IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 5, Issue 6, June 2018

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

The Architecture of Web-based P-CAD Systems

Ayedh Abdulaziz Mohsen

Department of Math's and Computer, Faculty of Science, IBB University, Yemen

Abstract*:*This article shows the architecture of modern web-based based applications, which are a collection of website elements that perform certain programmed actions, and which are the core of the **modular web-based P-CAD**. Web-based applications are designed to operate on web servers and use web browsers as the user interface. Typically, web-based applications are developed as part of a "client–server "architecture. However, the server side can vary in terms of its architecture solutions, as this article demonstrates. In addition, we will also demonstrate an example of the architecture of a particular **modular web-based p-cad**.

Keywords: web-server, modular p-cad, application architecture,"client–server "architecture.

I. INTRODUCTION

Currently, one of the P-CAD systems 'development lines is the development of distributed architectures and build on their basis module p-cad pcb based on Web. The implementation of such a distributed **P-CAD** architecture requires developing a special web-based application that would run and synchronize subsystems on the client and server sides, as well as transfer data between the client and server subsystems. Consider the basic methods for developing web-based applications. Initially, the World Wide Web (WWW) was considered by its creators as an information exchange space, in which people and computers would communicate with each other [1]. Therefore, the first web-based applications were simple file servers that returned static HTML pages to the requesting clients. That is, the Web was a document-oriented network at its early stage.

The next stage in the development of the Web was the emergence of the concept of applications that were based on interfaces such as CGI (or FastCGI), and later on ISAPI. The Common Gateway Interface (CGI) is a standard interface with servers, which allows executing server applications called via URL. The input information for such applications was the contents of the HTTP header (and the request body when using the POST protocol). CGI applications generated an HTML code that was returned to the browser. The main problem with CGI applications was that each client request resulted in the execution of a CGI program by the server in real time, loading it into a separate address space[2].

The emergence of the Internet Server API (ISAPI) allowed not only to solve performance issues typical of CGI-applications, but also to provide developers with a more functional software interface. ISAPI DLL could be associated with file name extensions through a special meta base. The two mechanisms (CGI and ISAPI) served as the basis for creating the first type of web-based applications, which executed the server code depending on any client-side actions. Thus, it became possible to dynamically generate the web pages 'content, which ceased to be purely static. The ISAPI interface is a peculiar feature of the Microsoft Internet Information Server. ISAPI applications are dynamic loadable libraries (DLLs) that are executed in the web server's address space. After a while, other web servers also learned to execute applications in the form of libraries [3]. In the case of Netscape web servers, this programming interface was called NSAPI (Netscape Server API). The fairly popular Apache web server also was able to run web-based applications implemented as libraries. Such libraries are called Apache DSO (Dynamic Shared Objects).

Note that as the amount of data used and the number of website visitors grows, the requirements for reliability, performance, and scalability of web-based applications also increase. The next stage in the evolution of such applications was the separation of business logic implemented in web-based applications, and often also separation of the data processing and transaction services from their interface. In this case, usually a web-based application itself only contains the so-called presentation section, while business logic, data processing, and transaction implementation are transferred to the application server as business objects. In addition to the access to corporate and partners 'data, the next step in the evolution of web-based applications was to gain access to corporate applications. This problem is solved through the integration of web-based applications, called corporate portals, with internal information systems of enterprises and with applications that provide interaction with customers and partners. Often, a portal solution includes website content management tools, as the amount of data available to users through the large companies 'websites is currently so huge that their manual management seems impossible [4].

Many of the above tasks arising in the development of modern web-based applications currently start being fulfilled by web services, i.e. software elements that do not depend on the platform, object model, and client software and can be called from client-side web-based applications (as well as from the web services themselves) through the SOAP protocol based on HTTP and XML. Web services are described using WSDL, an XML-like language, while the UDDI interface is used to organize web service registries, in which developers and companies can search for the services they

DOI 10.17148/IARJSET.2018.561

need, as well as publish information about their own services [5]. We have reviewed the possible architectures for building web-based applications and concluded that distributed P-CAD should be implemented based on the PHP technology. Consider using this technology in the development of a modular web-based PCB design P-CAD.

II. TRADITIONAL P-CAD SYSTEM ARCHITECTURE

The traditional P-CAD architecture (Figure 1) includes separate subsystems written in procedural or object-oriented programming languages that have powerful tools for handling various data structures such as lists, queues, stacks, trees, graphs, etc. Data exchange between subsystems is provided in the following ways: through a common memory area; through a binary file system; by means of specialized description languages (text files)[6], (Figure 3 Drawing electronic circuit using the schematic model, which produces a text file figure 2)

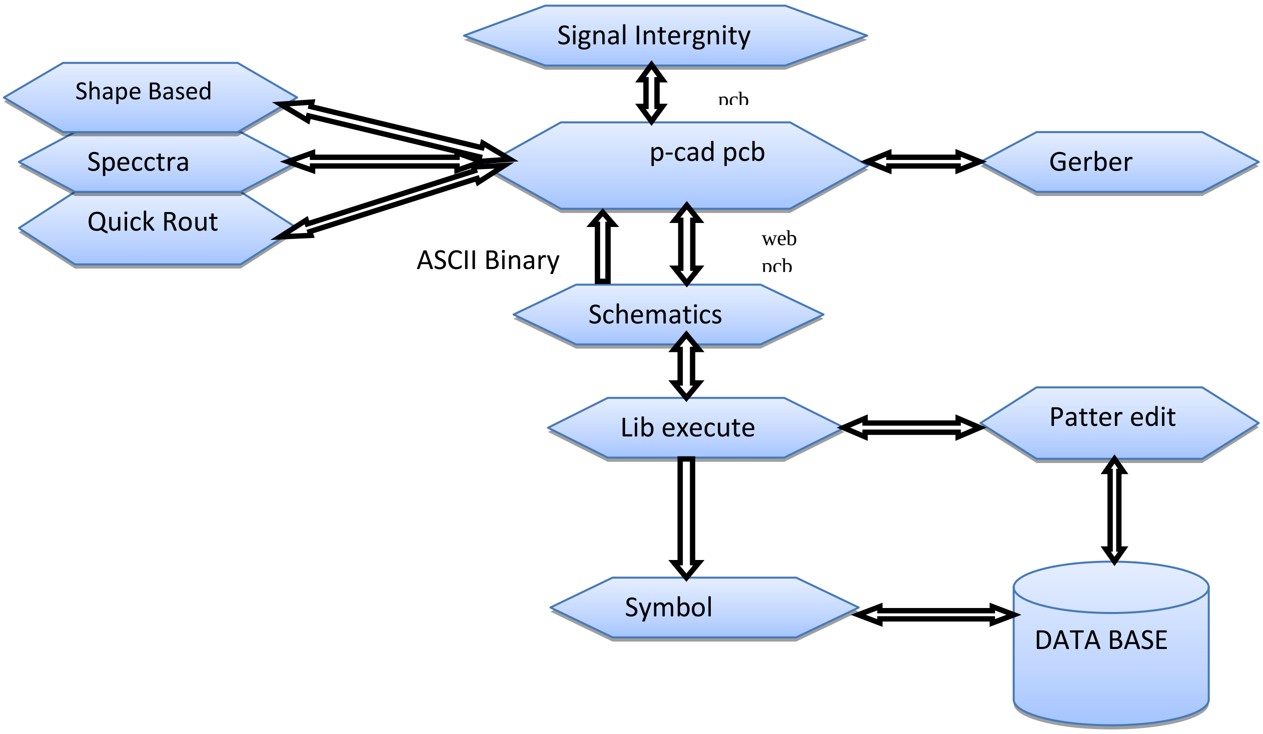


Fig1. Traditional P-CAD system architecture

Currently, the development and adaptation of special web-based versions of desktop P-CAD that would allow automating the process of designing objects of various physical nature on an enterprise scale, is relevant.

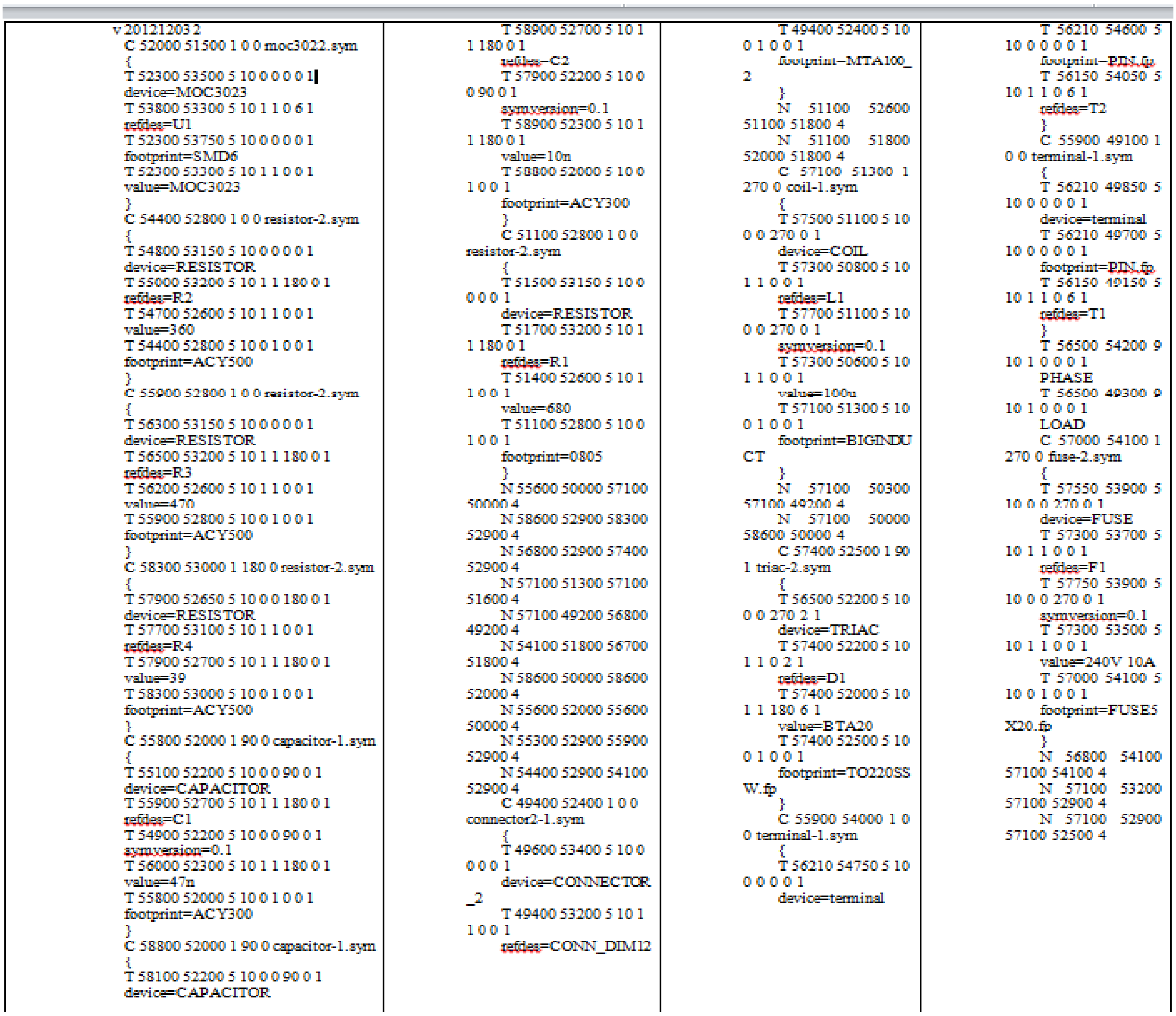
Consider the architecture of the web-based version of a similar P-CAD using the example of PCB P-CAD, which has become the most popular tool among electronic appliance developers [7].

The core of the P-CAD system is P-CAD PCB, the most well-known modification of SPECCTRA tracing, developed is an integrated software suite for the design of multi-layer printed circuit boards (PCBs) of radio electronic equipment (RE). It is adapted to the Windows operating environment and uses all the settings and features of the latter [8, 9].

The P-CAD includes the following subsystems:

- P-CAD includes the following software modules: P-CAD Library Executive, P-CAD Schematic, P-CAD PCB, P-CAD AutoRoute's, Symbol Editor, Pattern Editor, InterPlace PCS, Relay, Signal Integrity (Figure 1).
- The Library Executive utility consists of Library Manager, Symbol Editor, and Pattern Editor (to edit radio parts land patterns).
- P-CAD Schematic is a graphical electric circuit editor, designed for the development of electrical circuitry and used to create conditional graphic symbols of individual radio parts (.sch files).
- P-CAD PCB is the PCB graphic editor for developing the PCB design parameters, including: PCB size, conductor width, gap size, contact pad size, screen layer configuration, marking, radio part positioning, non-automatic tracing of conductors and forming processing equipment control files.
- P-CAD AutoRoute's is designed for automatic tracing of PCB conductors. It includes two AutoRoute's: Quick Route for designing the PCB pattern in not very complex electrical circuits and the grid less Shape-Based Router designed for multilayer PCBs with a high density of radio parts layout.

- Symbol Editor is an element symbol editor (.sym files), designed to create conditional graphic symbols of electrical circuit parts.
- Pattern Editor is the land pattern editor (.pat files), designed for the development of structural parts 'landing patterns on PCBs.
- Relay is the program for PCB viewing, radio part arrangement, basic attribute assignment, and processing limitation control.
- Signal Integrity is the PCB electrical parameter analysis software.
- SPECCTRA is the program for radio part arrangement on PCBs and manual, interactive, and automatic conductor tracing. Thanks to the grid less technology, it is a very efficient tracer for complex and high-density PCBs. Information about the PCB is transmitted to SPECCTRA using the PCB editor[10].



```
v 20121203 2
 C 52000 51500 1 0 0 moc3022.sym
 {
 T 52300 53500 5 10 0 0 0 0 1
 device=MOC3023
 T 53800 53300 5 10 1 1 0 6 1
 refdes=U1
 T 52300 53750 5 10 0 0 0 0 1
 footprint=SMD6
 T 52300 53300 5 10 1 1 0 0 1
 value=MOC3023
 }
 C 54400 52800 1 0 0 resistor-2.sym
 {
 T 54800 53150 5 10 0 0 0 0 1
 device=RESISTOR
 T 55000 53200 5 10 1 1 180 0 1
 refdes=R2
 T 54700 52600 5 10 1 1 0 0 1
 value=360
 T 54400 52800 5 10 0 1 0 0 1
 footprint=ACY500
 }
 C 55900 52800 1 0 0 resistor-2.sym
 {
 T 56300 53150 5 10 0 0 0 0 1
 device=RESISTOR
 T 56500 53200 5 10 1 1 180 0 1
 refdes=R3
 T 56200 52600 5 10 1 1 0 0 1
 value=470
 T 55900 52800 5 10 0 1 0 0 1
 footprint=ACY500
 }
 C 58300 53000 1 180 0 resistor-2.sym
 {
 T 57900 52650 5 10 0 0 180 0 1
 device=RESISTOR
 T 57700 53100 5 10 1 1 0 0 1
 refdes=R4
 T 57900 52700 5 10 1 1 180 0 1
 value=39
 T 58300 53000 5 10 0 1 0 0 1
 footprint=ACY500
 }
 C 55800 52000 1 90 0 capacitor-1.sym
 {
 T 55100 52200 5 10 0 0 90 0 1
 device=CAPACITOR
 T 55900 52700 5 10 1 1 180 0 1
 refdes=C1
 T 54900 52200 5 10 0 0 90 0 1
 symversion=0.1
 T 56000 52300 5 10 1 1 180 0 1
 value=47n
 T 55800 52000 5 10 0 1 0 0 1
 footprint=ACY300
 }
 C 58800 52000 1 90 0 capacitor-1.sym
 {
 T 58100 52200 5 10 0 0 90 0 1
 device=CAPACITOR

    T 58900 52700 5 10 1
1 180 0 1
    refdes=C2
    T 57900 52200 5 10 0
0 90 0 1
    symversion=0.1
    T 58900 52300 5 10 1
1 180 0 1
    value=10n
    T 58800 52000 5 10 0
1 0 0 1
    footprint=ACY300
    }
    C 51100 52800 1 0 0
resistor-2.sym
    {
    T 51500 53150 5 10 0
0 0 0 1
    device=RESISTOR
    T 51700 53200 5 10 1
1 180 0 1
    refdes=R1
    T 51400 52600 5 10 1
1 0 0 1
    value=680
    T 51100 52800 5 10 0
1 0 0 1
    footprint=0805
    }
    N 55600 50000 57100
50000 4
    N 58600 52900 58300
52900 4
    N 56800 52900 57400
52900 4
    N 57100 51300 57100
51600 4
    N 57100 49200 56800
49200 4
    N 54100 51800 56700
51800 4
    N 58600 50000 58600
52000 4
    N 55600 52000 55600
50000 4
    N 55300 52900 55900
52900 4
    N 54400 52900 54100
52900 4
    C 49400 52400 1 0 0
connector2-1.sym
    {
    T 49600 53400 5 10 0
0 0 0 1
    device=CONNECTOR
_2
    T 49400 53200 5 10 1
1 0 0 1
    refdes=CONN_DIM12

    T 49400 52400 5 10
0 1 0 0 1
    footprint=MTA100_
2
    }
    N   51100   52600
51100 51800 4
    N   51100   51800
52000 51800 4
    C  57100  51300  1
270 0 coil-1.sym
    {
    T 57500 51100 5 10
0 0 270 0 1
    device=COIL
    T 57300 50800 5 10
1 1 0 0 1
    refdes=L1
    T 57700 51100 5 10
0 0 270 0 1
    symversion=0.1
    T 57300 50600 5 10
1 1 0 0 1
    value=100u
    T 57100 51300 5 10
0 1 0 0 1
    footprint=BIGINDU
CT
    }
    N   57100   50300
57100 49200 4
    N   57100   50000
58600 50000 4
    C 57400 52500 1 90
1 triac-2.sym
    {
    T 56500 52200 5 10
0 0 270 2 1
    device=TRIAC
    T 57400 52200 5 10
1 1 0 2 1
    refdes=D1
    T 57400 52000 5 10
1 1 180 6 1
    value=BTA20
    T 57400 52500 5 10
0 1 0 0 1
    footprint=TO220SS
W.fp
    }
    C 55900 54000 1 0
0 terminal-1.sym
    {
    T 56210 54750 5 10
0 0 0 0 1
    device=terminal

    T  56210  54600  5
10 0 0 0 0 1
    footprint=PIN.fp
    T  56150  54050  5
10 1 1 0 6 1
    refdes=T2
    }
    C  55900  49100  1
0 0 terminal-1.sym
    {
    T  56210  49850  5
10 0 0 0 0 1
    device=terminal
    T  56210  49700  5
10 0 0 0 0 1
    footprint=PIN.fp
    T  56150  49150  5
10 1 1 0 6 1
    refdes=T1
    }
    T  56500  54200  9
10 1 0 0 0 1
    PHASE
    T  56500  49300  9
10 1 0 0 0 1
    LOAD
    C  57000  54100  1
270 0 fuse-2.sym
    {
    T  57550  53900  5
10 0 0 270 0 1
    device=FUSE
    T  57300  53700  5
10 1 1 0 0 1
    refdes=F1
    T  57750  53900  5
10 0 0 270 0 1
    symversion=0.1
    T  57300  53500  5
10 1 1 0 0 1
    value=240V 10A
    T  57000  54100  5
10 0 1 0 0 1
    footprint=FUSE5
X20.fp
    }
    N   56800   54100
57100 54100 4
    N   57100   53200
57100 52900 4
    N   57100   52900
57100 52500 4
```

Fig.2 Text file(.sch) for schematic model

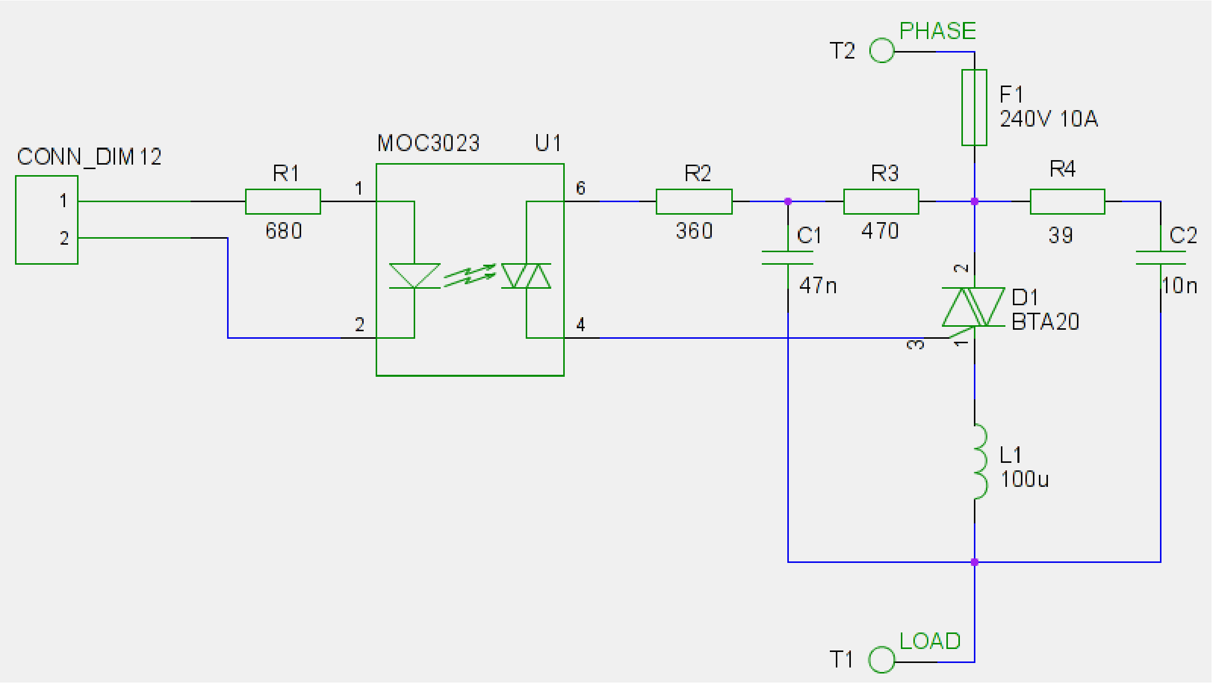


Fig. 3 Graphic file in p-cad schematic model

The process of designing a PCB for a device in P-CAD is divided into the following steps:

1. Creating a basic electrical circuit using the software package P-CAD Schematic. The result of the work is saved in a file with the extension .sch.
2. On the basis of the schematic diagram, information is generated about the connections between the elements and the types of components used. Therefore, the second stage in the creation of PES is the creation of a file of links describing the electrical connections of the elements and their types. After you complete this step, the output file format will be in .net format, which is saved in the connect list
3. Next in the P-CAD PCB based on the .net file. receive a drawing of the pcbs with the placed components, on which the lines of electrical connections between their terminals are indicated, (drawings of the component bodies are automatically loaded from the integrated .lib library). This operation is called packing the scheme on the pcbs, resulting in a file with the extension .pcb graphic file (fig5) and text file(In the link[43]),Which is the result of the .sch file.
4.After that you need to create a drawing of the overall dimensions of the printed circuit board. Actions are performed manually.
5. Next, automatic layout of the card's tracks using the P-CAD Quick Route auto-tracer is made, the routing strategy (rules) must first be set in the R-CAD PCB [11].
6.The P-CAD system uses integrated libraries that contain graphic and text information about components. Graphically, symbols and component bodies are represented, and text information, number of sections in the form, numbers and names of the conclusions. If there is no necessary element in the library when creating a schematic diagram, you can create it. For this, the programs included in the package of the P-CAD package (Library Manager (Library Executive, Symbol Editor-editor of the symbols of the component, Pattern Editor-packages editor) are used [12].

The data are transferred between theP-CAD Schematics and P-CAD PCB subsystems using a text file with a description of the circuit in the system's input language. The simulation data are transferred from PCAD to SPECCTRA using a binary file[13].The **P-CAD** actively uses libraries of the circuit component parameters, which are text files in the language of the model description specification and the circuit component parameters[14].

A. Flowchart for Traditional P-CAD system architecture

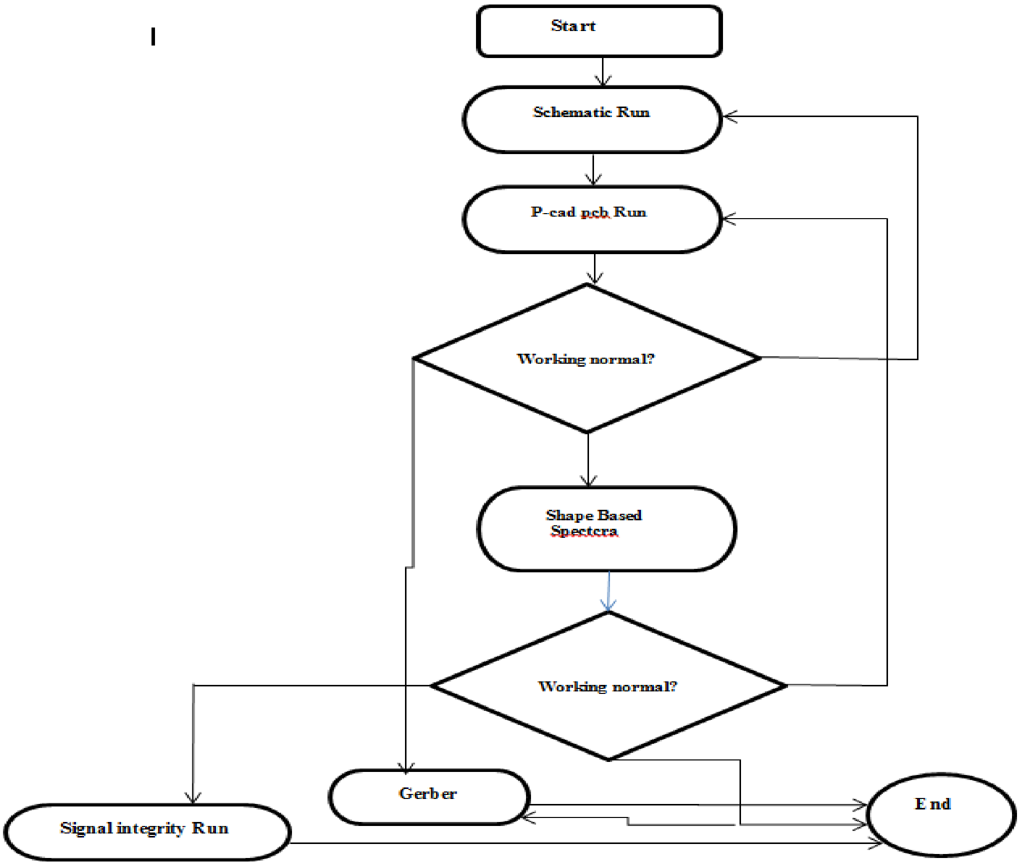


Fig. 4 Flowchart for Traditional P-CAD

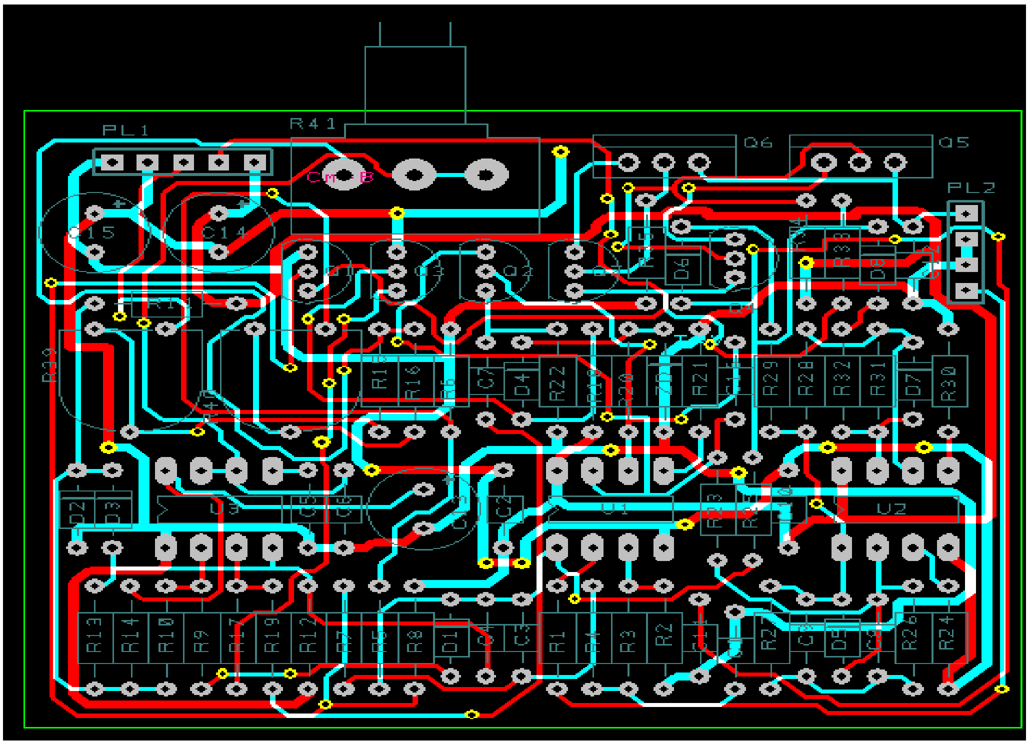


Fig. 5 File with the extension .pcb

DOI 10.17148/IARJSET.2018.561

III. ARCHITECTURE OF A WEB-BASED ELECTRONIC CIRCUIT P-CAD SYSTEM

To adapt the **P-CAD** MODULE system to online operation, it is necessary to develop a new distributed architecture of the **P-CAD**, differentiating functions between the client and the server sides to achieve optimal performance in the conditions of low Internet speed and limited web server resources [15]. For example, preprocessing and input of data sent to the server is reasonable to perform on the client side. This allows eliminating retransmissions of incorrectly drawn-up circuits. Graphical representation of the query results should also be performed on the client side, which significantly reduces the amount of data transmitted over the network [16]. Circuit simulation and access to circuit component parameter libraries is expedient to provide using the server resources. Note that Intranet is an excellent platform for processing information within the enterprise. The latest web browsers support any client-side systems [17].

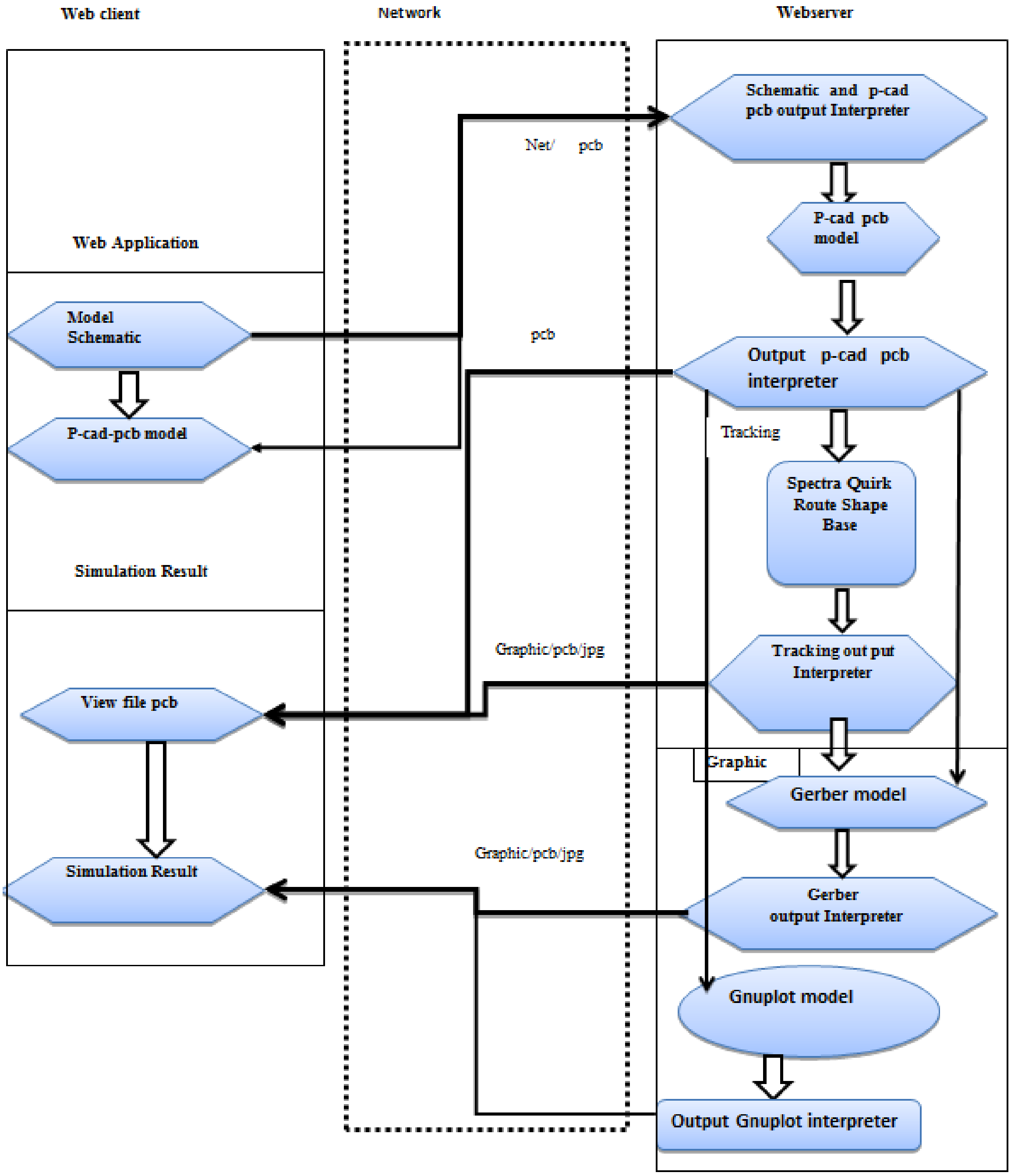


Fig.6 The architecture of a web-based P-CAD system

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

Currently, **modular web-based P-CAD** have begun to be developed. They are applications that consist of three blocks: the client-side, service, and design modules. The client-side module operates on the user's computer and generates data for their transfer to the server[18]. The service module launches parts of the client-side module via web pages. The design module running on the server processes the incoming data(Figure 7). Intranet is usually designed to be operated within a single compact or distributed enterprise and provides high security and performance. It is used to automate the workflow, provide information support for business processes, search and share data and documents of the company, and has gateways for connection to the Internet.

A. Flowchart for The architecture of a web-based electronic circuit P-CAD system

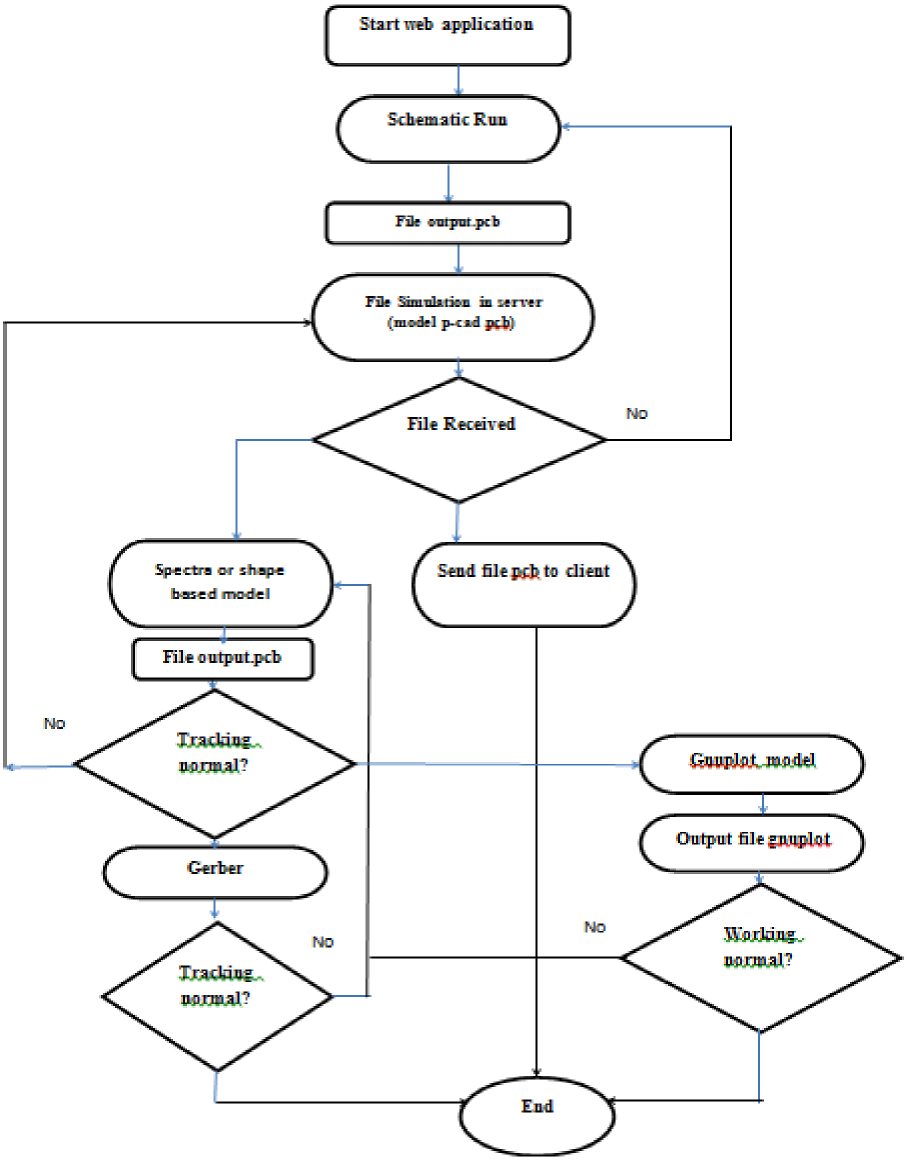


Fig 7 Flow chart for overall process.

IV. CONCLUSION

In this article, we have studied the evolution of the web-based solutions ‘architecture, from the simplest repositories of HTML pages to modern corporate solutions integrated with corporate information and partner information systems. At the same time, we have considered the problems that arise at each stage of development of web-based applications and technologies that solve such problems, including PHP, CGI, interaction with application servers and databases, creation and application of XML-based web services. We have shown the architecture of a specific **modular web-based P-CAD**.

REFERENCES

[1]. Gundavaram, S.,. *CGI Programming on the World Wide Web.* OReilly & Associates, Inc., 1996.
[2]. Jamsa K., Lalani S., Weakley S., Web Programming, Jamsa Press, Las Vegas, NV, 1996.
[3]. Wall,L., Christiansen, T., Schwartz, R.L. *Programming Perl,* OReilly & Associates, Inc., 1996
[4]. Flanagan D., JavaScript, The Definitive Guide, O'Reilly & Associates, Sebastopol, CA, 1997.
[5]. Hank Shiffman, Making Sense of Java, http://www.disordered.org/Java-QA.html
[6]. Hank Shiffman, Boosting Java Performance: Native Code and JIT Compilers,
[7]. http://www.disordered.org/Java-JIT.html
[8]. 2G6 User Guide, Department of Electrical and Computer Science, University of California, Berkeley, CA, 1981.
[9]. Quarles T., A. R. Newton, D. O. Pederson, and A. SangiovanniVincentelli, SPICE3 Version 3F5 User's
[10]. Manual, Department of Electrical and Computer Science, University of California, Berkeley, CA, 1994.
[11]. Thomas Williams & Colin Kelley// "gnuplot An Interactive Plotting Program "Version 4.2 organized by: Hans-Bernhard Br¨oker, Ethan A Merritt, and others .3 .March .2007. P.39-75.
[12]. Mall L., T. Christiansen, and R. L. Schwartz, Programming PERL, O'Reilly & Associates, 2nd edition 1996.
[13]. Object management Group, The Common Object Request Broker: Architecture and Specification, v,2.2.
[14]. published by Object Management Group, February 1998. see also http://www.cobra.org/
[15]. Wilamowski B. M. and Richard C. Jaeger, Computerized Circuit Analysis Using Spice Programs, McGraw-Hill, 1997.
[16]. David Kotz,//" LATEX and the Gnuplot Plotting Program":Principal author of this tutorial for gnuplot 3.0, July 3, 1991
[17]. Razevig, V.D., 1997. Micro-CAP V, the Circuit Simulation System. Moscow: SOLON, 273 p.
[18]. Recommended Practice for Architectural Description of Software-Intensive Systems, ANSI/IEEE Std 1471-2000.
[19]. Kuzmicz W., Nelayev V., Stempitsky V., Kudin K. // Proc. 9th Advanced Training Course on Mixed Design of Integrated Circuits and Systems. Education of Computer Aided Design of Modern ICs and Systems. Lodz, Poland, 2003. P. 665–668.
[20]. OrCAD PCB design solutions page: http://www.orcad.com/
[21]. Starzak L., Swiercz B., Zubert M., Napieralsky A. // "SPICE-Based simulation Website: Aplication to Teaching of Power
[22]. D.Alles and G.Vergottini, Taking a Look at Internet-based Design in the Year 2001, Electronic Design, Jan.6, 1997, P. 42-50.
[23]. Managed Achitecture of Existing Codeas a Practical Transition towards MDA, N. Mansurov, D. Campara, Klocwork, 1 Chrysalis Wat, Ottawa, Canada, K2G6P9, mansurov@klocwork.com.
[24]. Razevig, V.D., 1996. Design Center (PSpice), the Circuit Simulation and Printed Circuit Design System. Moscow: SK Press, 272 p
[25]. David Tansley,//"Using gnuplot to display data in your Web pages" IBM DEVELOPER WORKS System Administrator Ace Europe .19. January .2010. See also www.ibm.com/legal/copytrade.shtml.
[26]. Starzak L., Swiercz B., Zubert M., Napieralsky A. // "SPICE-Based simulation Website: Aplication to Teaching of Power Electronics", Proc. CADSM'2003. Lviv-Slavsko, Ukraine, 2003. P. 334–336.
[27]. Aleksander Malinowski, Bogdan M. Wilamowski,//" Web-based C++ Compiler "Bradley University, Peoria, IL / University of Wyoming, Laramie, WY Session 2532.
[28]. Nelayev V., Stempitsky V., Kudin K. // Proc. 8th EUROMICRO Conf. on Digital System Design. Porto, Portugal, 2005. P. 435–439.
[29]. Najbuk M., Nelayev V. // Proc. Int Conf. "e-learning jako metoda spomagająca proces kształcenia". Gdańsk, Poland, 2006. P. 72–76.
[30]. Kudrjavtsev P.A., Maximenya A.I., Nelayev V.V. // Proc. 2nd Global Congress on Engineering Education. Wismar, Germany, 2000, P. 144–147.
[31]. Electronics", Proc. CADSM'2003. Lviv-Slavsko, Ukraine, 2003. P. 334–336.
[32]. Insight: reverse engineer case tool, N. Rajala, D. Campara, N. Mansurov, IEEE Computer Society Press, 1999, Los Alamitos, CA, USA Pages: 630 – 633.
[33]. http://supportline.microfocus.com/Documentation/books/nx30books/piisapcn.htm (Date of viewing: 10.11.2009)
[34]. http://zone.ni.com/devzone/cda/epd/p/id/6363 (Date of view: 02.12.2009).
[35]. Nishanth Sastry (nishanth_sastry@us.ibm.com), Staff Software Engineer, IBM //Visualize your data with gnuplot .22 .Jul .2004, See also http://www.ibm.com/developerworks/library/l-gnuplot/.
[36]. http://www-tcad.stanford.edu.
[37]. http://www.php.net.
[38]. M.N. NAJBUK, V.V. NELAYEV//" Software-Hardware module GUI-SUPREM III for microelectronics technology design and training" Minsk, 220013, Belarusian 2007.P.136-144.
[39]. Sweet, W. and Geppert, L., "http:// It has changed everything, especially our engineering thinking," IEEE Spectrum, January 1997, P. 23-37.
[40]. http://www.gnuplot.info.
[41]. http://www.mediawiki.org.
[42]. http://www.wikipidia.org/.
[43]. https://github.com/OrenLederman/zigbee-dimmer/blob/ed438d57ddc088c26cce27c99b364b542b6ed1bc/eagle/zigbee_dimmer/zigbee-dimmer2.brd

BIOGRAPHY

Ayedh Abdulaziz Mohsen received his B.S. degree in engineering and technology from Saint Petersburg Electro technical University" LETI", Saint Petersburg, Russian Federation, in 2006, the M.S. degree in engineering and technology from Saint Petersburg Electro technical University "LETI", Saint Petersburg, Russian Federation, in 2008, and the Ph.D. degree in Technical Sciences from Saint Petersburg Electro technical University " LETI", Saint Petersburg, Russian Federation, in 2011. He is assistant professor in Department of Math's and Computer, Faculty of Science, Ibb University, Yemen and Department of information technology, Faculty of Computer and information technology, University of Science and technology "UST", Yemen. His research interests include Computer Aided Design (CAD), design web application, data mining through web.