A Method to Support Web Evolution by Modeling Static Structure and Dynamic Behavior

Junhua WU
College of Computer Science and Engineering,
Southeast University, Nanjing, China
College of Information Science and Engineering,
Nanjing University of Technology, Nanjing, China
wujh@njut.edu.cn

Baowen XU
College of Computer Science and Engineering,
Southeast University, Nanjing, China
bwxu@seu.edu.cn

Abstract — Software evolution is an important activity in software life cycle. The Web application is very different from traditional software in design and composition. It needs maintained more frequently. So the automatic evolution support is a critical part in Web engineering. This paper uses Markov chain to model Web users’ browser behavior by statistic analysis of Web log. And transition probability is used as weight in model. By it a model of browse centrality is built. Contrasting it with static model of structure centrality we propose an algorithm based on rules about automatic evolution of Web link structure.

Keywords—Software evolution; Web metrics; Web evolution

I. INTRODUCTION

With the development of Web and Internet, using Web site for business becomes one of the best choice. Because of more Web technology and user’s demands Web applications are more and more complex and huge in their content, function and structure. It has been evolved from pure information service system to those with business transaction, work flow, collaborating work environment, online business and portal[1]. Numbers of Web applications has faced to a lot of problems in their development, deployment and management. Moreover Web applications are different from traditional software. So, as a new discipline Web engineering was proposed on 1990s’.

Web applications are built usually in a variable environment with unstable demands. Compressed process, persistent evolution, rapid changeable environment, variable developer frequently, demands of high performance, mixed multi-discipline (graphic design, information structure, hyperlink etc.) make Web application different from traditional software[2]. Therefore maintaining Web application is a complex activity. Cutter Consortium’s report shows that some large Web applications present serious problems[2]: 53% system delivered doesn’t meet functional requirements, and 52% system is absent of quality. Nathan Heinze, Qing Hu’s research[3] illustrates that Web sites owned by standard & Poor’s 500 corporations has experienced obvious variety on their information presentation, interactive properties and services provided. These means it is necessary that efficiency of maintenance and evolution is improved.

Software evolution is an important activity in software life cycle. It aims at analyzing current state of software system and predicting future software activities. Meir M. Lehman and Juan F. Ramil propose a model of software evolution[4]. Its essential is by analysis in theory and continuous data collecting to define, generalize formal expression. Finally evolving rules and guidelines are made. Software evolution can be divided into two level, one is model and another is program[5]. Model evolution is improvement to analysis or design model and program evolution focuses on code optimization.

For a business Web site, insufficient design and usability will lead to lose some business. Evolving to Web is concerned with code restructuring, information restructuring and link restructuring etc. Generally, design of link structure bases designer’s experience. So improving design is important. Currently many researches aim at navigation after response to user’s request, such as link analysis[6,7], page rank based on hits[8] etc. These focus on dynamic reconstruction hyperlink on run-time when Web site response some request. We don’t think it is a kind of evolution of Web. This paper pays attention to reconstruct hyperlink when the Web site exists some hyperlink structure that can be optimization, and this hyperlink structure is decided in designing. So it is a kind of reconstruction to design model. Up to now, there have been many works on modeling and measuring to hyperlink structure. They can be considered as basis of design evolution[9-12]. Basis on which, this paper builds the model to measure the link structure by analyzing the Web log. And then some rules are defined to evolve the hyperlink structure.

II. LINK CENTRALITY OF WEB

Web application consists of static and dynamic pages by hyperlink. So Web hyperlink structure can be modeled as a link graph $W=(V, E)$. $V=\{p_i\}$ is a set of page $p_i$, and $E=\{e_i\}$ is a set of edge $e_i$ from page $p_i$ to $p_j$. 

$V \cup E$ is a set of vertex and edge which are graph's elements. 

$W$ is a union of $V$ and $E$. 

The Web evolution is a change in $W$ which includes addition and deletion of page and hyperlink. The change can be defined by transition from $W_t$ to $W_{t+1}$ where $W_t$ is a state of Web at time $t$. The transition is modeled by a Markov chain $M=(V, E)$ where $V$ is a set of states and $E$ is a set of edges.

The Markov chain is a stochastic process that describes a sequence of possible events where the probability of each event depends only on the state attained in the previous event. In the context of Web evolution, the states represent the possible configurations of the hyperlink structure, and the transitions between states are determined by the probability of a page being accessed.

The model is defined by a transition matrix $P$ where $P_{ij}$ is the probability of moving from state $i$ to state $j$. The transition probability is calculated based on the frequency of occurrence of page transitions in the Web log.

The centrality of a page is defined as the probability of a page being accessed after a certain number of transitions. This can be calculated as the eigenvector of the transition matrix $P$, corresponding to the largest eigenvalue.

The model is validated by analyzing the Web log of a real Web site and comparing the predicted link structure with the actual structure. The results show that the model is able to accurately predict the evolution of the hyperlink structure.

VIII. CONCLUSION

The paper proposes a model to support Web evolution by modeling static structure and dynamic behavior. The model is based on the transition probability of Web users’ browsing behavior, which is extracted from the Web log. The centrality of a page is defined as the probability of the page being accessed after a certain number of transitions, which is calculated as the eigenvector of the transition matrix.

The model is validated by analyzing the Web log of a real Web site, and the results show that the model is able to accurately predict the evolution of the hyperlink structure. The model can be used as a basis for the design and evolution of Web applications.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (61073154, 61272302), Natural Science Foundation of Jiangsu Province (BK2011566, BK2012578), and 863 Program of China (2012AA011205).

REFERENCES


© 2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Authorized licensed use limited to: TOKYO INSTITUTE OF TECHNOLOGY. Downloaded on November 9, 2009 at 06:34 from IEEE Xplore. Restrictions apply.
Link graph presents the organization of Web system. It can reflect readability of Web pages and easiness of browsing Web. For a large Web system illogical link structure will affect usability and efficiency.

A Web with \( n \) nodes can be represented as a distance matrix \( C_{[N,N]} \). \( C_{ij} \) called distance is the number of link. If there isn’t a path from \( i \) to \( j \) a big number is used for representing \([11]\).

According to link graph \( W \) and matrix \( C_{[N,N]} \) a metric called node centrality is defined as follow\([11]\):

\[
Od = \sum_{j} C_{ij} \quad (1)
\]

\[
Id = \sum_{j} C_{ji} \quad (2)
\]

Node centrality illustrates correlation between node \( i \) and other nodes. Less \( Od \) or \( Id \), nearer node \( i \) from other nodes. Normalizing \( 1 \) and \( 2 \) will result in formula \( 3 \) and \( 4 \):

\[
ROC = \frac{\sum_{j} C_{ij}}{\sum_{j} C_{j}} \quad (3)
\]

\[
RIC = \frac{\sum_{j} C_{ji}}{\sum_{j} C_{j}} \quad (4)
\]

\( ROC \) and \( RIC \) are considered as relative centrality.

The value is bigger, node is more central. The node with the highest relative centrality can be as root node\( [11] \) of hierarchy hyperlink . If a node is specified as root a hyperlink model can be transformed into a link tree. The root of tree is the node which possesses the highest \( ROC \). That is:

\[
\text{root} = i \quad \text{While} \quad ROC = \max \{ ROC|j=1,2,\cdots, n \}
\]

Definition 2.2: In link model average path length \( D_{ij} \) between two nodes is defined as:

\[
D_{ij} = \frac{\sum_{j} C_{ij}}{N \times N} \quad (5)
\]

After a Web site is developed \( ROC \), \( RIC \) and \( D_{ij} \) of any node can be computed out. If a node has a high centrality it’s browse path will be shorter. Average path length \( D_{ij} \) presents average length of link path of Web pages. These two metrics can be used in measure rationality and usability of link structure.

Usually a homepage should be a node with the highest centrality, as a root. While node \( k \) is specified as a root a spanning tree \( T(k) \) can be generated by use corresponding algorithm and the shortest paths are also resulted.

If distance between two nodes linked directly is considered as same \( ROC \), \( RIC \) and \( D_{ij} \) of node is decided by static link structure designed. Generally it is limited by designer’s experience. But we think quality of link structure can not be evaluated only by it. For example, some nodes with low centrality is hit frequently but encounters longer path, some link paths of pages are accessed far more than other reachable paths. Therefore browsers’ interest and real navigation path can be used to measure rationality of design indirectly. We think it would be a valid method for evolving Web by analyzing users’ browsing behavior.

III. METRICS TO LINK STRUCTURE WITH WEIGHT

Generally Web server records all requests from user, so information about access behavior can be caught by Web log. While responding a user’s hit request Web will generate a record, by which some analysis can be made. This paper will build a model for browsing behavior by statistic data from log. Extended log include some record fields\([13]\). See table 1.

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>remotehost</td>
<td>Remote host or IP</td>
</tr>
<tr>
<td>referrer</td>
<td>User’s login name</td>
</tr>
<tr>
<td>authuser</td>
<td>Authorized user by server</td>
</tr>
<tr>
<td>authuser</td>
<td>Authorized user by server</td>
</tr>
<tr>
<td>date</td>
<td>Date and time of responding to request</td>
</tr>
<tr>
<td>request</td>
<td>Request method(Gel!Head!Post)</td>
</tr>
<tr>
<td>status</td>
<td>HTTP status code</td>
</tr>
<tr>
<td>bytes</td>
<td>Transmission bytes</td>
</tr>
<tr>
<td>referer</td>
<td>The page will be accessed by this hit</td>
</tr>
<tr>
<td>gent</td>
<td>user’s OS or navigator</td>
</tr>
<tr>
<td>time2taken</td>
<td>Time will be taken for the transaction</td>
</tr>
</tbody>
</table>

Table 1. Record fields of Web log

Hit stream data, such as source and destination page, time of user accessing page etc, can be caught from log record. This record is simplified as \( L=(S,D,T) \) for our purpose, \( S \) is the source node of link and \( D \) is the destination, \( T \) is the time of accessing node. Transition probability from \( S \) to \( D \) can be computed by statistic of \( S \) and \( D \). Average accessing time can be computed by statistic of \( T \).

A. Transition model of link

Transition model of link is built by actual link path and hit frequency. Firstly we have some assumption:

- Any request will be responded in specified time.
- It is still considered as a valid hit although source and destination node is same one.
- Page jump produced by backup or forward is ignored.

A link behavior from page \( S \) to \( D \) is thought as a transition. Frequency of transition is labeled in edge \( e_{ij} \) of \( W=(V,E) \) as weight. If that browsing Web is considered as a random process we have following definition:

Definition 3.1 For a random process \( W_{T} = \{ W, t \in T = \{ 0,1,2,\cdots \} \} \), state space \( I = \{ 0,1,2,\cdots \} \) is the set of pages. Then \( 1 \) \( W_{T} \) is a Markov chain. \( 2 \) Condition probability \( P(W_{n+k} = j | W_{n} = i) \) is the probability of state being \( j \) at time \( n+k \) while at time \( n \) state is \( i \). That is transition probability from state \( i \) to state \( j \) through \( k \) steps.

Definition 3.2: If sum of frequency from \( i \) to all nodes
link directly is \( O_{num} \) then transition probability \( P_{ij} \) from \( i \) to \( j \) is \( e_{ij}/O_{num} \).

Accessed probability of specified page and average count of transfer to a page can be calculated by Markov model of Web.

Definition 3.3: \( \pi_j \) is the probability that user browsers page \( j \), defined as [9]:

\[
\pi_j = \sum_i \pi_i P_{ij}
\]

Here \( P_{ij} \) is transition probability from \( i \) to \( j \).

Definition 3.4: \( m_{ij} \) is average count of transfer from page \( i \) to \( j \), defined as [9]:

\[
m_{ij} = 1 + \sum_k m_{ik} P_{jk}
\]

So:

\[
m_{ij} = 1/\pi_j
\]

\( \pi_j \) represents users’ interest to page \( j \) and \( m_{ij} \) may reflect average path from page \( i \) to \( j \) indirectly. But sometimes page \( j \) is the necessary middle node to a hot page \( k \) while users’ destination is \( k \). So we propose another metric to illustrate users’ interest. That is average browse time \( A_i \). \( A_i \) can be computed by field time2taken of Web log. If \( T \) is the time of accessing node \( i \) then

\[
A_i = \sum(T)/\text{Count}(i)
\]

B. Measuring structure with weight of users’ behavior

Formula (1)-(4) have defined the centrality of static structure node. But it only presents the central property determined by designer. But actual central node of access should be observed by users’ browse behavior. Perhaps they are match or inconsistent. Formula (6)-(8) can figure out users’ browse trend by transition probability. If adding transition probability into \( W'=(V',E') \) we will obtain \( W''=(V'',E'') \) with weight edges. And then browse centrality \( BOD_i \) and \( BId_i \) are suggested:

\[
BOD_i = \sum_j 1/P_{ij}C_{ij}
\]

\[
BId_i = \sum_j 1/P_{ij}C_{ij}
\]

Here it is defined as distance from \( i \) to \( j \) with weight that distance \( C_{ij} \) multiple by \( 1/P_{ij} \). Less its value, shorter distance. Normalizing (9) and (10):

\[
RBOC_i = (\sum_i 1/P_{ij}C_{ij}) / (\sum_i 1/P_{ij}C_{ij})
\]

\[
RBIC_i = (\sum_i 1/P_{ij}C_{ij}) / (\sum_i 1/P_{ij}C_{ij})
\]

After normalization, higher \( RBOC_i \) and \( RBIC_i \) mean more interesting and nearer to browse central node. Spanning tree of \( W' \) is called \( T'(k') \). For each layer of \( T'(k') \) \( RBIC'_{lm} \) is an average browse centrality with average value of \( RBOC_i \) in layer \( m \).

IV. EVOLVING WEB STRUCTURE BY RULES

A. Algorithm principle and rules

\( T \) and \( T' \) are corresponding to two spanning tree with static and dynamic property. For the evolution to a Web running properties plays an important role. If \( T \) and \( T' \) is the same it is obvious that structure designed is meet with users’ behavioral structure. Or else there is some difference between them, so some improvement is needed.

Element value of distance matrix \( C[N,N] \) and \( C'[N,N] \) presents edge property of \( W \) and \( W' \). If \( i,j \in N \) and \( C_{ij}=1 \) then \( i \) is a direct preceding node of \( j \), and \( j \) is a direct succeeding node of \( i \). \( \text{pre}(j) \) presents all direct preceding nodes of \( j \) and \( \text{suc}(i) \) presents all direct succeeding nodes of \( i \).

After analyzing these models we think it is possible that link structure can be improved if subjective opinions of Web owner are ignored. Some points are following.

- If node \( i \) presents high browse centrality \( RBOC_i \) and \( RBIC_i \) it should be a high layer node. Node with the highest browse centrality should be root node. This will short average path length.
- For node \( ij \) if \( P_{ij} \) approximates to \( 0 \) then \( e_{ij} \) can be deleted. Because it means very seldom actual access from \( i \) to \( j \).
- Node \( i \) should be removed if \( \pi_i \) approximates to 0 or average browse time \( A_i \) is far less than that of all nodes. Because it means users aren’t any interesting in it.
- If \( i \) and \( j \) is connected directly, and \( P_{ij} \) approximates to \( 1 \) then make \( j \) merging into \( i \). It means that probability from \( i \) to other nodes is approximates to \( 0 \) and users browse \( i \) node almost visit \( j \) too. \( i \) and \( j \) exist high relativity. So they can be combined.

So some rules are defined to support evolution of Web structure. Automatic evolution algorithm to Web is a process to optimize Web graph \( W \) based on these rules and models discussed.

Rule 1: \( \forall j, j \in T'(k') \) \( i \), \( \exists \text{pre}(j) = i \) and \( P_{ij} < e \) \( \rightarrow W=W-e_{ij} \)

Rule 2: \( \exists i, i \in T'(k') \) and \( \pi_i < e \), \( \rightarrow \)

\( \forall j, j = \text{pre}(i) \land j = \text{suc}(j), T'(k') = T'(k') - i \land W=W-i, W=W-e_{ij} \lor W=W-e_{ji} \)

Rule 3: \( \exists i, i \in T'(k') \) and \( A_i < e \), \( \rightarrow \)

\( W=W-i \land W=W-e_{ij} \lor W=W-e_{ji} \land \exists m \in \text{suc}(j) \)

Rule 4: \( \forall i, i \in T'(k') \), \( \exists j, j = \text{suc}(i) \land P_{ij} > 1-e \), \( \rightarrow \)

\( W=W-i, T'(k') = T'(k') - i, \forall k, k = \text{pre}(i), W=W-e_{ik} \lor W=W-e_{ki} \)
\[ \forall m, m = \text{suc}(i), W = W_{em} \]

Rule 5: While \( i, m \in T'(k'), k = \text{Depth}(m) \), If \( \exists RBOC_i < RBOCL_{k-1} \) and \( RBOC_i > RBOCL_{k+1} \) \rightarrow \( \text{Depth}(i) = k \) and \( \text{pre}(i) = j, P_j = \max \{ P_{mi}, \text{Depth}(m) = k - i \} \)

B. Case analyzing

In order to illustrate the paper’s idea a simple case is analyzed by this method. Figure 1 is an example of hyperlink graph only with six nodes and its spanning tree. Table 2 lists distance matrix and centrality ROC, RIC computed out. Users’ access behavior hasn’t been taken on while computing centrality.

Figure 1 shows that Web is layer structure. According to its spanning tree the average path length is 5/7 from root to other nodes. Centrality values from Table 2 indicates that out center of this Web is node \( a, f \) with the highest ROC and in center is node \( e \) with the highest RIC.

Table 2. Distance matrix and centrality
(Number 7 means no path between two nodes)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>OD</th>
<th>ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>16.43</td>
</tr>
<tr>
<td>b</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>22</td>
<td>5.23</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>12.78</td>
</tr>
<tr>
<td>d</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>35</td>
<td>3.29</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>35</td>
<td>3.29</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>16.43</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>24</td>
<td>20</td>
<td>24</td>
<td>18</td>
<td>12</td>
<td>17</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>RIC</td>
<td>4.79</td>
<td>5.75</td>
<td>4.79</td>
<td>6.39</td>
<td>9.58</td>
<td>6.76</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Web graph from Figure 2 is labeled with transition probability as weight. Its spanning tree \( T' \) is different with \( T \) shown in figure 1. Table 3 only lists RBOCB and ignores the RBIC as simplicity.

Table 3. Distance matrix and centrality value with weight
(Number 1000 means no path between two nodes)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>BOD</th>
<th>RBOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>100</td>
<td>1.27</td>
<td>5</td>
<td>2.44</td>
<td>2.63</td>
<td>111.34</td>
<td>125.86</td>
</tr>
<tr>
<td>b</td>
<td>1000</td>
<td>0</td>
<td>1000</td>
<td>1000</td>
<td>1</td>
<td>0</td>
<td>3001</td>
<td>4.67</td>
</tr>
<tr>
<td>c</td>
<td>5.88</td>
<td>500</td>
<td>0</td>
<td>71.43</td>
<td>1.92</td>
<td>2.08</td>
<td>581.32</td>
<td>24.1</td>
</tr>
<tr>
<td>d</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
<td>1000</td>
<td>1000</td>
<td>5000</td>
<td>2.8</td>
</tr>
<tr>
<td>e</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
<td>1000</td>
<td>1000</td>
<td>5000</td>
<td>2.8</td>
</tr>
<tr>
<td>f</td>
<td>2.78</td>
<td>277.78</td>
<td>3.52</td>
<td>33.33</td>
<td>1.64</td>
<td>0</td>
<td>319.05</td>
<td>43.92</td>
</tr>
<tr>
<td>BID</td>
<td>3008.66</td>
<td>2877.78</td>
<td>3004.79</td>
<td>2109.76</td>
<td>1007</td>
<td>2004.72</td>
<td>14012.7</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. Web hyperlink graph with weight and its spanning tree \( T' \)

Table 2. Distance matrix and centrality
(Number 7 means no path between two nodes)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>OD</th>
<th>ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>16.43</td>
</tr>
<tr>
<td>b</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>22</td>
<td>5.23</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>12.78</td>
</tr>
<tr>
<td>d</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>35</td>
<td>3.29</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>35</td>
<td>3.29</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>16.43</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>24</td>
<td>20</td>
<td>24</td>
<td>18</td>
<td>12</td>
<td>17</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>RIC</td>
<td>4.79</td>
<td>5.75</td>
<td>4.79</td>
<td>6.39</td>
<td>9.58</td>
<td>6.76</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figures 1. Web hyperlink graph and spanning tree \( T \)

Figure 2. Web hyperlink graph with weight and its spanning tree \( T' \)
Browse centrality values RBOC from Table 3 indicates node $a$ is the out center and node $f$ is the secondary central node. After using rules to evolve the Web we will get a link structure optimized with two layers and average path is 1 shown in figure 3. Contrast Figure 3 with Figure 1 it is obvious that number of structure layer and average path length is decrease.

Actually while evolving Web structure opinion of Web’s owner is also important because topic and content of page is decided by them. So evolution tool only aids to improving design. It should be applied properly.

V. CONCLUSION

The quality of Web link structure usually depends on designer’s experiment. But metrics to actual access of Web can be basis for Web evolution. This paper suggests a method to evolution that is based on analyzing and measuring Web’s static structure and dynamic access behavior. By using markov transition model we construct the link centrality model with weight. According to the model some evolution rules are defined so as to implementing automatic evolution. Currently we are working to realizing the tools using this method. But Web evolution is a complex problem. Metrics in the model are only extracted from Web log with some limits. More properties will be taken into account. It will be our objective in the future.

REFERENCE