Reverse Engineering Method of Web Application to UML Presentation Model Using Vision Based Segmentation Method

Natheer Khasawneh, Safwan Al-Omari
Department of Software Engineering
Jordan University of Science and Technology
Irbid, Jordan
{natheer,ssomari}@just.edu.jo

Oduy Samarah
Department of Computer Engineering
Jordan University of Science and Technology
Irbid, Jordan
oasamarah09@cit.just.edu.jo

Abstract: In recent years, many web applications are available to use. Most of these applications are poorly modeled or not modeled at all. One of the main modeling techniques is presentation modeling in which the layout of the page is shown. In this paper we present a new reverse engineering method, which takes a web page as input and returns a UML presentation model that represents the page. We applied vision based segmentation method to determine the blocks of the page and the contents of the blocks such as anchor, image, etc. Finally we transform the block structure into a structured format based on UML presentation model.

Keywords
Reverse-engineering, Web Application, UML.

1. Introduction

In the last years many applications and services have evolved from being stand-alone monolithic application into web application. According to recent studies [3] most of exiting web applications lack proper modeling, which is necessary for maintenance, reengineering, and proper evolution to emerging web technologies. For the purpose of modeling legacy web applications, there is a need to have a reverse engineering method to extract models of existing web applications. Chikofsky describes Reverse Engineering **(**RE) as "*the process of analyzing a subject system to identify the system's components and their interrelationships and create representations of the system in another form or at a higher level of abstraction*" [1].

Current modeling languages and methodologies are not sufficient for capturing all aspects of web applications. UML for example is not sufficient to express the hyperlinks between different HTML pages. Three levels were introduced to model web applications:

1. Content modeling: focuses on modeling data in an HTML page.
2. Hyper text modeling: focuses on modeling links between HTML pages in a web application.
3. Presentation modeling: focuses on the layout of the items inside a particular HTML page.

In this paper we focus on the presentation model, which is used to model the page layout in a UML presentation model. The approach we follow in generating a presentation model is based on page segmentation method [8]. Page segmentation divides the page into different blocks according to its visual appearance when rendered in a web browser.

The paper is structured as follows. In Section 2 we provide an overview on the related works. In Section 3 we introduce the proposed approach in details, in section 4 we show case studies of our approach, finally in Section 5, we sketch concluding remarks and future works.

2. Related Works

In the literature there are many methods and tools of web reverse engineering which are built on the standard of RE techniques, these methods and tools can be used to describe and model the web applications with respect to different levels ((i) content, (ii) hypertext (iii) presentation):

UML based approaches:
The UML modeling language is the most widely used during the forward engineering design process and also in the most of web application (WA) reverse engineering techniques [3, 4], UML is convenient and stable to reengineer the different web applications through providing an extensive package of models such as Use-cases, activity- and class diagrams which can be used to describe and represent the behavior and the structure of the Web application in term of reverse engineering.

Data based approach:
This approach [2] based on the different structured techniques (Page segmentation) and on the model based techniques to identify the structure and meaning of data in a Web application and building a conceptual representation model of the Web application, in this approach the HTML page is divided into several blocks according to a cognitive visual analysis after that the specific patterns with these blocks are extracted to produce structural blocks and through these structural blocks a conceptual model is represented .

Ontology based approach:
This approach [6] relies on HTML pages analysis through extracting the useful information from web page and analysis the extraction information using the domain ontology and form the analysis results the UML conceptual schema is generated .

3. Reverse Engineering of Web Application to UML Presentation Model Using Vision Based Segmentation Method

In this paper we present a new method to reverse engineer exiting web applications. The method focuses on discovering the structure of the web page and presenting the structure in UML presentation model, as shown in Figure 1.

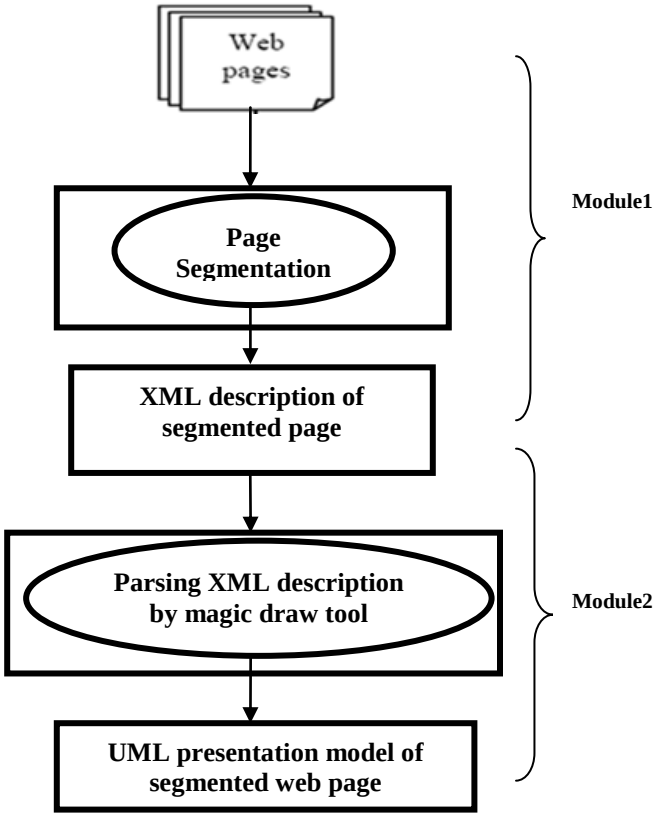

Fig 1: The architecture of our approach

The presented method consists of two stages: page segmentation stage and UML generation stage. Page segmentation stage accepts an HTML page as input and produces an XML description of the page. UML generation stage accepts an XML description of the segmented page and output the UML presentation in XML.

In the rest of this section, we describe, in detail, the page segmentation step and the generation of the UML presentation model step.

3.1 Page Segmentation

In the segmentation process, we use the VIPS (VIsion-based Page Segmentation) algorithm [8]. VIPS leverages the DOM tree and visual cues to extract the semantic structure of an HTML page.

The VIPS algorithm uses the page layout to divide the page into suitable blocks using the HTML DOM tree, and then it recursively divides the resulted blocks further into smaller blocks through the separators between these divided blocks. Typically, separators are images, horizontal, and vertical lines. Finally, the XML description for the web page structure is generated.

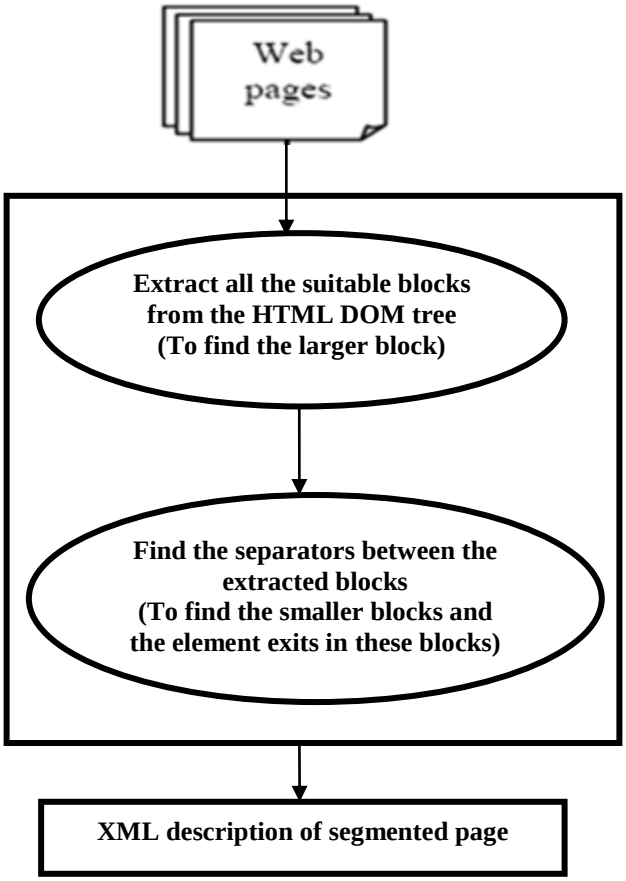

Fig 2: Flowchart for the segmentation process

3.2. Generation of XML description for block in segmented page:

After the page segmentation process, each resulted block and sub-block is associated with two XML descriptions: first XML description is the immediate result of the VIPS algorithm (VIPS XML description); whereas, the second XML description is derived from the first one and follows the Web UML format.

The VIPS XML description is very simple description and defined per block and sub-block in the segmented page. The VIPS XML description captures and describes some properties of each block in a web page such as determining whether the block contains any tables, images, anchors, and so on. The VIPS XML description also determines the coordinates of blocks in the HTML page. Figure 3 shows a simple example of the VIPS XML description.

```
<LayoutNode FrameSourceIndex="0" SourceIndex="50"
DoC="11" ContainImg="0" IsImg="false"
ContainTable="false" ContainP="0" TextLen="15"
LinkTextLen="0" DOMCldNum="1" FontSize="16"
FontWeight="400" BgColor="transparent"
ObjectRectLeft="4" ObjectRectTop="93"
ObjectRectWidth="164" ObjectRectHeight="34"
Content=" Members log-in " SRC="&lt;TD
class=navMenusHeader&gt;&lt;IMG
src=&quot;http://aff.kooora.com/i/icons/iball.gif&quot;&gt
;&amp;nbsp;Members log-in&lt;/TD&gt; " ID="1-1-3-2"
order="11"/>
```

Fig. 3: Example of VIPS XML format.

The Web UML XML description is produced from transforming the VIPS XML description to Web UML XML description. Web UML provides several notation elements that help in capturing the presentation aspects of a web application. Magicdraw[11] is one tool that has a plugin for Web UML; therefore, it can interpret and render the Web UML notation. The following is a brief description of some of these elements: «page» is a presentation group that contains all elements, which are presented together to the user in response to one request; «presentation class» groups a set of user interface elements representing a logical unit of presentation such as DIV element in HTML, there are also elements such as «anchor», «text», «image» and «button».

Figure 4 is a simple part of an HTML page, which is used to illustrate how UML Web modeling elements are used to build a presentation model. Figure 5, Figure 6, and Figure 7 depicts the resulted Web UML XML description. This example contains one text box, one button and a grouping element (DIV).

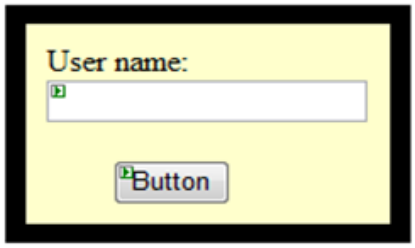

Figure 4: Simple part of a web page

Figure 5, Figure 6, and Figure7 show the UML XML description for our example and these three parts are necessary to build any blocks in UML presentation model. As shown in Figure 5, the UML XML description shows the type for each element in Figure 4 grouping element (DIV), textbox element, and button element as a presentation group, textbox and button elements for UML presentation model respectively.

```
<UWE_PROFILE:PRESENTATIONGROUP XMI:ID=
'_16_8_BE002F7_1275495351662_340026_2423'
BASE_PROPERTY=
'_15_0_1_11B203A6_1220619856562_410069_4909'/>
<UWE_PROFILE:TEXTINPUT XMI:ID=
'_16_8_BE002F7_1275495464341_145116_2463'
BASE_PROPERTY=
'_16_8_BE002F7_1275495361370_400990_2424'/>
<UWE_PROFILE:BUTTON XMI:ID=
'_16_8_BE002F7_1275495510638_160598_2467'
BASE_PROPERTY=
'_16_8_BE002F7_1275495408599_760043_2450'/>
```

Figure 5: UML XML descriptions specifying the type of elements.

[1] http://www.magicdraw.com/

```
<NESTEDCLASSIFIER XMI:TYPE='UML:CLASS'
XMI:ID='_16_8_BE002F7_1275495316200_978891_2421'
NAME='MEMBER LOGIN' VISIBILITY='PUBLIC'>
<OWNEDATTRIBUTE XMI:TYPE='UML:PROPERTY'
XMI:ID='_16_8_BE002F7_1275495361370_400990_2424'
VISIBILITY='PRIVATE' AGGREGATION='COMPOSITE'
TYPE='_16_8_BE002F7_1275495476131_720499_2464'/>
<OWNEDATTRIBUTE XMI:TYPE='UML:PROPERTY'
XMI:ID='_16_8_BE002F7_1275495408599_760043_2450'
VISIBILITY='PRIVATE' AGGREGATION='COMPOSITE'
TYPE='_16_8_BE002F7_1275495521433_746720_2468'/>
<NESTEDCLASSIFIER XMI:TYPE='UML:CLASS'
XMI:ID='_16_8_BE002F7_1275495476131_720499_2464'
NAME='USERNAME' VISIBILITY='PUBLIC'/>
<NESTEDCLASSIFIER XMI:TYPE='UML:CLASS'
XMI:ID='_16_8_BE002F7_1275495521433_746720_2468'
NAME='LOGIN' VISIBILITY='PUBLIC'/>
</NESTEDCLASSIFIER>
```

Figure 6: UML XML descriptions specifying the specification of elements.

In Figure 6, each element in the example has some specifications such as ID and name, and they have some properties such as visibility, which determines if the element is visible or not in UML presentation model.

In Figure 7, each element has size and coordinates properties which specify the size and the position of the element in the page, which is used to represent these elements in the UML presentation model and also specify the parts for each element (block) if they exist.

```
<MDELEMENT ELEMENTCLASS='PART' XMI:ID=
'_15_1_11B203A6_1225312275437_30737_4725'>
<ELEMENTID XMI:IDREF=
'_15_0_1_11B203A6_1220619856562_410069_4909'/
<GEOMETRY>511, 330, 315, 224</GEOMETRY>
<PARTS>
<MDELEMENT ELEMENTCLASS='PART' XMI:ID=
'_16_8_BE002F7_1275495361377_100975_2425'>
<ELEMENTID
XMI:IDREF='_16_8_BE002F7_1275495361370_400990_2424'/>
<GEOMETRY>602, 379, 175, 42</GEOMETRY>
</MDELEMENT>
<MDELEMENT ELEMENTCLASS='PART'
XMI:ID='_16_8_BE002F7_1275495408604_684882_2451'>
<ELEMENTID
XMI:IDREF='_16_8_BE002F7_1275495408599_760043_2450'/>
<GEOMETRY>602, 498, 175, 42</GEOMETRY>
      </MDELEMENT>
</PARTS>
 </MDELEMENT>
```

Figure7: UML XML descriptions specify the size, coordination, and parts for each element.

After understanding the format of Web UML XML descriptions, we need to transform the VIPS XML description formats to UML XML formats; the following Pseudo code in Figure 8 shows the transforming process:

```
1:  Input: VIPS XML description for blocks of segmented Web Page.
2: Output: UML XML description for blocks of segmented Web page.
 Begin
Set counter to 1          // counter for blocks.
Set total_blocks to N   // N is the total number of blocks and sub-blocks
in segmented page.
For counter to total_blocks step by 1
  Set info to the required specifications and properties for current block
in VIPS XML.
  Set new_info   to   Mapping _to_WebUMLXML (info).
   While block contains elements with VIPS XML description
     Set data to the required specifications and properties for specific
element in VIPS XML
     Set   new_ data to   Mapping _to_WebUMLXML (data).
  End While
   Add the information of block and the elements of block into Web
UML XML file.
End for
```

Figure 8: Pseudo code of transforming process.

4. Case Study of our Approach

In the previous sections we have presented a technique to segment a Web page into structural blocks and also presented the approach which is used to describe each block, sub-blocks, and elements in VIPS XML and Web UML XML descriptions.In this section we will illustrate these two steps by applying the proposed approach to a specific website.

4.1 Page Segmentation

Figure 9 shows the home page for goalzz web site[2], this page is segmented by VIPS algorithm into blocks as shown in Figure 10, some of these blocks are small and others are large in size depending on the structure of the page. The page layout of segmented page is shown in Figure 11.

[2] http://www.goalzz.com

Fig 9: Web page before the segmentation process

Fig 10: Web page after the segmentation process

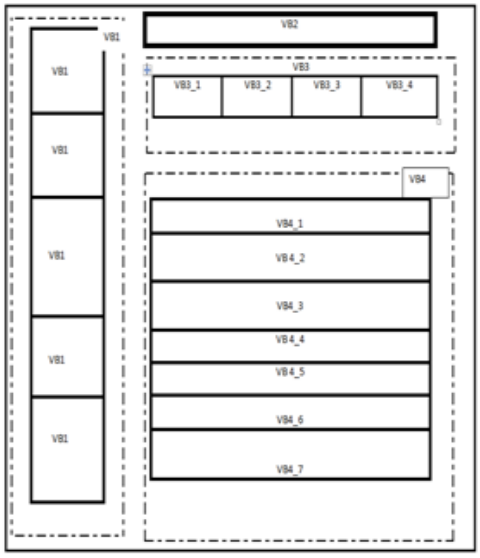

Fig 11: Page Layout for goalzz home page

4.2 Generation of XML Description

After the page segmentation process, the VIPS algorithm assigns each block, sub-block, and element with XML descriptions; these descriptions capture some properties of each block and sub-block in the web page. The VIPS XML format is mapped and transformed into the Web UML XML format as illustrated in Figure 2.

After the mapping and transformation process is achieved, the presentation model can be imported and manipulated in the Magicdraw modeling tool. Figure 10 shows the presentation model for goalzz home page.

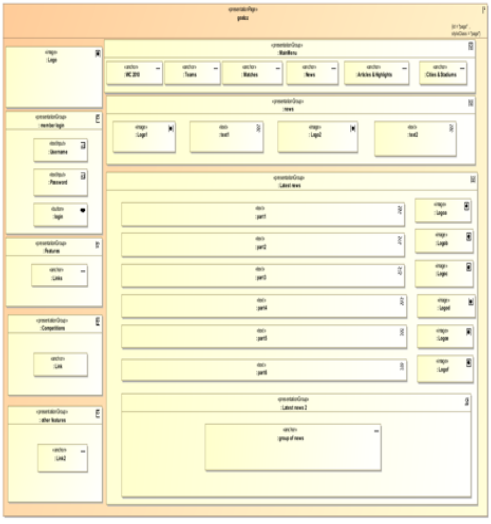

Fig 10: UML presentation model for goalzz home page.

5. Summary and Future Work

In this paper we presented a method of reverse engineering web applications into UML web presentation model. This issue is evolved from the more generic reverse engineering process, concentrating on the structure of the web page. We have presented an approach to the identification of structure in a web page and model this structure in UML presentation model. The approach relies on a number of structured techniques such as page segmentation.

Future work will concentrate on building a complete framework which automatically build the UML presentation model for any given application. The process of presenting the UML presentation model will be automated and will apply content mining along with the segmentation technique the find the accurately identify different blocks of the web page.

REFERENCES

[1] Chikofsky, E.J., Cross, J.H.: Reverse Engineering and Design Recovery: A Taxonomy.IEEE Software 7(1), 13–17 (1990).

[2] Roberto De Virgilio and Riccardo Torlone Universit`a Roma Tre, Italy: A Structured Approach to Data Reverse Engineering of Web Applications(2009)

[3] Di Lucca, G.A., Fasolino, A.R., Tramontana, P.: Reverse engineering Web applications: the WARE approach. Journal of Software Maintenance 16(1-2), 71–101 (2004)

[4] Chung, S., Lee, Y.S.: Reverse Software Engineering with UML for Web Site Maintenance.In: Proc. of the 1th Int. Conf. on Web Information Systems Engineering (WISE 2000), Hong Kong, China (2000)

[5] Laender, A., Ribeiro-Neto, B., Da Silva, A., Teixeira, J.S.: A brief survey of web data extraction tools. ACM SIGMOD Record 31(2), 84–93 (2002)

[6] Bouchiha, D., Malki, M., Benslimane, S.M.: Ontology based Web Application Reverse Engineering Approach. INFOCOMP Journal of Computer Science 6(1), 37–46 (2007).

[7] Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: Extracting Content Structure for Web Pages based on Visual Representation. In: Zhou, X., Zhang, Y., Orlowska, M.E. (eds.) APWeb 2003. LNCS, vol. 2642, pp. 406–417. Springer, Heidelberg (2003).

[8] Tilley, S., Huang, S. (2001). *Evaluating the Reverse Engineering Capabilities of Web Toolsfor Understanding Site Content and Structure: A Case Study*. Proc. 23rd International Conference on Software Engineering, IEEE, pp514-523.