

# Mockup To Web Page Conversion

Nupura Mohite

B.E. Computer, MMCOE  
Pune, India.

Prajakta Haldule

B. E. Computer, MMCOE  
Pune, India.

**Abstract**— In any web page development process the developer has to face many challenges. For example, insufficient requirement specification about the development from the user as well as the changes in the web pages with the upcoming technology integrated within it. In order to overcome these challenges the paper emphasis on a faster implementation of a web page with the help of mockups. There are many mockup tools available; Balsamiq Mockup Tool is one of them. Simple drag and drop functionality as well as offline version of the Balsamiq Mockup tool distinguishes it from the other mockup tools which makes creation of the graphical user interface(GUI) faster.

The paper proposes a methodology based on JAVA for conversion of mockups built in Balsamiq Mockup Tool into web page. User will simply input the Balsamiq mockup file to this application and on one click the application will convert the mockup provided to it into the web page. The main purpose for developing such application is to obtain the web page in a relatively less amount of time according to the user's specification as well as any modifications done in the mockup will quickly get reflected into the web page thereby saving time.

**Keywords**—*Balsamiq Mockup Tool; XML; HTML; SAX; DOM; bmml; SGML; W3C; Mockup; XML Parsing;*

## I. INTRODUCTION

The web page development process faces a number of challenges, such as the ones mentioned below:

- Insufficient requirements specification, since website development usually starts from unstructured and vague requirements.
- The gap between the requirements and the hypermedia design.
- Web pages constantly change their requirements so as to meet the volatile demands from the market.
- Request of integration with the latest web technology, as it is imperative to evolve quickly changing technology driven by competition.
- Web applications usually request short development cycle in order to sustain the competition from others.

In order to overcome these challenges the paper emphasis on a faster implementation of a web page with the help of mockups.

## II. RELATED WORK

### A. Conversion of PDF to HTML [9]

The main technique used in the conversion of PDF to HTML is the recognition of various segments in PDF page and then the information is extracted from these segments. Afterwards the actual conversion procedure starts. The main difficulty in this is the layout similarities between the PDF file and its equivalent HTML. The author made the use of open source tool called PDFBox for the extraction purpose of PDF and then applied certain methodologies such as the Text Segment Detection algorithm to recognize the text segments and finally the conversion starts segment by segment that converts the PDF to HTML.

### B. Mockup-driven Fast-prototyping Methodology for Web Requirements Engineering [5]

Web application development process is differs from the traditional software development in several ways. This author proposes methodology called MODFM - mockup-driven fast-prototyping methodology to help elicit and finalize system requirements, as well as facilitate adjustment to quickly changing user requirements typical to web applications. This MODFM Seamlessly integrates with the most recent web technologies as well as guarantees to deliver running web application prototypes to incrementally elicit, validate, and finalize user requirements early and consistent. Consequently it reduces the cost of nearly inevitable changes to the business rules, programming environment, or software design.

Our paper proposes the novel idea of developing the Application framework for the conversion of Balsamiq Mockup (.bmml file which is in format of XML) into the Web page (.html file).The paper uses the techniques such as SAX parsing of the .bmml file and ultimately converting it into .html file. The reason for developing such framework is rapid development of web page in time efficient manner with ease of modifications in web page.

## III. EXTENSIBLE MARKUP LANGUAGE

Extensible Markup Language (XML) is the latest buzzword on the internet, but it's also a rapidly maturing technology with powerful real-world applications, particularly for the management, display and organization of data.XML is

a subset of SGML with the same goals (markup of any type of data) , but with as much of the complexity eliminated as possible. XML is a set of rules for encoding documents in machine-readable form [6]. It was defined in the XML 1.0 Specification produced by the W3C [3], and several other related specifications. It can be used anywhere to transfer and store data. Also XML is platform and language independent, which means it doesn't matter that one computer maybe using, for example, Visual Basic on Microsoft operating system, and the other is a UNIX machine with a Java code. Really, any time one computer program needs to communicate with another program, XML is a potential fit for the exchange format.

However, it is important to realize that XML is not really a "language" at all, but a standard for creating languages which meet the XML criteria.

#### IV. XML PARSING

The mainstream methods of parsing xml are DOM(Document Object Model) and SAX (Simple API for XML).DOM is based on a tree structure and SAX is based on an event driven model[1][8].

##### A. DOM

DOM (Document Object Model) is an interface-oriented Application Programming Interface formulated by the W3C organization that allows for navigation of the entire document as if it were a tree of "Node" objects representing the document's contents. Since the tree structure is persistent in memory, we can modify data for the application and navigation through the tree. However, as to huge documents, parsing and load the entire document can be very low speed and waste lots of resource. In such a situation, we prefer other means to deal with it, such as SAX parser [7].

##### B. SAX

SAX is an event-driven interface in which a document is read serially and its contents are reported as callbacks to various methods on a handler object of the user's design as illustrated in Fig 1[7]. The analysis can start immediately, rather than waiting for the handling result of the entire data file. Data do not need to be stored in memory because the applications only check data while reading. In fact, the applications even do not need to analyze the whole document, it can just parse information data only when condition is satisfied and ignore the information, which is not interested in. Fig 2 illustrates the comparison of the SAX and DOM representation of the XML document.

The paper gives emphasis on using SAX as parsing method because the application need not traverse the XML back and forth and it comes with the added advantage that it need not build the memory representation of XML thus saving memory consumption as well as the output produced will be faster than that of DOM used as parsing method.

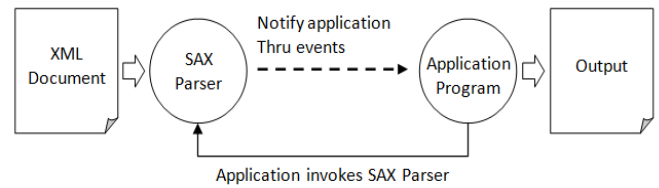


Fig. 1:SAX Parsing Mechanism

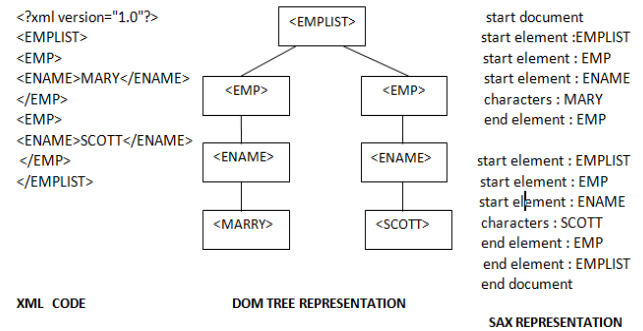


Fig. 2:Comparison of the SAX and DOM Mechanism

#### C. Comparison of SAX and DOM

Table I. Comparison of SAX and DOM [4]

API	ADVANTAGES	DISADVANTAGE
DOM	<ul style="list-style-type: none"> <li>• Easy navigation</li> <li>• Entire tree loaded into memory.</li> <li>• Random access to XML document.</li> </ul>	<ul style="list-style-type: none"> <li>• XML document must be parsed at one time.</li> <li>• It is expensive to load entire tree into memory.</li> </ul>
SAX	<ul style="list-style-type: none"> <li>• Entire document not loaded into memory which results in low memory consumption.</li> <li>• Allows registration of multiple Content Handlers.</li> </ul>	<ul style="list-style-type: none"> <li>• No built-in document navigation support.</li> <li>• No random access to XML document.</li> <li>• No support for modifying XML in place.</li> </ul>

#### V. BALSAMIQ MOCKUP TOOL

Balsamiq Mockups is a graphical user interface mockups builder application [2]. It allows the designer to arrange pre-built widgets using a drag-and-drop WYSIWYG editor. In computing, a WYSIWYG editor (what you see is what you get) is a system in which content

## VI. PROPOSED SYSTEM

The user initially uses the Balsamiq mock up tool to develop a framework of the web page since mockups are a faster method to create the GUI of any website. The web designer saves the mockups in Balsamiq (which comes with a .bmml extension). The paper proposes to develop an application to convert this .bmml file to an .html file. Conventionally the designer first creates the mockups and then separately has to create the website. This is a time consuming process. The paper tries to overcome this obstacle by making the application serve as an intermediate which will directly convert mockups into a web page thereby reducing the efforts of the designer. Fig. 4 illustrates the architecture of proposed system.

## VII. ALGORITHM

1. Read the .bmml file in Java.
2. Parse each component using SAX parsing till end of file. If start tag is encountered, parse the component till its corresponding end tag is obtained through SAX event handling mechanism.
3. Store each component i.e. all the properties of component in a form of name-value pair in suitable data structure like HashMap while parsing.
4. Create html string of each component with the help of HashMap in which properties of each component are stored previously.
5. Append the html string into either <head> or <body> tag according to the component parsed (for ex. button goes in body tag).
6. Repeat steps 2 to 5 till we get the html representation of all the components.
7. The required web page is obtained as the output.

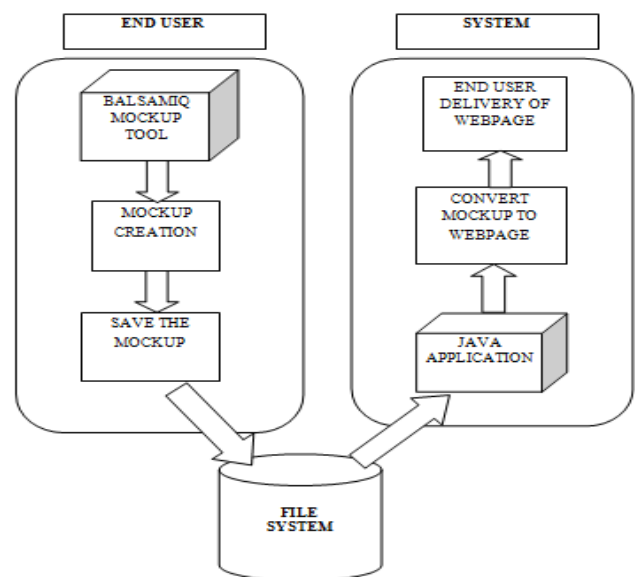


Fig. 5: System Architecture

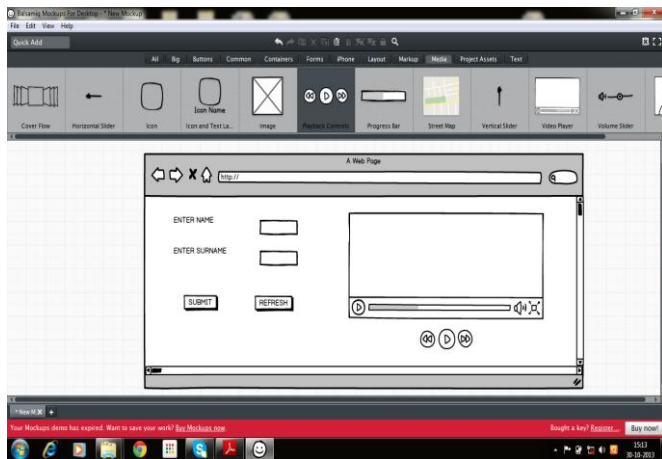


Fig. 3: Balsamiq Mockup Tool

(Text and graphics) displayed onscreen during editing appears in a form closely corresponding to its appearance when printed or displayed as a finished product. BMML is the Balsamiq Mockups Markup Language, the flavor of XML used by Balsamiq Mockups to save its data. Fig. 3 shows a snapshot of the Balsamiq Tool which the user would require while creating the mockup. It comprises of a number of basic components which are used while creating the mockup. Fig. 4 demonstrates the snapshot of Balsamiq Mockup Tool's component (Button Component) and its representation is in the form of XML is as follows:

```

<mockup version="1.0" skin="sketch" fontFace="Balsamiq
Sans" measuredW="542" measuredH="157" mockupW="72"
mockupH="27">
<controls>
  <control controlID="0"
controlTypeID="com.balsamiq.mockups::Button"
x="470" y="130" w="-1" h="-1"
measuredW="72" measuredH="27"
zOrder="0" locked="false" isInGroup="-1">
  <controlProperties>
    <text>SUBMIT</text>
  </controlProperties>
</control>
</controls>
</mockup>
  
```

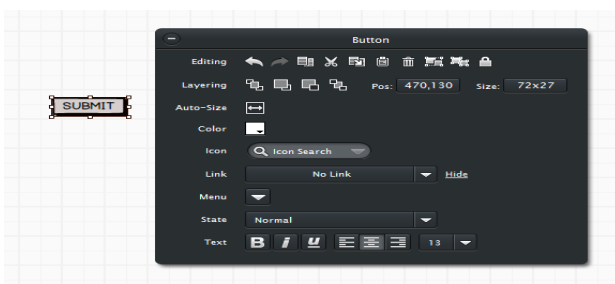


Fig. 4: Snapshot of Balsamiq component (button)

## VII.RESULTS

Fig. 6: Snapshot of a Customer Order Form Mockup

Fig. 7: Equivalent Html representation of the Mockup

Fig. 6 Illustrates a snapshot of the mockup created for a customer order form. The Fig. 7 illustrates the Html equivalent of the customer order form mockup.

## IX.CONCLUSION

Thus through this paper we have introduced the novel idea of developing the innovative framework for the conversion of mockup to web page. The usage of simple and effective mockup tool the website GUI can be created with relatively less amount of time and within a single click user will be able to view equivalent representation of the mockup in the browse i.e. desired web page. Moreover in future if the

user modifies the mockup, the respective modifications will get reflected into the web page within few seconds thus providing high flexibility.

## ACKNOWLEDGEMENT

We would like to express the deepest appreciation to Persistent Systems Pvt. Ltd. for giving us a opportunity to get associated with them through BE project sponsorship. Also we would like to thank our mentor in Persistent Systems Pvt. Ltd. Kapil Kulkarni who has shown the attitude and the substance of a genius. He continually and persuasively conveyed a spirit of adventure in regard to research. Without his supervision and constant help this dissertation would not have been possible.

We would like to thank our Institutional Head of Computer Department, MMCOE Pune, Professor Ram Joshi as well as our institutional project guide, Professor Rahul Dagade for their knowledgeable guidance and support.

## REFERENCES

- [1] Application of XML parser in Wireless Networking Miao Zhang School of Software Engineering, Tongji University,2011 IEEE.
- [2] Balsamiq Mockup Tool: <http://www.balsamiq.com>
- [3] World Wide Web Council: [www.w3.org](http://www.w3.org)
- [4] A Comparative Study and Benchmarking on XML Parsers, ICACT 2007.
- [5] Mockup-driven Fast-prototyping Methodology for Web Applications
- [6] Beginning XML, David Hunter.
- [7] Parsing an XML using SAX[Java Tutorials]: [docs.oracle.com/javase/tutorial/jaxp/sax/parsing](http://docs.oracle.com/javase/tutorial/jaxp/sax/parsing)
- [8] Introduction to XML Parser: [http://www.w3schools.com/xml/xml\\_parser.asp](http://www.w3schools.com/xml/xml_parser.asp)
- [9]Converting PDF to HTML Approach based on text Detection by Deliang JIANG and Xiaohu YANG College of Computer Science and Technology Zhejiang University, 2009 ICIS.