

## 페이지 접근의 대중성에 따른 웹사이트 재구성에 관한 연구

조석필\*

### 요 약

웹 사이트를 방문하는 인터넷 사용자들의 빈도에 따라 웹사이트의 성능 과 그 품질이 평가되곤한다. 본 논문에서는 각 웹페이지의 관계 대중성을 측정하여 그 값을 계산함으로써, 취약한 웹페이지를 자동으로 재구성 할 수 있는 방법과 개선방법을 제시한다. 여기서 관계 대중성에 관한 계수는 웹사이트를 방문하는 사용자가 얼마나 빠르고 쉽게 요청한 자료를 얻을 수 있는지와 링크 구성의 트리의 깊이에 대한 접근성에 따라서 좌우된다.

### 1. 서론

HTTP로그 파일의 기록사항들은 요청 시간 및 날짜, 그리고 요청한 파일 및 기타 정보들로 구성하며 클라이언트가 이러한 요소들을 서버에 요청한다[1][6]. 이러한 정보의 소스는 웹사이트 관찰의 기본으로 설정하며 시험관찰의 핵심으로 하는 페이지 대중성이 본 논문의 주제이다. 여기서는 페이지 대중성을 정의하는 방법을 시험하고 그리고 웹사이트를 재배열하여 페이지의 대중성을 사용하여 보다 실질적으로 용이하게 접근할 수 있고 보다 효과적으로 하이퍼텍스트를 구성하는 방법을 보여준다. 유사한 특징에서, 골로빈스키[Golovchinsky]는 브라우징하는동안 수집된 사용자들로부터 피드백에 근거하여 링크 배열을 자동으로 탐색한다[2][7].

기록 파일 분석의 개념은 새로운 것이 아니다 그러나 여기서의 접근은 웹사이트의 페이지를 자동으로 재배열하기위해 관계있는 페이지 대중성의 분석을 사용한다. 본 논문에서는

이러한 기능을 수행하는 시험도구를 개발하였고 그리고 네베게이션상에 개선된 결과는 다른 웹사이트에있는 페이지 접근에 적용하는 것을 보여준다.

### II. 페이지 대중성의 정의

페이지의 대중성을 평가하는 하나의 용이한 방법은 지정된 기록파일의 페이지에 접근한 것을 근거로하여 접근된 횟수를 계산한다.(기록파일에 작성된 모든 http 접속 성분에 있어서의 참조: <http://www.w3.org/Daemon/User/Config/Logging.html>). 그러나 기록 파일로 절대 접근 횟수를 계산하는 것은 오류를 발생할 수도 있다. 홈페이지(서버의 초기 페이지)에 접근하는 페이지는 보다 절대 접근성이 강하다. 그이유는 홈페이지와 대상 페이지사이의 경로에 의존하기 때문이다. 대상페이지는 HTML트리구조에 보다 깊숙히 위치하고 있으며 홈페이지는 최상위에 위치한 루트이고 다른 페이지에 연결되는 하이

\* 성결대학교 컴퓨터학부 교수

퍼링크는 부-자 관계를 배열하는 계층구조이다. 여기서 하위 계층의 자 페이지는 차 상위 페이지 중의 하나에 연결되기 때문에 트리구조라는 용어는 적절한 표현은 아니다. 페이지의 대중성을 보다 잘 측정하기 위해서는 다음의 사항을 고려할 수 있다.[3]

- 페이지의 깊이(홈페이지로부터 몇 단계인지를 나타낸다.) :  $d$
- 측정되어지는 페이지로서 동일한 깊이에 있는 페이지 수 :  $n$
- 서버의 다른 페이지로부터 특정한 페이지에 연결되는 하이퍼링크의 수:  $r$

여기 계수  $a$ 가 이러한 모든 매개변수를 포함한다고 가정할 경우, 새로운 관계접근 RA(relative accesses)에 대한 수식은 다음과 같이 표현할 수 있다.

$$RA = a * AA \text{-----}(1)$$

여기 AA는 절대접근을 그리고 RA는 관계접근을 의미한다.

### 계수 $a$ 의 정의(4)(5)

페이지 깊이  $d$ 는 페이지의 대중성으로부터 멀어지기 때문에, 계수  $a$ 가  $d$ 에 비례한다고 가정할 수 있다. 게다가, 동일한 깊이  $n_d$ 지점에 페이지의 수는  $a$ 에 비례한다. 그 이유는 선택의 수가 크면 클수록, 특정한 페이지 선택의 중요성이 보다 크다. 다른 페이지  $r$ 에 대한 참조는 일반적으로 페이지의 대중성을 높인다. 그리고  $r$ 는 계수  $a$ 에 역비례한다.

최근 웹서버는 그들의 페이지중 어느것에 연

결하고 "log."라고 불리워지는 주소에 기록된 웹 페이지를 추적하도록 배열하였다. 대부분의 웹 브라우저는 이러한 기능을 지원하며 브라우저의 HTTP요청과 동시에 서버페이지에 링크(참조)를 송부한다.

$r$ 를 서버 페이지에서 동일 서버 페이지로 연결하는 수를 기초하여 평가하면,  $a_i, d_i, n_i$ , 및  $r_i$  사이의 관계 즉 페이지  $i$ 에 대한  $a, d, n$  및  $r$ 의 값은 다음과 같이 나타낼 수 있다.

$$a_i = F(d_i, n_i, 1/r_i), = 1 \dots K, \text{---}(2)$$

여기서  $K$ 는 웹 페이지의 수이며  $F$ 는 사용자의 행위를 반영하여 정의하는 특정한 기능이다. 즉, 사용자의 행위는 특정한 웹사이트를 찾아다니는 동안 사용자가 방문하는 페이지, 순서, 사용기간 등이다.

### 링크 편집 도구

웹서버 관리자가 설정한 범위에 따라 페이지 사이에 링크를 재배열하는 측정용 도구를 사용하였다. 다음의 측정에서, 주어진 페이지의 관계적 대중성이 적어도 하나 이상의 상위 부 페이지(홈페이지에서 보다 짧은 페이지)의 대중성을 초과할 때 링크편집 도구는 링크를 재배열한다. 측정용 도구는 주기적(예를 들면 매주단위로)으로 관계 페이지 대중성을 접근할 수 있다. 계산 단위는 다음과 같다.

AA---페이지 당 절대접근의 수

RA---페이지 당 관계접근의 수

PT---페이지 시간을 의미하며, 사용자가 특정 페이지에서 걸리는 시간

UT---사용자 시간을 의미하며, 사용자가 매 세션 마다 서버에서 보내는 시간  
 NP---페이지의 수를 의미하며, 사용자가 매 세션을 방문시 얼마나 다른 페이지를 방문하는지의 의미

“측정용 도구는 웹서버관리자가 설정한 범주에 따라서 페이지 사이의 링크를 재배열한다.”

본 분석에서 새로운 세션은 사용자가 우선 웹 사이트를 접근할 때 시작된다. 모든 성공적인 접근은 세션 간의 간격이 주어진 임계시간을 넘지 않는다면 동일한 세션에 속한다[6]. 링크 편집 알고리즘에 대한 단순 개정은 어떤 페이지가 그 상위 부 페이지 보다 높은 관계접근(RA)값을 가지면 그 페이지는 부 페이지로 교환한다. 매 페이지가 보다 높은 관계접근(RA)값을 가진 부 페이지를 가질 때 종전의 단계를 반복한다.

### 사례연구 정의(7)

웹서버의 취약한 구조가 인터넷 사용자를 어떠한 형태로 위축시킬 수 있는 것인지를 보여주는 것과 관계접근에 기초한 링크편집이 이 문제를 어떻게 해결할 수 있는지에 대해 보여주는 계수  $a_i$ 에 대해 본 알고리즘을 적용하였다.

측정 결과가 가능한 한 정확하다는 것을 확인하기 위해, 본 사례 연구에서는 웹서버에 접근하기 위해서 확실한 사용자의 이름과 패스워드를 가진 인증 된 사용자에 한해서만 허용하였다. 사용자의 인증은 링크 편집 알고리즘상 그렇게 불편한 것은 아니다. 즉 평가목적에서만 본 요구를 적용하였다. 더욱이, 외부서버로부터 본 사이트에 참조링크가 없었으며, 매개변수  $r_i$

의 값이 정확하였다. 결과에 대한 설명을 용이하게 설명하기 위해서 페이지와 마찬가지로 사용자의 그룹을 조심스럽게 선택하였다.

보다 검증을 쉽게 하기 위해 계수를 정하는 가능한 방법은 다음과 같다.

$$a_i = c_1 * d_i + c_2 * n_i/r_i - a_i - (3)$$

여기  $c_1$  과  $c_2$ 는 웹사이트의 구조에 따라 변하는 상수이다. 분명한 것은 매개 변수  $c_1$  과  $c_2$ 에 대한 정의는 보다 검토되어야 할 상수이다. 즉, 지속적으로  $c_1$ 과  $c_2$ 는 분명히 알고리즘 실행과정에 영향을 미칠 수 있으며, 그 값이 성능을 향상시킬 수 있는 연구가 남아있다.

예를 들면  $c_1$ 의 값을 더 높게 만들면 만들수록 알고리즘을 HTML구조가 보다 깊이 링크가 활발하게 진행되도록 한다.  $c_1$ 이 최대가 되었을 경우에는 페이지 배열을 지워버리고 링크 편집 알고리즘이 실행 후에, 새로운 배열이 일어나게 하였다. 이것은 단순한 이론이 아니라 규형잡힌 페이지 배열을 달성시키는데 그 목적이 있다. 이것은 만약 관계 접근이 뚜렷하게 변하지 않는다면 알고리즘 실행에 의해 영향을 미치지 않는 형태로 남게된다. 매개변수  $c_1$  및  $c_2$  에 1의 값을 대입하면 방정식 3은 다음과 같이 다시 정리할 수 있다.

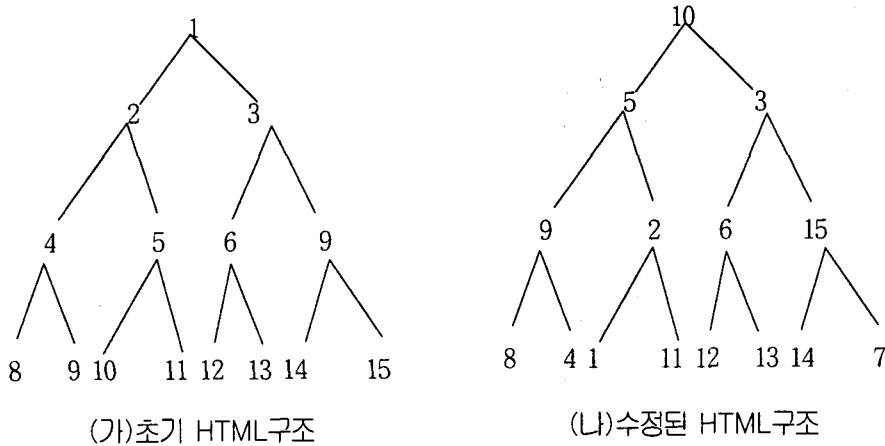


그림1 . 링크 편집 알고리즘은 웹사이트 내에서 페이지를 재배열.

$$a_i = d_i + n_i/r_i$$

첫째로 수집된 모든 PT 와 UT로부터 너무 짧거나 너무 긴 시간 값은 제거하고, 매우 짧은 시간은 링크를 통하여 사용자가 오로지 클릭만으로 다음 페이지를 나타낼 수 있다. 한시간 이상되는 너무 긴 시간은 외부 사정으로 브라우저를 중지하는 사용자의 경우를 나타낸다.

이진 트리구조를 가진 15개페이지(페이지1, 2, .....15)로 배열된 평가 사이트는 그림1. (가)에서 보여주고 있다. 그림에서 초기 홈페이지 인 페이지 1은 두 개의 하위계층의 자 페이지를 가지며 이는 페이지 2와 페이지 3이다. 페이지 1에서부터 7까지 매 페이지 n에 있어서, 페이지 n은 두 개의 하위계층 자 페이지를 가지며, 이는 페이지 2n 과 페이지(2n + 1)이다. 각 페이지(홈페이지를 제외한 부 페이지로 부터)에 대해 오로지 하나의 참조만 있다. 즉 알고리즘 r은 매페이지 경우에 1의 값을 가진다. 적용한 알고리즘은 (나)에서 나타내고 있는 개정된 이진 구조를 배열하고 있다. 변경내용은 구조내에서 보여주는

페이지 의 위치이다. 예를 들어서 링크 편집 절차의 마지막에, 페이지 2는 HTML 트리의 깊은 수준에있는 위치를 점유한다. 표1은 링크 편집을 실행하기 전에 시스템에 대한 접근을 종합하였다. 예를 들어서 페이지2는 27초의 평균 페이지 시간, 97절대접근, 291관계 접근(RA= a \* AA, 여기 a = d + n/r = 1 + 2/1 =3)을 가진다. 표 1에서 예측할 수 있는 바와 같이 초기 HTML구조는 오히려 부적절하다. 여기 페이지 10은 예컨대 과소평가되며, 다른 페이지는 홈페이지에 가까이 갈수록 보다 높은 RA를 나타낸다. 표 2에서는 개선된 이진 트리에 기초한 측정을 종합했다.

표 2에서부터 쉽게 볼 수 있는 것과 같이, 새로운 트리구조는 실제적으로 다른 AA를 가진 페이지에 기인한다. AA에 대한 종합 합은 다음과 같다.

$$SUM(AA) = AA(페이지) + AA(페이지2) + \dots + AA(페이지 15)$$

이것은 증가상태를 뚜렷이 나타내고 있으며,

종전 SUM(AA)는 726이고 새로운 SUM(AA)는 866이고 사용자 평균 시간 UT는 112초에서 146초로 증가 한다. 여기서 PT는 변동한다. 더욱이 새로운 페이지 배열은 2.04에서 2.67까 사용자 (NP)당 평균 페이지 수를 증가시키며 UT에 대한 것과 유사하다.

〈표 1〉 링크 편집에 앞서 접근에 대한 측정(그림 1 (가))

페이지	PT(초)	AA	RA
1	41	227	227
2	27	97	291
3	153	117	351
4	29	30	180
5	52	60	360
6	17	28	168
7	61	43	258
8	32	8	88
9	47	17	187
10	131	41	451
11	29	7	77
12	57	14	154
13	33	9	99
14	11	3	33
15	101	25	275

〈표 2〉 링크 편집 후 접근에 대한 측정: 수정된 HTML구조(그림 1. (나))

페이지	PT(초)	AA	RA
1	43	11	121
2	28	37	222
3	151	160	480
4	31	8	88
5	49	103	309
6	17	43	258
7	63	16	176
8	30	12	132
9	45	47	282
10	121	326	326
11	32	13	143
12	62	21	231
13	35	12	132
14	13	2	22
15	97	55	330

비록 측정값이 통계적이긴 하지만, 특정한 측정값은 유도할 수 있다. 첫째, 관계 페이지 접근에 근거한 링크 편집에 단순한 접근일지라도 특정한 웹사이트 쪽으로 사용자의 행위에 뚜렷이 영향을 미칠 수 있다. 표 1과 2를 비교하여 보면 페이지의 내용을 변경함이 없이 19%정도 본 사이트에 접근의 수는 증가 하였다는 것을 추론할 수 있다. 31%보다 높은 NP와 UT는 잠재적 사용자들에게 새로운 조직이 보다 편리하고 보다 매력적인 듯한 의미를 나타낸다.

이러한 재배열에 대한 다른 중요한 관점은 자동화 될 수 있다는 것이다. 즉, 단순한 링크 알고리즘을 사용하므로써, 보다 균형잡히고 조직된 상태로 웹사이트를 배열한다.

측정 도구의 기준 요구사항들은 적절하며 메인 메모리 16MB이상 가진 펜티움급 개인용 컴퓨터 정도 이면 이 프로그램을 동작 시키는데는 충분하고 페이지의 대중성을 계산할수 있다[8]. 본 연구에 사용된 웹서버는 알고리즘이 웹 서버 소프트웨어와 독립적이기 때문에 여하한 웹서버도 원래의 웹페이지나 수정된 웹페이지를 다룰 수 있었다.

전체적으로 다른 페이지 구조에 몇 개의 웹사이트에 측정 알고리즘을 적용했으며 페이지 접근에 있어서 개선된 것을 퍼센트로 표현하였으며 이는 표 3에서 보여준다. 분명한 것은 HTML 구조가 부적합하면 할수록 그것을 개선한 링크 편집 알고리즘에 좀더 관심을 가질 수가 있다. 알고리즘의 효율에 영향을 미치며 페이지간의 문맥의 결합성과 같은 구체적인 특성을 시험하기 위해서, 주어진 요구사항에 좀더 밀접하게 응답할 수 있도록 사이트를 생성하였다.

더욱이 측정 소프트웨어를 시험하기 위해 웹 사이트를 개발하였다. 이러한 새로운 서비스의 목적은 인터넷 사용자가 클릭을 사용하여 다른

서비스, 제품, 정보를 찾을수 있도록 도움을 주었다. 링크에 가변 정보를 가진 70개 웹 사이트 페이지를 설정하였다.

하기 위해서 각 가지쪽으로 구조의 깊은 수준에 감추어진 몇가지 선물용 페이지를 추가하였다.

〈표 3〉 링크 편집을 통한 웹페이지 접근 개선

사이트	노드의 수	AA 개선 (%)
취약한 이진 트리 페이지 배열 및 매 페이지는 자유스럽게 아주 가까운 관계 페이지로 상호교환 되어질 수 있다.	31	21
취약한 이진 트리 페이지 배열 및 매 페이지는 자유스럽게 아주 가까운 관계 페이지로 상호 교환되질 수있다.	15	19
도식 HTML구조는 부분적으로 배열되었다. 오로지 초기 홈페이지는 안정한 노드로 정의했다.	31	12
잘 배열된 사이트는 초기 홈페이지와 그 자 페이지는 안정한 노드로 정의 했다.	15	4

〈표 4〉 웹 사이트 : 링크 편집전과 후의 접근 측정

사 이 트	절대 접근의 수	평균 페이지 시간(초)
초 기	13.305	6.5
개선조치후	15.168	7.2

웹사이트의 초기 구조에 대해 몇 가지의 논리적 제한을 사용했으며, 사이트가 어떻게 생긴것이지에 관해서 순간적인 결정을 하였다. 그래서 15일 동안 접근이 가능하도록 사이트를 남겼다. 다음 초기 HTML 구조를 개선하기위해 링크 편집 알고리즘을 사용했으며 동일 시간 주기동안 접근성을 재시험했다. 표 4에서 와 같이 결과는 기대치에 매우 근접하였으며, 절대 접근에 14% 증가되었다. 평균 페이지 시간이 11%증가한 것은 사이트가 보다 사용자에게 매력을 느끼게 되었다는 것을 보여주고 있다.

페이지 내의 링크는 그림 2와 같이 도표를 배열하며 사이트에 대한 접근은 익명으로 되도록 하였고, 매 페이지는 방문자가 선택한 변경 링크를 추가해서 홈페이지에 링크가 가능하게 하였다. 더욱이 사용자가 보다 오랫동안 머물게

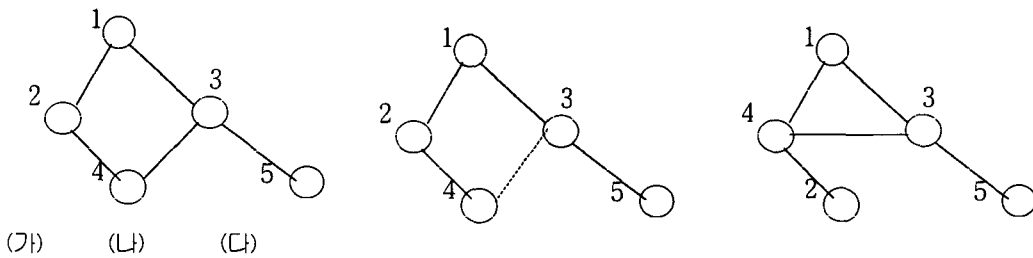
## 접근에 대한 제한[9]

HTML 재배열이 적절한 것이 아닌경우, 웹사이트의 구조가 논리적 제한의 규정을 따른다. 예를 들면, 음악내용의 기록 보관장소로 사용된 사이트의 사례를 보자, 초기페이지는 내용의 표이다.(A-Z, 여기 A는 모든 예술가의 이름이 A로부터 시작되도록 하였다.) 즉, HTML 트리의 가지는 각 예술가의 노래들이다. 비록 특별한 노래는 다른 사람들 보다 실질적으로 접근한다 손 치더라도 나머지 음악의 불합리한 여러 가지 문제점들에 대한 내비게이션을 만들기 때문에 HTML트리의 루트가 되어질 수 없다.

그래서 웹페이지 내용은 중요한것이며 관리자는 이를 페이지 재배열에서 고려하여야만 한다. 측정 도구의 관리자가 안정한 노드와 안정한 접속을 정의할수 있도록 가능하게 함으로서 그러한 문제점을 다룰 수 있다. 만약 관리자가 안정상태로 접속을 정의하면 접속된 노드는 교환되어질 수 없다. 동일하게 안정한 노드에 적용은 상위계층의 부경로나 하위계층의 자경로 노드로 상호교환 할수 없다.

도표 사이트에 대한 링크 편집

평가 목적만을 위해 이진 트리 구조를 가정하였다. 링크 편집 알고리즘은 도표사이트를 재배열함과 동시에 바로 그 작업을 수행한다. 이것은 모든 종류의 링크 즉, 부, 자, 손자 링크의 접속을 허용한다. 그림 2는 도표 사이트의 재배열을 보여준다.



- (가) 링크 편집 알고리즘을 적용하기 전, 각 노드에 개별적 id번호 부여. 즉 id = 4 를 가진 노드 4는 노드 2와 노드 3에 연결된다. 그러나 노드 2는 노드 4의 부노드 이다.
- (나) 실행하는 동안에 알고리즘은 노드 4와 노드 3사이의 링크는 무시한다.
- (라) 재 배열이 완성 될 때, 알고리즘은 노드 4와 노드 3사이의 링크를 재 설정한다.

그림 2. 도표 웹 사이트의 재배열

링크 편집 알고리즘의 다른 측면의 효과는 HTML 페이지의 위치를 변화시키는 것이 날자가지난 외부 사이트(예컨대, 검색엔진, 주제 카타로그, 등)에 링크를 내어줄 수 있는 것이다. 관리자들은 특별히 링크 편집 알고리즘이 유용한 많은 페이지의 교체결과를 가져오거나 빈번히 실행되어진다면, 깨어진 링크는 네트워크 사용자들을 괴롭힐 수도 있다는 것을 관리자들은 명심해야한다.

잉크햄(D. Ingham)과 동료들은 깨어진 링크를 면하기 위해 대체지향 접근을 제안하였다 [10]. 그리고 케이프(F. Kappe)는 갱신된 정보를 전달하기 위해 서버 갱신 프로토콜을 제안하였다 [11]. 그러나 웹관리자로부터 작은 효과를 요구하는 보다 용이한 많은 방법들이 있다. 링크 편집 절차동안 측정 도구는 교체되어지는 페이지의 URL과 그들의 새로운 URL들을 세밀하게

표현할 수도 있고 그들을 데이터베이스에 담을 수도 있다. 즉 관계형 데이터베이스나 웹서버로부터 단순한 파일접근이 가능한 데이터 베이스에 담을 수도 있다. 그래서 관리자는 사용자가 갱신되지 않는 외부 링크를 연결하고자 할 때 본 데이터 베이스를 점검하기 위해 특별히 오류 처리자를 배열할 수 있다 (오류 조건을 처리하기 위해 아파치 웹 서버를 배열하는 방법에 관한 정보는 "808 Not found"라는 메시지를 되돌리는 대신에 오류 처리자는 날자지난 링크가 링크 편집 동안에 다른 사람에 의해 교체할 것인지 아닌지를 결정한다. 만약 이러한 페이지가 데이터베이스에 존재하면 메시지는 새로운 URL에 대한 사용자에게 정보를 주며 자동적으로 페이지의 새로운 위치로 지시하기 위해 브라우저에게 재지시한다. 만약 파일이 존재하지 않으면 브라우저는 오류메세지를 나타낸다. 이러한

방법으로 만약 링크 편집 알고리즘이 모든 웹 서버의 페이지를 수정하거나 검색엔진의 참조까지도 그들의 URL들을 변경한다면 치더라도 그것은 깨어진 링크는 아니다.

오류 스크립트 처리자의 실행에 의해 발생한 웹서버에 부가적인 부하를 줄이기 위해 여기서는 관리자가 링크를 갱신 할 수 있도록 홈페이지 URL을 검색 엔진으로 재송부하는 것을 권고한다. 갱신은 보통 한달 간격으로 검색 엔진 재배포후에, 모든 네트워크 사용자에게 가능하다.

### III. 결론

HTML의 잘못된 재 배열은 인터넷 사용자들이 웹사이트를 방문하거나 잠시 접속하여 머물고자 할때 실망을 안겨줄 수 있다. 본 논문에서는 링크 편집 알고리즘을 통하여 각 페이지의 상대적 대중성을 계산하여 영성한 배열의 구조를 자동으로 고정시키므로써 상대적 대중성을 높일 수 있다는 것을 보여주었다. 상대적 대중성을 사용한 단순한 접근 즉 방정식 1과 2의 결합은 실제적으로 향상된 배열을 이르게한다.

본 연구에서는 목표가 사용자가 요청한 데이터를 찾는 것을 좀더 쉽게하는 것이다. 즉 검색을 빠르면 빨수록 웹 서버의 구조는 보다 더 좋다. 웹 사이트가 상업 자료 즉 광고 및 기타 등을 가진다면 그 목표는 다를 수 있다. 즉 최선의 구조는 상업적 관심을 가진 페이지에 대해 가장 높은 AA를 달성하는 것 중 하나가 될 가능성이 있다. 이러한 경우 각 페이지에 대해 가중 인자  $w$ 와 연관 시킬 수 있으며 페이지가 보다 높은  $w$ 가 보다 활발하게 페이지를 진행시킬 수 있도록 하는 것이 바람직 할 것이다.

대중성의 개념을 판단하는 많은 다른 방법과 페이지 사이에서 링크를 재배열 하기 위한 많은 다른 접근이 있다. 예를 들면 계수  $a$ 를 방정식 3에서 주어진 것보다 더 복잡한 정의로 제공할 수 있거나 또는 링크 편집 알고리즘보다 더 복잡한 방법을 사용하여 초기 페이지 구조를 재배열할 수 있다. 그럼에도 불구하고 본 논문에서 나타난 접근은 과소평가 되지 않아야만 한다, 반면에 분석하기가 쉽고 페이지 접근에서 퍼센트 증가를 고려하면 대부분의 경우 적절하다는 것이 검증되었다. 새로운 링크를 추가할 수 있고 상호 교환하는 페이지에 추가에 있어서 다른 것들을 버릴 수 있는 링크 편집 알고리즘은 보다 향상된 것을 보여 줄 수있을 것으로 사료된다.

### 참고문헌

- [1] M. Creech, "Author-Oriented Link Management," Computer Networks and ISND Systems, Vol. 28, 1996, pp.998-1028.
- [2] [http://www.siteowner.com/\(SiteOwner Link Exchange Site\)](http://www.siteowner.com/(SiteOwner Link Exchange Site)).
- [3] [http://www.webtrends.com/\(WebTrends\)](http://www.webtrends.com/(WebTrends))
- [4] R. Beasley and M. Waugh, "Cognitive Mapping Architectures and Hypermedia Disorientation: an Empirical Study," J. Educational Multimedia and Hypermedia, Vol. 4, No. 2/3, 1995, pp.230-260.
- [5] P.Zaphiris and L. Mtei, "Depth ve Breadth in the Arrangement of Web Links," unpublished paper, available online of <http://www.otal.umd.edu/SHORE/bs04/index.html>



- 
- [6] HTTP Request Fields(RFC822 header lines), available online at <http://www.w3.org/Protocols/HTTP/HTRQHeaders.html>.
- [7] G. Golvchinsky, "What the Query Told the Link: The Integration of Hypertext and Information Retrieval, "Proc. 8th ACM Conf. Hypertext, ACM Press, New York, 1997, pp. 67-74.
- [8] SOALA, Tech, Report CTI-TR98.3.16, Computer Technology Inst., Patras, Greece, 1998.
- [9] A. Baylor, "Disorientation on the Web-Adventure or Distraction" Proc. WebNet '98: World Conference of the WWW, Internet, and Intranet, Assoc. for the Advancement of Computing in Education (AACE), Cahrlottesville, Va., 1998, pp. 45-60
- [10] F. Kappe, "A Scalable Architecture for Maintaining Referential Integrity in Distributed Information Systems, "J. Unversal Computer Scieence, Vol., 1, No. 2, Feb. 1995, pp.84-104
- [11] <http://www.webring.org>

## A Study on Reorganization of Web Site Based on Approach Using Page Popularity.

Suk-Pal, Cho\*

### Abstracts

The performance and quality of Web sites are often being estimated by the frequency that the users approach its site. This paper suggest how a link-editing method can automatically fix a poor organization by calculating each page's relative popularity, and how improve its poor pages. Page's relative popularity depended on only cases where the objective is to make it easier for a user to find the requested data; the faster the access, the better the organization of the Web sites according to tree depth.

---

\* Dept. of Computer Based Information Eng. SungKyu Christian Univeristy.