International Journal of Computing and Business Research (IJCBR)
ISSN (Online) : 2229-6166
Volume 2 Issue 1 2011

AN APPROACH FOR TESTING THE DESIGN OF WEBSITE

Vandana Khatkar, MPhil Scholar
Chaudhary Devi Lal University
Sirsa, Haryana (INDIA)

ABSTRACT

In this paper, a new approach has been described that models a web-design and tests it for reachability between web pages. Testing of web-design is important because a fault in the design when allowed to travel further down the development process proves costlier as compared to testing the design for fault. The single most important operational feature of websites is the connectivity among its different components. A website has been called fully connected if each of the web page is reachable from all other web pages in the website. The tool described in this paper is of general nature as opposed to other automated testing tools/aids, which are of highly specific nature and need to be adjusted as soon as system specifications suffer even a minute change.

Key terms: Website, Reachability Matrix, Connectivity Between Pages.

I. INTRODUCTION

Web-based software systems (also referred to as website) are developed in a way much similar to their stand-alone counterparts (Wyncoop, 2000; Yourdon et al. 1978). This means:

- System Requirement Specification (SRS) in respect of websites is prepared and verified with active participation of user (Miller, 1979);
- Information to be presented to users of the website is gathered and verified;
- Feasibility aspects of website are looked into (Miller, 1979);
- Cost of website is estimated using an appropriate costing model (Miller, 1979);
- Web design is prepared and represented using a suitable tool;
- Web design is coded using a markup language and a general purpose distributed programming language (Ince, 2000), e.g., Java;
- Output of each major website development activity is tested so that faults are not carried deeper down the website development process (Biezer, 1999; Hetzel, 1984; Myers, 1979);

Several system development activities bear a precedence relationship among themselves with little or no concurrency. Testing activity, on the other hand, accompanies or follows almost every other software development activity (e.g., *User Requirement Analysis* yields an output document called *System Requirements Specifications*. The SRS is a highly technical document and becomes the foundation of further phases. A fault in SRS would be resulting in a faulty *System Design*. A faulty *System Design* when implemented can never turn into a correct system. This means, the implemented system would not be able to meet *System Specifications/User Requirements.* Moreover, it is rather expensive to go back and prepare SRS; diagnose, locate, and correct the fault in *System Specifications*; make necessary changes in the *System Design*; recode the changed design; and so on (Brooks, 1975).

A better approach is not to allow the software faults creep deeper into the development process. What does this tool do is to test the design of a website before it is coded and implemented using a markup language (Miller, 1979).

A website is an integrated collection of several web-based components that are either local or distributed over geographically dispersed computers. For successful operations of a website it is important that different components be accessible/reachable from anywhere in the website. This means that from within any web page of a website all the other web pages must be accessible. This, in turn, means that there must exist a path between all pairs of web pages in the website. A website is called “fully connected” if it satisfies the just mentioned reachability criteria. Therefore, it is quite reasonable that before website is implemented, its design be tested to establish the connectivity among web pages. This paper describes a modeling technique to test the design of website for reachability criteria (Wynkoop, 2000).

II. MODELING OF WEB-TESTING PROCESS

While designing a website the design team comes out with a detailed design of proposed software system. Software designs are represented using some design representation tool. A web design is not suitably represented using a conventional tool like a DFD or a Decision Table. The modeling technique described and used in this tool is independent of design representation tool as long as the web design can be transformed into a matrix form. Fig. 1 below depicts the modeling technique used in the tool described herein (Ince, 2000).

The test planner acquires web design developed by the web design/development team. The test planner then transforms web design into a directed graph form. In a directed graph, directed arrows represent hyperlinks between web pages and nodes represent web pages themselves. There exists a directed arrow from node i to node j, if there originates a hyperlink from i^{th} web page going to j^{th} web page.

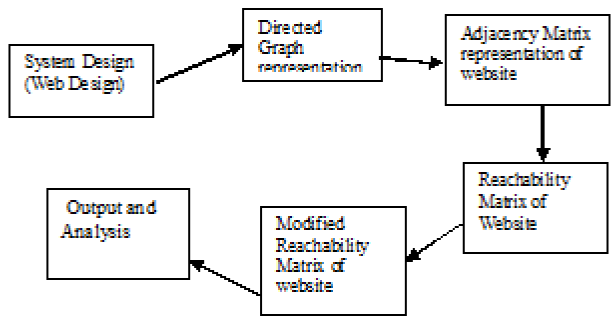

Fig. 1 Block diagram of modeling technique

The directed graph thus obtained is represented as a square matrix of size equal to number of nodes in the directed graph obtained above. The (*i, j*) element of matrix is a 1 if there exists a directed edge from i^{th} node to j^{th} node in the diagraph, i.e., if there is a hyperlink from i^{th} web page to j^{th} web page. Otherwise, if there is not a hyperlink from i^{th} web page to j^{th} web page, then there is a 0 in j^{th} column of i^{th} row of the matrix that represents the digraph.

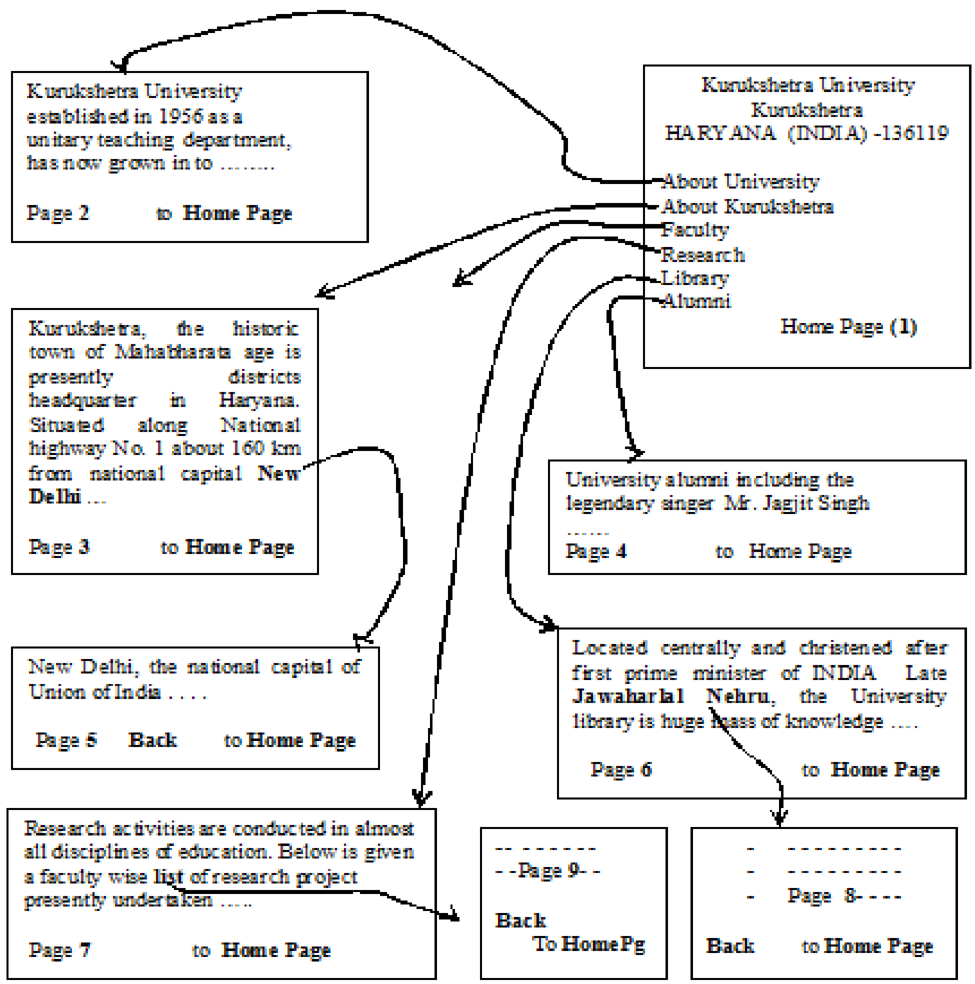

Fig. 2 A segment of hypothetical website of Kurukshetra University

The square matrix obtained in foregoing step is stored in a two dimensional array of size *n***n* where *n* is the number of web pages in the website. At this point web design has been transformed into an array form.

The testing tool then acts upon this tow dimensional array to test the web design for reachability among web pages. The output of the test tool is categorical in the sense that it answers in yes or no about connectivity of all possible pairs of web pages. If website is not strongly connected, then the output will categorically describe all the web pages from which a web page is not reachable.

Now, let's illustrate the way the modeling and testing technique described herein works by considering an example web design (Fig. 2). Web design developed by the design team is then represented using a directed graph (Fig. 3).

In the directed graph numbered nodes represent the respective web page in the original design of website. Thereafter, the directed graph is transformed into matrix form (Table 1). The matrix of Table 1 is called the *adjacency matrix*.

An algorithm processes adjacency matrix and generates another matrix hereby called the *reachability matrix* (Table 2).

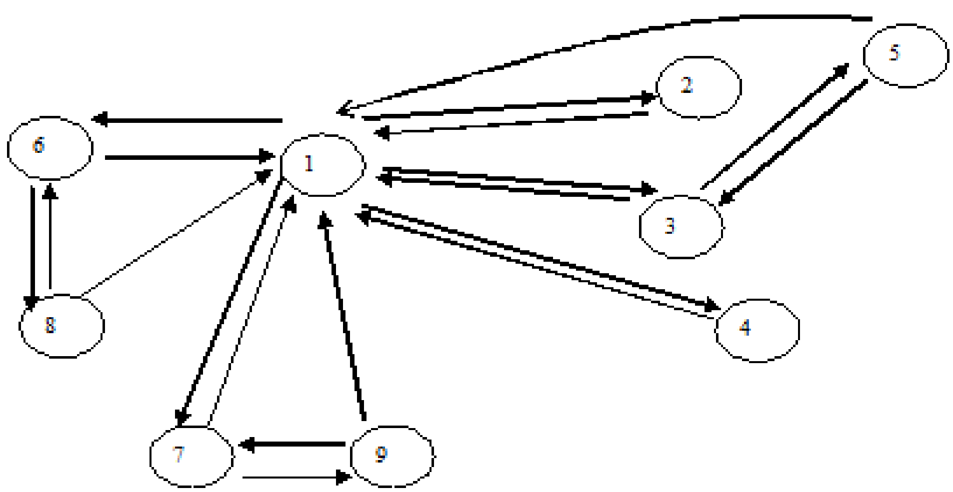

Fig. 3 Directed graph model for the website of Fig. 2

Node	1	2	3	4	5	6	7	8	9
1	0	1	1	1	0	1	1	0	0
2	1	0	0	0	0	0	0	0	0
3	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	0	0
5	1	0	1	0	0	0	0	0	0
6	1	0	0	0	0	0	0	1	0

7	1	0	0	0	0	0	0	0	1
8	1	0	0	0	0	1	0	0	0
9	1	0	0	0	0	0	1	0	0

Table 1 The adjacency matrix

The reachability matrix is transformed such that an entry (*i, j*) equals "1" if "*i*" equals "*j*", else it remains unchanged. The matrix thus obtained is called a *modified reachability matrix*. For our example website, the reachability matrix remains unchanged after abovementioned transformation.

Node	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1

Table 2 The reachability matrix

The modified reachability matrix is then analyzed to produce the final output of the testing tool. For a fully/strongly connected web design (as is our example web design) the reachability matrix has all entries as one's. Modified reachability matrix for web design of Figure 2 is given in Table 3.

Node	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1

Table 3 The modified reachability matrix

III. THE CONNECTIVITY TESTING ALGORITHM

Several algorithms are available for computing reachability matrix M^r for a directed graph. An entry at $(i, j)^{th}$ position in reachability matrix M^r is a "1" if node "*j*" is reachable from node "*i*" through a directed path of length "*p*" or less, where "*p*" is the total number of edges in the directed graph that corresponds to the total number of hyperlinks in the web design. Otherwise, $(i, j)^{th}$ entry is a "0". Therefore, it follows that the $(i, j)^{th}$ entry of the matrix M^r is a "1" if and only if the $(i, j)^{th}$ entry is a "1" in one of the matrices M^q for $1<=q<=n$ (Trembley et al., 1988). Thus we have

$M^r = M^1$ OR M^2 OR M^3 OR … OR M^n

Where $M^2 = M^1$ AND M^1

$M^3 = M^1$ AND M^1 AND M^1, and so on.

In this algorithm it is not needed to compute M^q $\forall$ $1 <= q <= n$. If at any stage M^j: $j <= n$, is all ones. Then M^q need not be computed $\forall$ $q > j$. Still this algorithm for computing reachability matrix is inefficient from temporal viewpoint, for its algorithmic complexity is of the order of $O(n^4)$. Algorithmic complexity of the order of $O(n^4)$ have severe implication on execution time if there are a large number of pages in the website. In the modeling technique described here an alternate inductive algorithm has been used that is much more efficient temporally. The algorithm used herein for computing reachability matrix of directed graph has complexity of the order of $O(n^2)$ (Deo, 1987).

Algorithm 1: *REACH_MAT*

Let *D* be the diagraph that represents the web design. And let *M* be "*n* x *n*" adjacency matrix that represents connectivity between all pairs of nodes in the diagraph, i.e., between all pairs of web pages in the web design. A sequence of matrices M^0, M^1, M^2, …, M^n is constructed inductively as follows (Deo, 1987):

Step 1. Let $M^0 = M$.

Step 2. Suppose matrix M^k has been computed for $0 <= k < n$. And let $M^k(i, j)$ be the entry at $(i, j)^{th}$ position in matrix M^k. Now, to find $M^{k+1}(i, j)$, i.e., entry at $(i, j)^{th}$ position in matrix M^{k+1}, following logical operations are applied.

$M^{k+1}(i, j) = M^k(i, j)$OR ($M^k(i, k+1)$ AND $M^k(k+1, j)$)

Step 3. M^n is the reachability matrix M^r of diagraph of *D*.

Below is described the website modeling and connectivity testing algorithm that 1) takes as input the adjacency matrix, 2) computes basic reachability matrix, 3) computes modified reachability matrix, 4) analyzes it, and 5) produces the analysis output.

Algorithm 2 *WEB_SIM*

Step 1. *Read n.*

[Input the total number of web pages “n” in the web design.]

Step 2. *Read M.*

[Input in an “n x n” two-dimensional array the adjacency matrix “M” of the diagraph that represents the web design to be tested for connectivity.]

Step 3. *Compute* M^r.

[Invoke algorithm ***REACH_MAT*** to compute M^n, the basic reachability matrix.

Step 4. *Transform* M^r *into modified reachability matrix* M^r_m

Step 5. *Analyze* M^r_m.

[Analyze the contents of modified reachability matrix M^r_m generated in Step 4. Objective of analysis is to look into interconnectivity of web pages. If $M^r_m(i, j)=0$ for some *i, j,* then there exists no path from i^{th} web page to j^{th} web page, direct or indirect.]

Step 6. *Produce the output.*

[Write onto an output device the matrix M^r_m (modified reachability matrix), and also the results of analyses.]

VI. DISCUSSION ON RESULTS AND CONCLUSION

Output of reachability algorithm is so simple and so much structured from numerical viewpoint that even a computer program can analyze it automatically. The singular feature of web design connectivity testing algorithm presented here is that it incorporates the analyzing program also and produces an output in narrative form as opposed to technical output, which necessitates manual analysis on the part of testing professional. Two distinct cases 1) strongly connected web design, and 2) not so strongly connected web design, are considered for output analysis.

Case 1

Consider the reachability matrix (Table 2 above) generated by the algorithm for web design of Figure 2. There are nine web pages in the design of web software. Therefore, adjacency matrix of the directed graph corresponding to web design will be a "9 x 9" square matrix. So will be the reachability matrix of directed graph. All the entries in reachability matrix are "1" each.

This signifies that the web design is strongly connected. That means – from within any location (web page) in the web software, the user can go anywhere (any web page) in the web software. The web system design is ready for implementation from connectivity/reachability viewpoint.

Case 2

Now consider a web design whose diagraph representation is given below in Figure 4. Table 4 describes the adjacency matrix of the diagraph. Correspondingly the reachability algorithm generates the basic reachability matrix of Table 5. And, this reachability matrix is then modified (all the diagonal elements are made "1" each) as in Table 6.

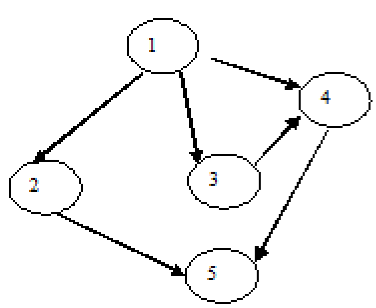

Fig. 4 Directed graph representation of web design

Node	1	2	3	4	5
1	0	1	1	1	0
2	0	0	0	0	1
3	0	0	0	1	0
4	0	0	0	0	1
5	0	0	0	0	0

Table 4 The adjacency matrix

Node	1	2	3	4	5
1	1	1	1	1	1
2	0	0	0	0	1
3	0	0	0	1	1
4	0	0	0	0	1
5	0	0	0	0	0

Table 5 The basic reachability matrix

Node	1	2	3	4	5
1	1	1	1	1	1
2	0	1	0	0	1
3	0	0	1	1	1
4	0	0	0	1	1
5	0	0	0	0	1

Table 6 The modified reachability matrix

In the web design of case 2 the test algorithm draws following inferences:

- Each web page is accessible from web page number 1.
- From web page number 2 three pages, numbered 1, 3, and 4, are inaccessible.
- Web page number 1 and 2 are inaccessible from page number 3.
- Web pages number 1, 2, and 3 are not reachable from page number 4.
- And finally, it is nowhere to go from web page 5.

REFERENCES

[1] Biezer, B. (1999), Software Testing Techniques, Second Edition, Van Nostrand Reinhold, New York.

[2] Hetzel, W. (1984), The Complete Guide to Software Testing, QED Information Sciences.

[3] Miller, E. (1979), Automated Tools for Software Engineering, IEEE Computer Society Press.

[4] Wynkoop, E. (2000), Developing and Running a Perfect Web Site, Second Edition, Prentice Hall International, Englewood Cliffs – New Jersey.

[5] Brooks, F.P. (1975), The Mythical Man Month: Essays on Software Engineering, Second Edition, Addison-Wesley, Reading, Massachusetts.

[6] Comer, E. (2000), The Internet Book: Everything You Need to Know about Computer Networking and How the INTERNET Works (2e), Prentice Hall.

[7] Trembley, J.P. and Sorenson, P.G. (1988), An Introduction to Data Structures with Applications (2e), Tata McGraw Hill.

[8] Yourdon E.N., and Constantine, L.L. (1978), Structured Design, Yourdon Press.

[9] Deo, N.S. (1987), Graph Theory (2e), Prentice Hall India Pvt. Ltd..

[10] Ince, D. (2000), Programming the Internet with Java, Addison Wesley Longman.

[11] Myers, G.J. (1979), The Art of Software Testing, John Wiley International.