model to represent the structural aspects of the different views. Figure (1) shows how the scope of modeling spans these three orthogonal dimensions: development stages, systems' views, and aspects [4].

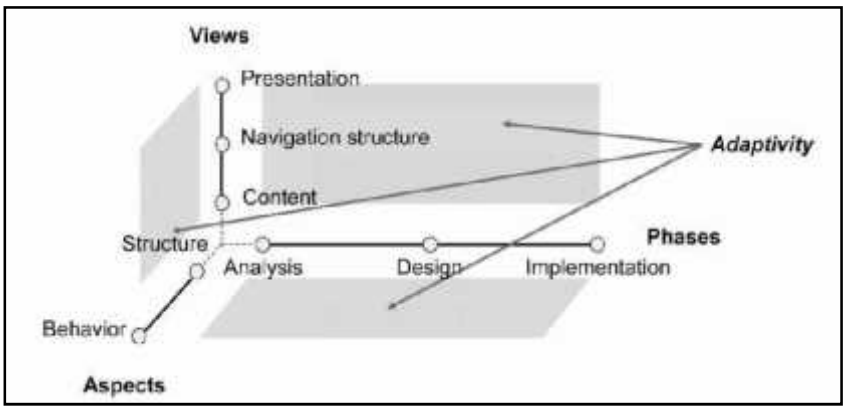


Figure (1) Modeling aspects in UWE

UWE comprises a notation, a method and tool support. The notation is defined as a UML profile, i.e. using the extension mechanism provided by the UML itselfwhich allows for the refinement of UML in a strictly additive manner by stereotypes,tag definitions and constraints, providing the required additional annotations [6].

3. User Experience (UX) Modeling

Requirement models and architecture models focusing on specific Web aspects complete the specification of the Web system. Separation of concerns offers advantages in the maintenance and re- engineering of a Web system as well as for the generation of Web systems for different contexts and platforms. UWE emphasizes the relevance of requirements engineering starting with modeling activities in this early development phase. Therefore, the UWE meta-model includes a set of modeling primitives that allows for simpler and more specific specification of the requirements of Web systems [4].

The user experience (UX) Model describes the screens of the system, the dynamic content that appears on the screens and how the user navigates through the screens to execute the system functionality (i.e., the use cases) [7].

Since use cases do not specify user interface details, something else is needed to do that. This is where the UX model and storyboards are very effective. The UX model is conceptual; it specifies user experience elements, screens, and screen content in an abstract representation. It helps architects and designers determine what will go into the UI (User Interface) before worrying about the details of specific UI elements (buttons, pull-down menus, etc.) The UX model encourages good interface architecture. In fact, it serves as a sort of contract between the UI team and the development team, ensuring that developers don't just "make it up" as they go.UX modeling is not required on every project. Creating a model takes effort that should not be expended without weighing the potential benefit. Typically, the benefit of a UX model outweighs the cost and effort required if the user experience is critical to the system's success, if the system is very new to the users, or if UI errors might have costly consequences [8].

4. Integrating Web Theme into UML

The UML is a general purpose, tool supported, and standardized modeling language that is used in order to specify, visualize, construct and document all the elements of a wide range of system intensive processes. It promotes a use case driven, architecture centric, iterative and incremental process, which is object oriented and component based. The UML is broadly applicable to different types of systems, domains, methods and processes, which is why it is such a popular and broadly used language [9].

However, even though the UML is very well-defined, there might be situations in which it might find the wanting to bend or extend the language in some controlled way to tailor it to the specific problem domain in order to simplify the communication of the objective. This is where the UML extension mechanisms come in [9].

4.1 UML Extension Mechanisms

The extensibility mechanisms allows to customize and extend the UML by addingnew building blocks, creating new properties, and specifying new semantics in order to make the language suitable for the specific problem domain. There are three common extensibility mechanisms that are defined by the UML: stereotypes, tagged values, and constraints [9]:

- **Stereotypes**

Stereotypes allows to extend the vocabulary of the UML so that it can createnew model elements, derived from existing ones, but that have specific properties that are suitable for the problem domain. They are used for classifying or marking the UML building blocks in order to introduce newbuilding blocks that speak the language of the domain and that look like primitive, or basic, model elements.For example, when modeling a network it might need to have symbols forrepresenting

routers and hubs. By using stereotyped nodes it can be make these things appear as primitive building blocks.As another example, let us consider exception classes in

Java or C++, which itmight sometimes have to model. Ideally you would only want to allow them to be thrown and caught, nothing else. Now, by marking them with a suitable stereotype you can make these classes into first class citizens in your model; in other words, you make them appear as basic building blocks [9].

Stereotypes also allowintroducing new graphical symbols for providing visualcues to the models that speak the vocabulary of the specific domain. Graphically, a stereotype is rendered as a name enclosed by guillemots and placedabove the name of another element as shown in figure (2.A). Alternatively, it can render the stereotyped element by using a new icon associated with that stereotype as shown in figure (2.B) [9].

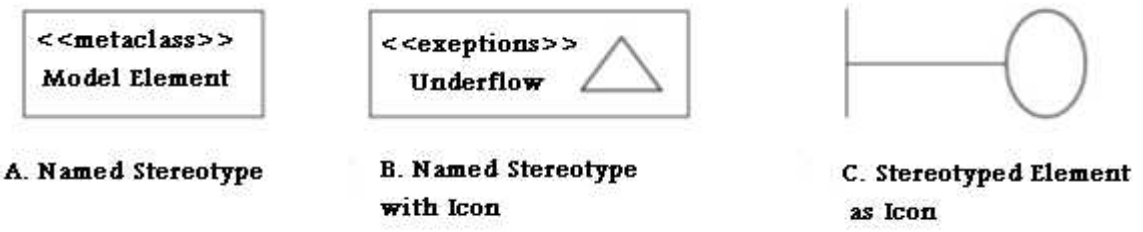


Figure (2) Stereotypes

- **Tagged Values**

Tagged values are properties for specifying keyword-value pairs of model elements, where the keywords are attributes. They allow extending the properties of a UML building block so that it creates new information in the specification of that element. Tagged values can be defined for existing model elements, or for individual stereotypes, so that everything with that stereotype has that tagged value. It is important to mention that a tagged value is not equal to a class attribute. Instead, it can regard a tagged value as being a metadata, since its value applies to the element itself and not to its instances.

One of the most common uses of a tagged value is to specify properties that are relevant to code generation or configuration management. So, for example, you can make use of a tagged value in order to specify the programming language to which you map a particular class, or you can use it to denote the author and the version of a component. As another example of where tagged values can be useful, consider the release team of a project, which is responsible for assembling, testing, and deploying releases. In such a case it might be feasible to keep track of the version number and test results for each main subsystem, and so one way of adding this information to the models is to use tagged values [9].

- **Constraints**

Constraints are properties for specifying semantics and/or conditions that must beheld true at all times for the elements of a model. They allow extending the semantics of a UML building block by adding new rules, or modifying existing ones. For example, when modeling hard real time systems it could be useful to adorn the

models with some additional information, such as time budgets and deadlines. By making use of constraints these timing requirements can easily be captured [9].

Graphically, a constraint is rendered as a string enclosed by brackets, which is placed near the associated element(s), or connected to the element(s) by dependency relationships. This notation can also be used to adorn a model element's basic notation, in order to visualize parts of an element's specification that have no graphical cue. For example, you can use constraint notation to provide some properties of associations, such as order and changeability as shown in figure (3) [9].

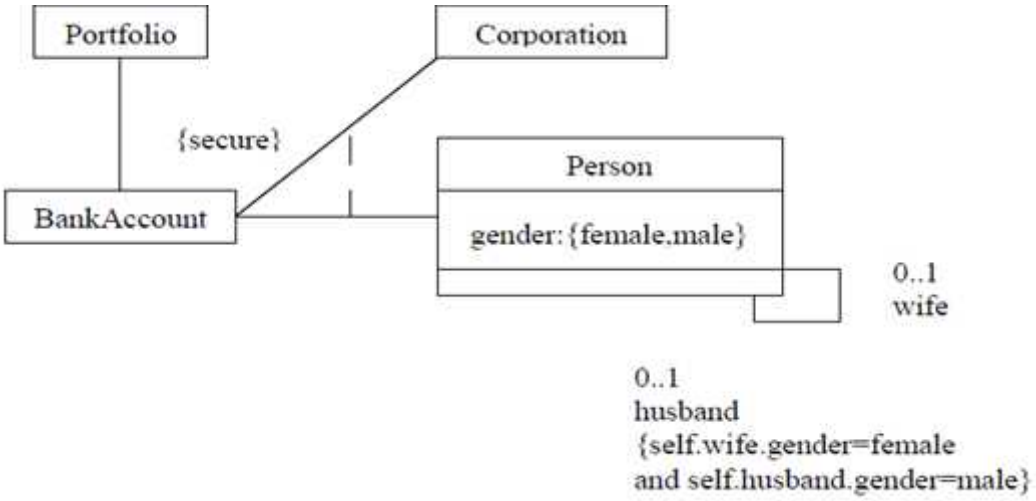


Figure (3) Formal Constraint

5. The Proposed Web Style Diagram

Every company, business or corporation has its own styling factors which are essential for the success of the institute's mission, and therefore should be referred to as 'requirements' The Web Style Diagram (**WSD**) is proposed to cover the theme and style aspects which are essential to the institute's website or web application requirements. This view is not covered in the user experience modeling and is, never the less, an important side of web requirement analysis. The proposed WSD consists of two levels of structure: the *top level style sheet* and the *web screen*.

(1) The Style Sheet

The ***style sheet*** is the structuring element that represents the styling aspects that are common over several web pages or screens in a web site or web application. As an example, most web sites and applications have the same company's or institute's logo placed at the same location in all the web site or web applications pages. Also a common page background, font or animation can be found in many web pages of the web site. All these common style aspects can be grouped into style sheets in the WSD .The proposed style sheet can be represented as:

Stereotype<<Style Sheet>> **ObjectIcon**

Object Decoration

The *style sheet* has a distinct object icon of the *paint pallet* which reflects the style view and specialization of the style sheet. The *object decoration* has three sections the first is the *style sheet name* which denotes the style sheet distinction from other style sheets in the WSD. For example a style sheet named "Background Style" denotes that this style sheet includes the web pages background aspects. The second is the *attributes section* which contains the styling attributes (properties) common to different pages. For example the attribute "background: url('img_logo.png) top left" indicates that this common background image is the company's logo located at the top left part of the page; the image type is png which can be reached through a specific URL. The third section is for the *operations* which include the style operations common to a number of web pages. For example the operation "animation: orange _ world _wide 5s infinite ()" indicates that this common animation is the orange worldwide animation which is 5 seconds in duration and is repeatable (infinite). Figure (4) shows an example of a style sheet for setting the background features of several web pages.

The style sheet contains the style attributes and operations which are cascaded onto several web pages or 'web screens' as referred to in the user experience modeling. It includes the common attributes and operations between several screens.

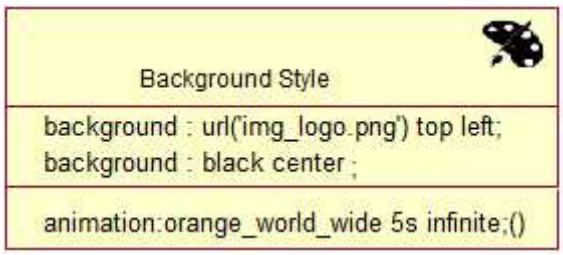
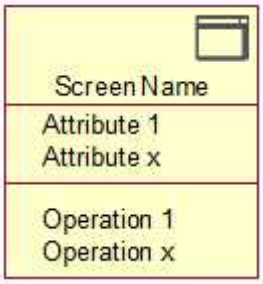


Figure (4) Style Sheet Example

(2) The Web Screen

The ***web screen*** is the structuring element that represents the styling aspects that are specific to a certain web page in the web site or application and are not common to any other page. These specific style aspects can be modeled with specific web screens. The proposed web screen consists of the following stereotype and object icon:

Stereotype<<Web Screen>> **Object Icon** 



Object Decoration

The *web screen* has the distinct object icon shown above and also like the style sheet its object decoration has three sections the first is the *web screen name* which denotes the web screen distinction from other web screens in the WSD. For example a web screen named “Home Page” denotes that this web screen will include the specific style aspects for the home page only, the second section is the *attributes section* which includes the style properties specific for the home page. In an example the attribute “background: yellow center; ” indicates that the background color of the home page is yellow and its position is in the center. The third section is the *operations section* which includes the style operations specific to the home page only. In example the operation "animation: lays header 5s infinite;()” indicates that the home page includes an animation at the header of the page which is 5 seconds in duration and is repeatable (infinite).The web screen includes all the styling attributes and operations affecting that particular screen only. Figure (5) shows an example of a web screen for a website home page.

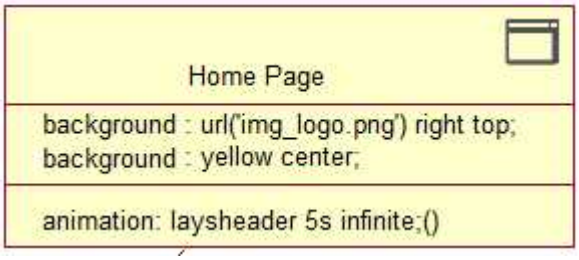


Figure (5) A web screen example

⟶ Basic arrows will be used to connect between style sheets and the affected web screens which represent the relationship between them. The arrow direction will always be from the style sheet towards the web screens to reflect their influence.

A WSD may contain one or more style sheets according to the website or web application style structure. Figure (6) shows a general structure of a WSD.

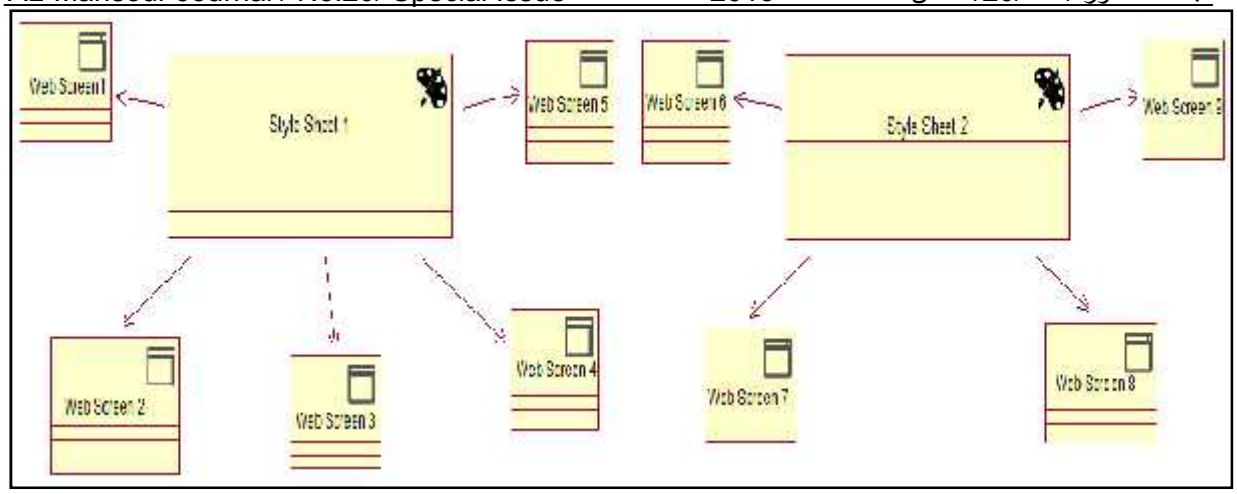


Figure (6) a general structure of a web style diagram.

6. Web Style Properties and Operations

As explained earlier the WSD reflects the company's style or organization's style and theme through a number of attributes and operations. These attributes and operations will be imposed when building the diagram as suitable for the web site or web application style requirements. But the attributes and operations should be within a predefined syntax so that they can be understood by anyone who reads the diagram as it is with any other UML diagram. Therefore syntax is proposed for each of the attributes and operations to provide a standard in defining these aspects.

In the WSD attributes can be differentiated from operations in their concept. Attributes represent static styling aspects such as page backgrounds, text properties, font properties … etc. while operations represent dynamic styling aspects that include movement such as animation, transition, marquees … etc. as these aspects include timing functions.

The Proposed Syntax for attributes is: ***Attribute: name, style, color, position, size, repeat;***
Where each elements description is shown in table (1):

Table (1) Attributes Elements Description

Attribute	*Specifies the type of attribute being described i.e. background, font, list, text …etc.*
name	*Specifies the name of the attribute if it was an external image or anything else that needs naming.*
style	*Specifies the style of the attribute element if it was font, text or anything else that needs style definition.*
color	*Specifies the color of the attribute element if it was font, text, background or anything else that needs color definition.*

position	*Specifies the position of the attribute element in the screen if it was font, text, background, image or anything else that needs position definition.*
size	*Specifies the size of the attribute element if it was font, text, background or anything else that needs size definition.*
repeat	*Specifies the repeating properties of the attribute element if it was font, text, background or anything else that needs repeating definition. I.e. how many times an image is repeated in the background of the screen.*

The Proposed Syntax for operations is: ***Operation: name, duration, timing-function, delay, iteration-count, direction;***
Where each elements description is shown in table (2):

Table (2) Operations Elements Description

Operation	*Specifies the type of operation being described i.e. animation, marquee, transition …etc.*
name	*Specifies the name of the operation if it was an animation or anything else that needs naming.*
duration	*Specifies the operation duration, or in other words how many seconds or milliseconds an operation takes to complete one cycle*
timing-function	*Specifies the speed curve of the operation*
delay	*Specifies the delay period before an operation would start*
iteration-count	*Specifies the number of times an operation should be repeated, and if it is continues it should be set as (infinite).*
direction	*Specifies whether or not the operation should play in reverse on alternate cycles.*

Of course not all the specifications mentioned in the syntax definition are needed at one time. Therefore it is possible to omit the specifications that are not of benefit in defining the attribute or operation wanted, but the important thing is to keep the sequence of specifications as defined in the syntax so it is possible to track the specifications omitted.
As an example, to define a blue background of a web page which is centered in the page and not repeated, this attribute can be defined as follows: **Background: blue, center;**

As it is obvious that only the color and position are needed in defining this attribute, and of course the sequence (color, position) is reserved.

Also to define an operation of a marquee displaying a company's highlighted news and information which is 15 seconds in duration and repeated infinitely, this operation can be defined as follows: **Marquee: company _info 15s infinite ();**

As it is obvious that only the operation name, duration and iteration count are needed in defining this operation and of course the sequence (name, duration, iteration count) is reserved.

7. Implementation of Web Style Diagram

The WSD was built upon the characteristics, semantics and diagrams of UML 2.0, that it will be able to cover the styling aspects according to the company's theme, As an example, Kit Kat, McDonalds and other companies have a distinguished font style while Nokia Company has a distinguished animation that shows two hands shaking with a particular rhythm. Other companies such as Zain or Asia cell have distinguished background colors and images. To explain the implementation of WSD two web site examples have been chosen and analyzed according to their style structure.

7.1. Orange Telecom Website

Orange is a French multinational telecommunications corporation. It is a global provider for mobile phone, landline, Internet, mobile internet, and IP television services, with 226 million customers as of December 2011 and, under the brand Orange Business Services, is one of the world leaders in providing telecommunication services to multinational companies. Orange telecom website has two common styling groups. It consists of pages that have black backgrounds as shown in figure (A.1) and figure (A.3) in Appendix A and pages that have white backgrounds as shown in figure (A.2) and (A.4) in Appendix A; also all of the mentioned pages have the orange logo on the top left part of the page. Also the home page has a main animation that reflects the company's strategy. These are the main style aspects that reflect the company's theme.

Figure (7) shows the WSD for Orange Telecom website using the proposed idea which consists of two style sheets: the first named *background style 1* and the second named *background style 2*. The names reflect the purpose of the style sheets which is the pages background style. The first style sheet *background style 1* contains two attributes:

1. "background: url ('img_logo.png') top left;"

This indicates that there is a common image between the home page and timeline page which is the company logo. This logo is located at the top left part of the pages.

2. "background: black center;"

This indicates that there is a common background color between the same two pages (home page and timeline page) which is black.

Now by examining the second background style sheet it is obvious that it also contains two attributes:

1. "background: url('img_logo.png') top left;"

This indicates that the company logo is also common between the 12 pages (Group Sites, Governance, Responsibility, Networks, Innovation, Finance, Press, Sponsorship, Live Orange Blog, Orange Jobs, Inside Orange, Products and Services) connected to this style sheet.

2. "background: white center;"

This indicates that all the previously mentioned 12 pages all have a white background centered and filling all the background.

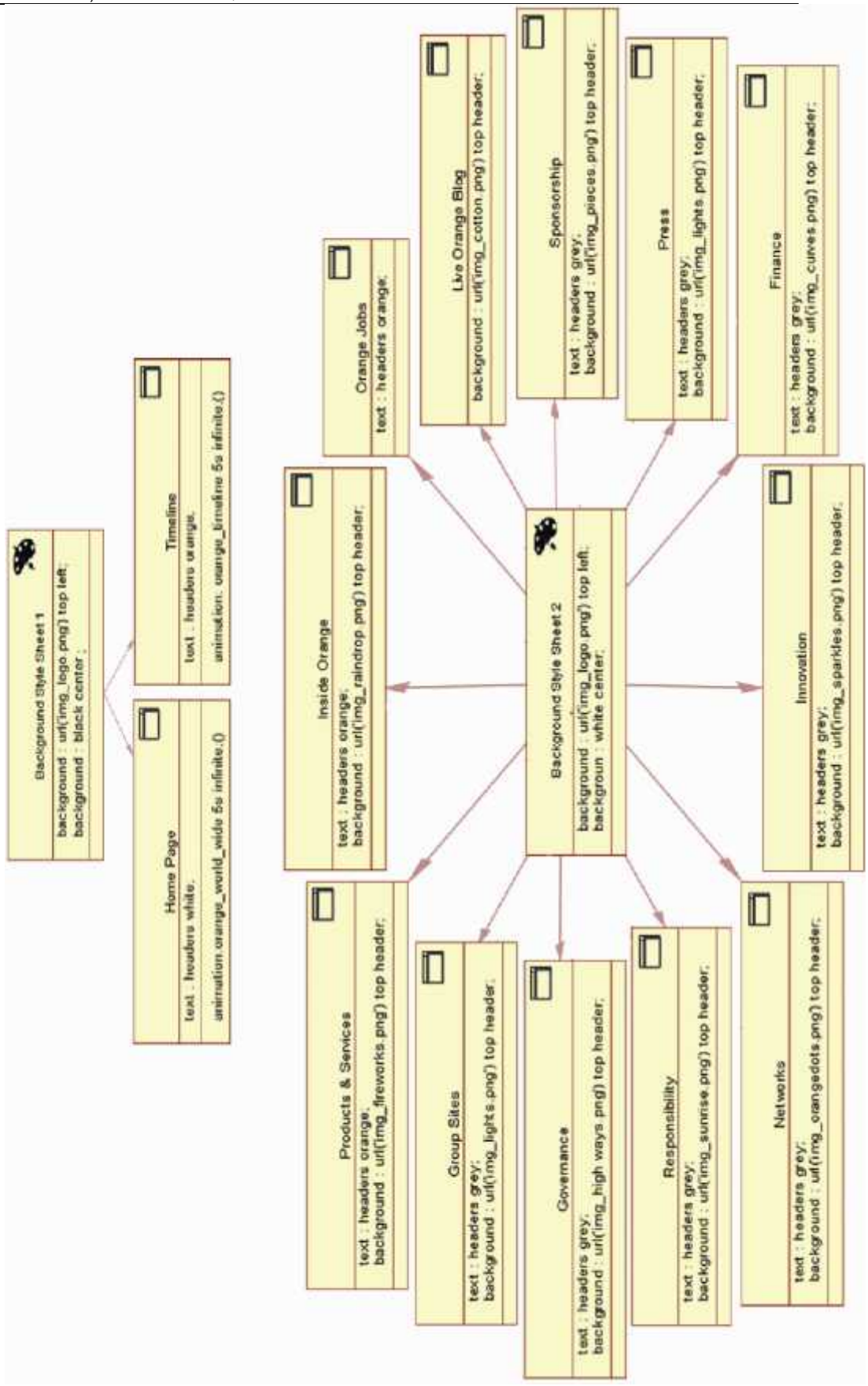


Figure (7) The Style Sheet Diagram for Orange Telecom Website

It is obvious that the two style sheets have one common attribute which is the company's logo, as all the website pages have the company's logo on the top left side of the page. The two style sheets are differentiated by the second attribute which shows that two pages of the website have a black background while the rest 12 pages have white backgrounds.

The rest of the elements of the diagram are web screens that hold the individual attributes and operations of each screen other than the ones defined in the style sheets.

The ***home page*** screen has the attribute "text: headers white;" which indicates that the home page text headers are overall white in color. In addition the home page has an operation "animation: orange _world _wide 5s infinite;()" which indicates that the page includes an animation called orange worldwide which is 5 seconds in duration and is repeated infinitely.

The ***timeline page*** also has one individual attribute and one individual operation, the attribute indicates that the page text headers are overall orange in color, and the operation indicates that the page includes an animation called orange timeline which is 5 seconds in duration and is repeated infinitely.

The ***group sites page*** has two attributes. The first attribute "text: headers grey;" which indicates that the page text headers are overall grey in color. The second attribute "background :url('img_lights.png') top header;" indicates that the page has a background image named lights and its location is at the top header of the page.

The ***governance page*** also has two attributes. The first attribute indicates that the page text headers are overall grey in color and the second attribute indicates that the page has a background image named highways and its location is at the top header of the page.

The ***responsibility page*** has two attributes. The first attribute indicates that the page text headers are overall grey in color and the second attribute indicates that the page has a background image named sunrise and its location is at the top header of the page.

The***networks page*** also has two attributes. The first attribute indicates that the page text headers are overall grey in color and the second attribute indicates that the page has a background image named orange dots and its location is at the top header of the page.

The ***innovation page*** also has two attributes. The first attribute indicates that the page text headers are overall grey in color and the second attribute indicates that the page has a background image named sparkles and its location is at the top header of the page.

The ***finance page*** has two attributes. The first attribute indicates that the page text headers are overall grey in color and the second attribute indicates that the page has a background image named curves and its location is at the top header of the page.

The ***press page*** also has two attributes. The first attribute indicates that the page text headers are overall grey in color and the second attribute indicates that the page has a background image named lights and its location is at the top header of the page. The sponsorship page has two attributes. The first attribute indicates that the page text headers are overall grey in color and the

second attribute indicates that the page has a background image named pieces and its location is at the top header of the page.

The ***live orange blog*** page has one attribute. The attribute indicates that the page has a background image named cotton and its location is at the top header of the page.

The ***orange jobs page*** has one attribute. The attribute indicates that the page text headers are overall orange in color. The inside orange page has two attributes. The first attribute indicates that the page text headers are overall orange in color and the second attribute indicates that the page has a background image named raindrop and its location is at the top header of the page.

The ***products and services page*** also has two attributes. The first attribute indicates that the page text headers are overall orange in color and the second attribute indicates that the page has a background image named fireworks and its location is at the top header of the page.

7.2 Lays Potato Chips Website

Lay's is the brand name for a number of potato chips varieties as well as the name of the company that founded the chip brand in 1932. Lay's chips has been marketed as a division of Frito-Lay, a company owned by PepsiCo since 1965.Lays website has one common styling group which consists of a yellow background and a lays logo at the top right part of the screen in addition to a common header animation on all the website pages. The home page also has a distinguished main animation for the new company product 'lays for no' as shown in figures (A.5) ,(A.6) in Appendix A.

Figure (8) shows the WSD for Lays website which will contain one style sheet named common style with two attributes and a single operation:

1. **"background : url('img_logo.png') right top;"**

This indicates that there is a common image between all the website pages which is the company's logo and its position is at the top right corner of the page.

2. **"background: yellow center;"**

This indicates that there is a common background color between all the website pages which is yellow.

3. **"animation: lays header 5s infinite;()"**

This is the common operation which indicates that all the pages include an animation called lays header which is 5 seconds in duration and is repeated infinitely.

The ***home page*** screen reflects the individual styling aspects related to this screen which is the operation **"animation: for no 5s;()"** indicating that the home page includes an animation named for no, this animation is five seconds in duration and is not to be repeated.

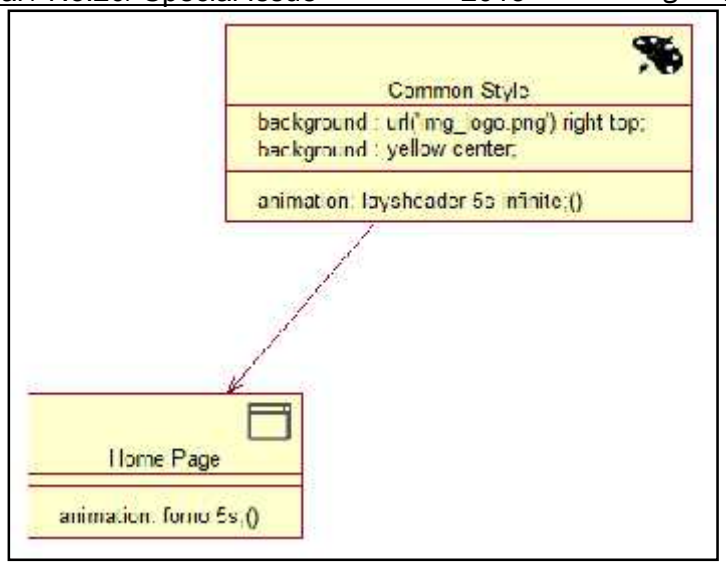


Figure (8) Lays Web Style Diagram

8. Conclusions

Web theme or web style is an essential web requirement that has been neglected in the UML analysis of web sites or web applications, although the visualization and user interaction are the most important factors in successive web requirement engineering.

WSD will give an early vision of the companies theme tips applied to its website or web application as it is already applied to all its products, TV advertisements, posters, company buildings and any other company related artifacts.

The proposed WSD represents a new dimension in the user experience modeling in UML which enriches the UX modeling with information about the style and theme of the system being analyzed. By adding this new diagram to the requirement analysis phase, the customer (institute requesting the web site or application) will be able to ensure that the resulting web site or application will meet their style identity.

The WSD was built upon the characteristics, semantics and diagrams of UML 2.0 but still is compatible with its subsequent versions as they all only include minor revisions up to version 2.5 which is still in the finalizing phase.

It can be also concluded that the WSD meets the conditions imposed by the UML extension mechanisms in order to provide a solid new UML diagram that meets all standards.

The WSD was specifically built for the requirement analysis phase to help system analyst, requirement engineers and customers to agree upon specific styling aspects that will show on the website or application and reflect the customers institute identity. The diagram can be used in further development phases like design by introducing more details into the diagram as the system development evolves and more information is available to the developers.

9. References

1. Koch N. , Knapp A. , Kozuruba S. ," Assessment of Effort Reduction due to Model-to-Model Transformations in the Web Domain", Springer-Verlag Berlin, Heidelberg ,(2012).
2. Kappel G., Proll B., Reich S., RetschitzeggerW.,"Web Engineering: The Discipline of Systematic Development of Web Applications", John Wiley & Sons(2006).
3. Koch N., Baumeister H., Hennicker R., Mandel L., "Extending UML to Model Navigation and Presentation in Web Applications",In Proc. of the Workshop Modeling Web Applications in the UML, UML'00 (2000).
4. Gustavo Rossi G., Pastor O., Schwabe D., OlsinaL.,"Web Engineering: Modelling and Implementing Web Applications", Springer,(2008).
5. Busch M., Masi M., Pugliese R., Tiezzi F., Koch N.," Towards Model-Driven Development of Access Control Policies for Web Applications", Springer,(2012).
6. Kozurub S., Koch N.,"Requirements Models as First Class Entities in Model-DrivenWeb Engineering", Springer,(2012).
7. Kozaczynski W., Thario J.," Transforming User Experience Model To Presentation Layer Implementations", Proceedings of the Second Workshop on Domain-Specific Visual Languages, OOPSLA, (2002).
8. Heumann J.," User experience storyboards: Building better UIs with RUP, UML, and use Cases", Rational Software, (2003).
9. Farhad J., "The UML Extension Mechanisms", term for Advanced Software Engineering (3C05), (2002).

اثراء هندسة متطلبات الويب باضافةعنصر النمطية للغة النمذجة الموحدة

م .د.زينب محمد حسين * محمد قاسم شوكت **

المستخلص

هندسة المتطلبات تعتبر اهم مرحلة من مراحل عملية بناء و تطوير البرمجيات بسبب الكلفة العالية لتصحيح اخطاء مرحلة المتطلبات , لذا فان هندسة متطلبات الويب تعتبر من اهم مواضيع البحث في يومنا هذا. من خلال ملاحظة معظم البحوث المطبقة يمكن الاستنتاج انها تركز على تطبيق تجربة المستخدم من خلال وجهة نظر المحتوى , الملاحة و العرض.

لغة النمذجة الموحدة (UML) هي لغة نمذجة مرئية متعددة الاغراض لتحديد و بناء و توثيق الانظمة والتي يمكن استخدامها في جميع مجالات التطبيق الرئيسية ومنصات تنفيذ.

في هذا البحث سوف يتم تقديم بعدا جديدا في نمذجة تجربة المستخدم من خلال لغة النمذجة الموحدة الذي يمثل عامل النمطية (style) أو بعبارة أخرى العوامل المميزة. كل شركة او مؤسسة تجارية لديها عوامل التصميم الخاصة بها والتي تعتبر ضرورية لنجاح مهمة الشركة او المؤسسة. و لهذا السبب يجب ان تعتبر ضمن المتطلبات (Requirements). سيتم ترجمة هذا البعد الجديد الى مخطط النمطية للويب (Web Style Diagram) مع كل العلاقات و القيود الخاصة به.

مخطط نمطية الويب سيمكن الزبون (المؤسسة الطالبة لموقع او تطبيق الويب) من التأكد بان موقع او تطبيق الويب الناتج سيكون مطابقا للهوية النمطية للمؤسسة.

* كلية المنصور الجامعة, قسم هندسة البرمجيات
** شركة زين للاتصالات

APPINDEX A
Web Page Snapshots

Figure (A.1) Orange Home Page

Figure (A.2) Inside Orange Page

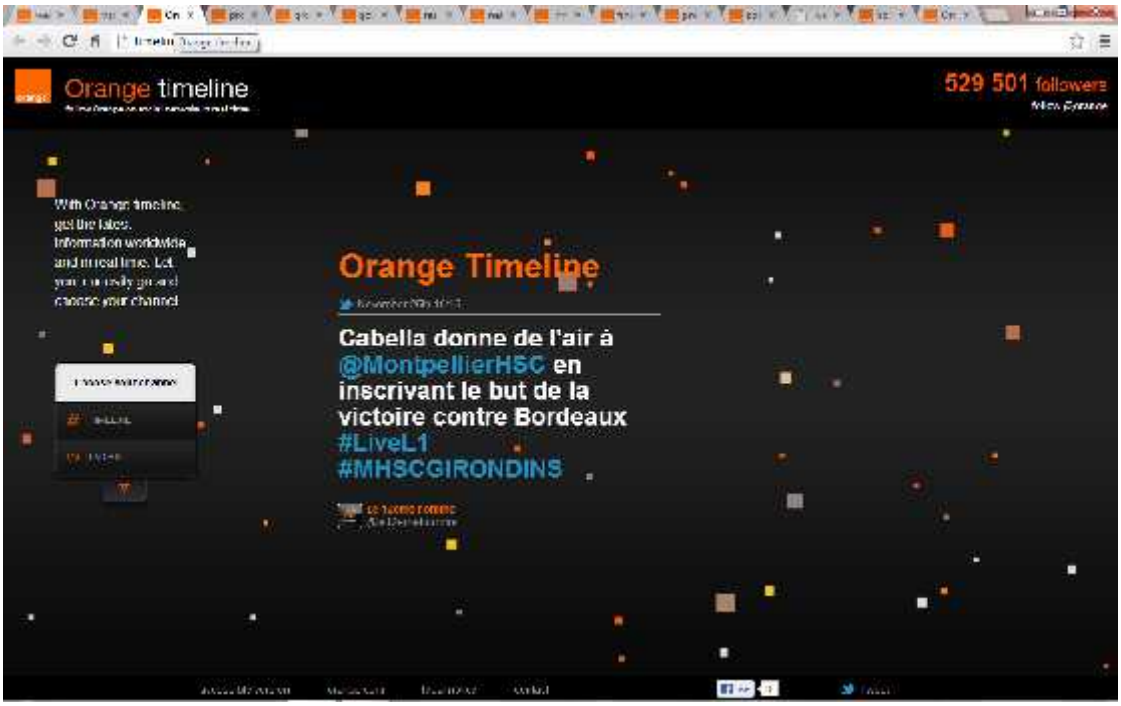


Figure (A.3) Orange Timeline Page

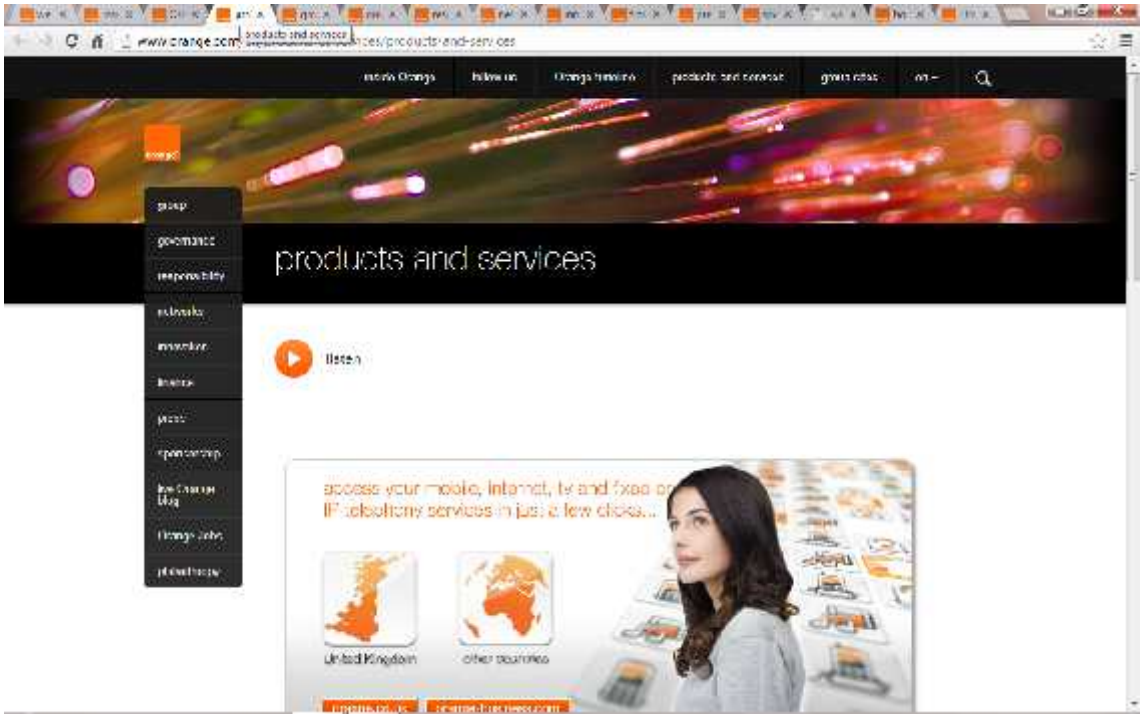


Figure (A.4) Products and Services Page

Figure (A.5) Lays Home Page

Figure (A.6) Great Moments Page