DERIVE A NEW MARKUP LANGUAGE FROM THE EXTENSIBLE MARKUP LANGUAGE (XML) FOR SUPPORTING ELECTRIC ENGINEERING

Areej Yousef Abu Hadbah[1] & Hebah H.O. Nasereddin[2]
Middle East University, Jordan
Email: areej_shadi@hotmail.com [1], hnasereddin@meu.edu.jo [2]

ABSTRACT

This paper aims to derive a new markup language from xml. These languages presented to the electric engineering area to help in drawing online electric circuit. all symbols and elements of electric engineering was derived along with supporting this language with a web application for drawing electric circuits via a web page without the need for drawing programmers-depending on the derived language containing engineering symbols. The drawn circuits using this application is easily-modified compared to circuit drawn using engineering programmers. Browsers (searching engines) can extract data from the circuit through the web since it is transformed in text format not an image format. Besides, drawing in this application is done within the environment of (SVG), a technology used to specify drawings instead of displaying them in pixel format.

Keywords: *SVG, XML, DTD, CML, ELECTRIC ENGINEERING.*

1. INTRODUCTION

With the development of technology the need for sharing and exchanging information over the Web increased. The programmer looks for developed ways and languages to match these needs. Many researches direct to xml language and customize it in our projects. XML (Extensible Markup Language) is a language designed to facilitate the ability to transfer data and to share information with others over the web. XML language used to derive other languages to achieve certain goals in different areas, such as math and chemistry that designed a derived markup language to get facilities in those areas,
(H.M.Deitel, 2001) "XML is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets. (walsh, 2000) "XML is Meta language which means it's a language that lets to create your own markup languages. XML is design to help store, structure and transfer data; (Holzner, 2003) "because it's written using plain text, it can be sent on the internet and handled by software on many different platforms".
There are many languages derived from XML such as XSLT, XPath, XQuery, XLink , XPointer, XSchema, XSL-FO, SVG, Attention Profiling Mark-up Language (APML), CXML, (ebXML),
The new application used to draw electrical circuit online, base xml language using descriptive code. There are many software program used to draw electrical circuit such as (Express SCH). (Trappey, 1997) describes the Schematic Design Software the (Express SCH) schematic design program design to use as our PCB (Printed circuit board) layout software, program having user interface. In this program, the user should download the program in their computer that require 10 MG space then start to draw by drag dropping the element on work space. The drawing output from this program export as image (bitmap pixels).
In this paper it is wanted to derive a new markup language base xml language. This language is derived to support electrical engineers and help him to draw their electrical circuit over the web. Then support that language with web application shows important of the new language and what facilities provide to the user by using xml in electrical engineering.

2. PROBLEM FORMULATION

XML can solve many problems in any field and provide facilities when using it. In this paper we will use this advantage to solve some problems facing electrical engineering, and discuss the problems facing engineers when browsing the web especially in their research and design their electrical circuit. The engineers can't draw their scheme on web directly; they should install the software programs to draw, after drawing the electrical circuit by software programs the outputs are represented as image.
This image has a large size and the search engines can't extract information from that image. Also, if engineers want to modify any element in the image, they face a problem. Also this program is not platform independent. But when

using a new markup language, they can exchange this image or circuit as a code (text) not as an image. This solve all problems listed because it offers less loading time and space because the output for drawing as text not as an image (xml code).

3.   METHODOLOGY

3.1 Create the new markup language for electrical engineering.

 To create the new markup languages for electrical engineering base xml. Should create XML document has the owner tags; xml language can create tags as needed. To create this document:

3.1.1 Collected and Classification electrical symbols which will be used.
Every element has image, all images we need to draw in the electrical circuit are collected in table and all images will be given the same size (48*48) pixels to be standard. The next figure (1) shows images collected to use in model.
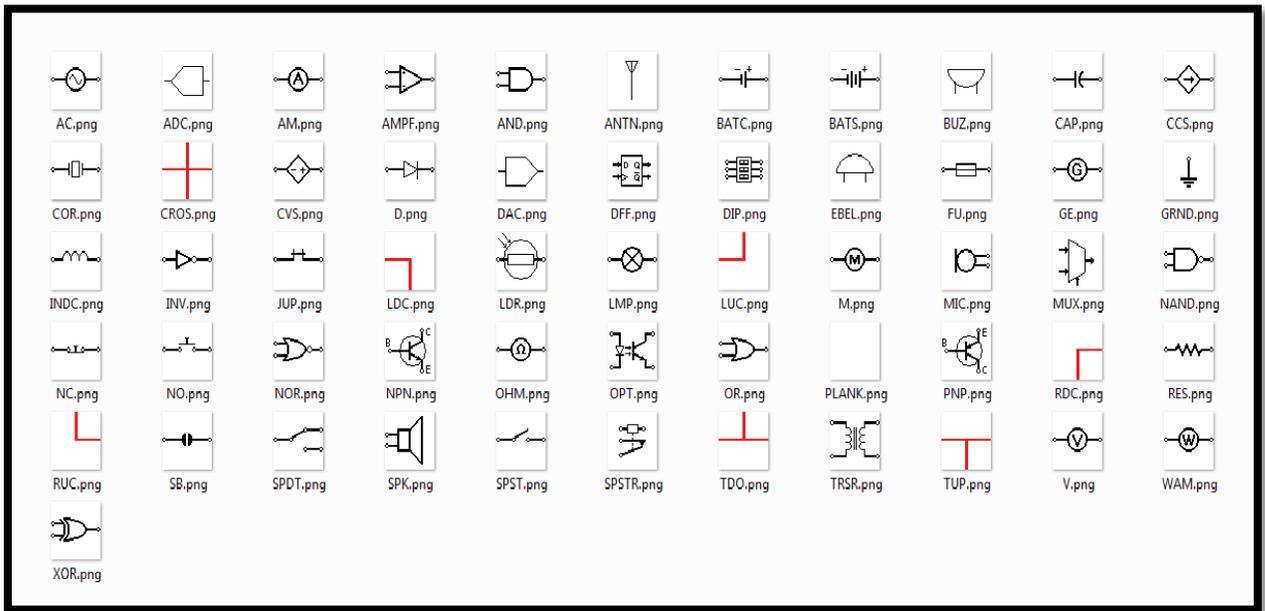

Figure (1) collected the shape for element.

3.1.2 Syntax and rules in XML:
The new language derived from xml language. So, to create this language, all condition and rules in xml language should apply with care about the xml syntax.

3.2 Write the descriptive Cod in text box.
Code must have the root element, this root is <C> for all circuit. Then call the element wanted to draw by tag from xml document and write all attribute in this tag. Following figure (2) explain the structure for descriptive code.

```
<C>

<   Element   id='string   value'   pos='number   value'   conside='number(1:4)'>

</Element>

</C>
```

Figure (2): structure to descriptive code.

Where:

(Id) represents element ID.

(Pos) represents the element position on screen.

(Conside) represents connection point from (my Side): to element want to connect with (Target Side).

Four (Conside) is: 1: first left side, 2: second left side, 3: first right side, 4: second right   side {l1, r1, l2, r2} separated by ";".

(Rotate) if you want to rotate element 90° "clockwise".

Determine the ID.

The user can write any (id) needed in tag, but this id should be unique.

Determine the position.

After writing the id the user should give position for elements. The positions determine by division the screen to small squares, every square has 48 high and 48 width pixels. Screen splits based image size as small square every square have a number from (0 to (48*48) -1). Screen spilt 48*48. Each square represent position number as shown in figure (3).

0	1	2	3	.	.	46	47
48	49	50	51	.	.	94	95
96	97	98	99	.	.	142	143
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
2255	.	.	.	.	.	2302	2303

Figure (3): Table represents position number on screen.

In html, the user should find (x, y) point for element, but that is not a flexible way in this model, so the user doesn't need to find (x, y) point for element, only give the position for element base table in figure (3)

Such as (0, 1, 2, 3, 4,….2303).When user write the position in text cod the model find (x, y) point automatically by equitation in figure (4).

```
p.X = (pos Mod 48) * 48
p.Y = Math.Floor(pos / 48) * 48
```

Figure (4): equitation to find (x, y) point automatically.

This equation finds the point (X, Y) for element when user gives the position number. If the user wants to draw element in position (4) the equation, get the point by this equation is (240, 0). The model collects this point automatically

(X=240,   Y=0). This equation gives more flexibility to determine position. Figure (5) shows (x, y) point for position (4).

(240,0)							
0	1	2	3	4	.	.	47
48	49	50	51	52	.	.	95
96	97	98	99	100	.	.	.
.							.
.							.
2256	.	.	.	.	.	.	2203

Figure (5): (x, y) point for square that have a position four.

(Olsson, September 26, 2008) In html the position is determined by three ways:
1. Position static 2.Position relative 3.Fixed Positioning. In this model the position is determined by table way as figure (3.21) and by relative way.

Determine connection side
Conside in this model mean connection point from (my Side) to element wanting to connect with (Target Side) If there more connection side spirited by ";".
figure (6) explains two elements every element have two conection side left 1, right 1.

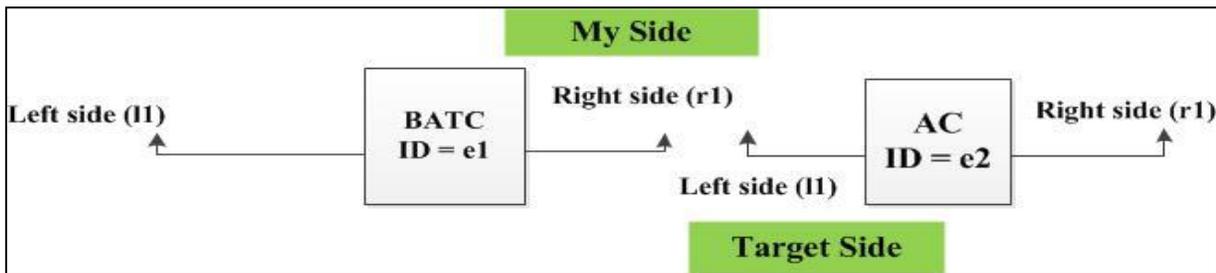Figure: (6) Explain connection side.

```
1  <C>
2  <BATC id='e1' pos='1'  conside='r1:e2.l1' > </BATC>
3  <AC id='e2' pos='3'  conside='r1:e1.l1'> </AC>
4  </C>
5
```

Figure: (7) Descriptive Code.

Main screen
This web site has two pages; the first one is to write the markup descriptive code. The second page is to show drawing to electrical circuit and have two buttons the first one is to draw circuit, second button is directed to help and search page.
In first page their text box. The user writes the descriptive code base a new markup language as input then press draw to get electrical circuit. After press draw button the model will check if there is no error in descriptive code before the electrical circuit will appear. The following figure (8) shows the main screen in web site has a code written by user in text box:

```
1
2  <C>
3    <RES id='r1' pos='0' conside='r1:r2.l1;r1:a1.l1' rotate='true' > </RES>
4    <RES id='r2' pos='96' conside='r1:g1.l1' rotate='true' > </RES>
5    <PLANK id='p6' pos='1' conside='l1:a1.l1'    > </PLANK>
6    <AND id='a1' pos='50' conside='r1:r3.l1'  > </AND>
7    <RES id='r3' pos='52' conside='r1:a2.l2'  > </RES>
8    <PLANK3 id='p5' pos='8' conside='l1:r1.l1;l1:r6.l1'  rotate='true'   > </PLANK3>
9    <PLANK id='p7' pos='5' conside='l1:p6.l1;l1:a2.l1'    > </PLANK>
10   <AND id='a2' pos='54' conside='!'  > </AND>
11   <BATC id='b1' pos='146' conside='l1:a1.l2'></BATC>
12   <RES id='r5' pos='150' conside='r1:a2.r1'></RES>
13   <PLANK id='p1' pos='148' conside='l1:r5.l1;l1:r3.r1'  > </PLANK>
14   <PLANK id='p2' pos='147' conside='l1:b1.r1;l1:r3.l1'  > </PLANK>
15   <PLANK id='p3' pos='193' conside='l1:r4.l1;l1:b1.l1'  > </PLANK>
16   <RES id='r4' pos='196' conside='!'  > </RES>
17   <PLANK id='p4' pos='198' conside='l1:r4.r1;l1:r5.r1'  > </PLANK>
18   |
19   <GRND id='g1' pos='192' conside='!'></GRND>
20   <D id='d1' pos='246' conside='!' rotate='true'></D>
```

Draw   **Help**

Figure (8) main screen have code written by user.

Now the user has finished the descriptive code, then presses the button (draw) to show the electrical circuitin figure (9).
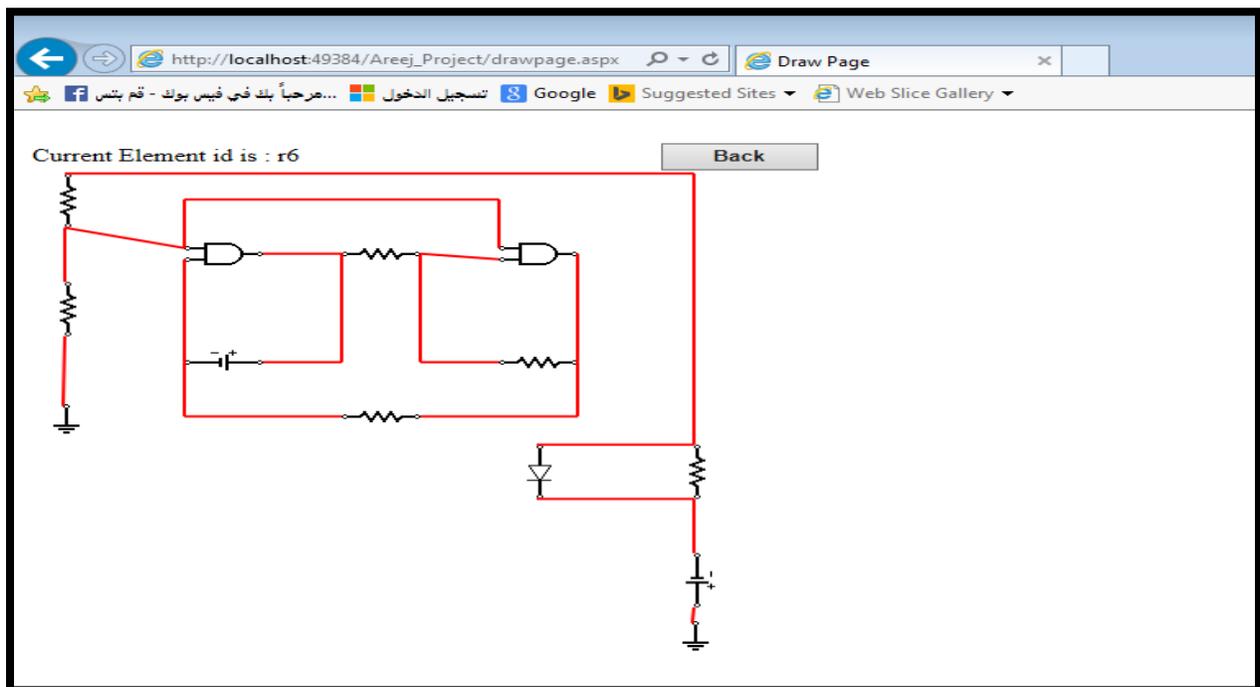

Figure (9): electrical circuit in draw page.

4. COMPARISONS AND RESULTS
The compression done between three software program is (Express PCB), (Circuit Maker 2000) and (SEE Electrical LT) with (Electrical Circuit drawing base XML)

	Express PCB	Circuit Maker 2000	SEE Electrical LT	Electrical Circuit drawing base XML
File of type	Bitmap image (.bmp)	WMF File (.wmf)	EMF File (.emf)	XML File (.xml)
Image size	96.3 KB (98,622 bytes)	8.06 KB (8,254 bytes)	2.63 MB (2,765,080 bytes)	870 bytes (870 bytes)
Size on disk	100 KB (102,400 bytes)	12.0 KB (12,288 bytes)	2.64 MB (2,768,896 bytes)	4.00 KB (4,096 bytes)
Requirement	Download software program size 10.1 MB	Download software program size 35.1 MB	Download software program size 110.1 MB	Connection with internet and internet browser
platform independent	Not platform independent	Not platform independent	Not platform independent	platform independent
Edit on circuit	Hard edit on image maybe Again draw	Hard edit on image maybe Again draw	Hard edit on image maybe Again draw	Easily modifiable because editing done on text not on image
Extract information from drawing	Can't extract information from drawing	Can't extract information from drawing	Can't extract information from drawing	Can extract information from drawing
Media type	Bitmap pixels	Bitmap pixels	Bitmap pixels	Drawing in Vectore.SVG
Accuracy with Zoom	Accuracy affect with zoom	Accuracy affect with zoom	Accuracy affect with zoom	Don't affect with zoom
Sharing and exchange information	Not flexible	Not flexible	Not flexible	More flexibility

Table (1): the comparisons table.

5.   ACKNOWLEDGEMENTS

6.   REFERENCES

[1] H.M.Deitel, p. (2003). xml how to program. new jersey : prentice hall.
[2] Holzner, S. (2003). Teach Yourself XML in 21 Days (3rd Edition ed.). Sams.
[3] http://www.w3.org/.
[4] Lu, W., Chiu, K., & Pan, Y. (2006, September). A parallel approach to XML parsing. In Grid Computing, 7th.
[5] IEEE/ACM International Conference on (pp. 223-230). IEEE.
[6] Murray-Rust P, R. H. (2003). Chemical markup, XML, and the World Wide Web. 4. CML schema. National Center for Biotechnology Information, U.S. National Library of Medicine.
[7] Olsson, T. (September 26, 2008). CSS absolute and fixed positioning. Attribution-NonCommercial-ShareAlike 2.5 Generic.
[8] walsh, n. (2000). a technical introduction to xml. 4 , 49.