(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
G06F 17/30 (2006.01)

(45) 공고일자 2006년04월03일
(11) 등록번호 10-0567005
(24) 등록일자 2006년03월27일

(21) 출원번호 10-2000-7004808
(22) 출원일자 2000년05월03일
번역문 제출일자 2000년05월03일
(86) 국제출원번호 PCT/US1998/022331
국제출원일자 1998년10월21일
(65) 공개번호 10-2001-0031743
(43) 공개일자 2001년04월16일
(87) 국제공개번호 WO 1999/23581
국제공개일자 1999년05월14일

(81) 지정국 국내특허 : 알바니아, 아르메니아, 오스트리아, 아제르바이잔, 보스니아 헤르체고비나, 바르바도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀란드, 영국, 그루지야, 헝가리, 이스라엘, 아이슬랜드, 일본, 케냐, 키르키즈스탐, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 리베이라, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아공화국, 몽고, 말라위, 멕시코, 노르웨이, 슬로베니아, 슬로바키아, 타지키스탄, 투르크맨, 터어키, 트리니아드토바고, 우크라이나, 우간다, 미국, 우즈베키스탄, 베트남, 폴란드, 포르투칼, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 가나, 감비아, 짐바브웨, 세르비아 앤 몬테네그로, 인도네시아, 시에라리온, 크로아티아, 그라나다,

AP ARIPO특허 : 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 가나, 감비아, 짐바브웨,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르키즈스탐, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크맨,

OA OAPI특허 : 부르키나파소, 베닌, 중앙아프리카, 콩고, 코트디브와르, 카메룬, 가봉, 기니, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기니 비사우,

(30) 우선권주장 08/963,372 1997년11월03일 미국(US)

(73) 특허권자 야후, 인코포레이티드
미국, 캘리포니아 95051, 산타 클라라 , 센트랄 익스프레스웨이 3400, 수트 201

(72) 발명자 우,지옹
미국,캘리포니아94555,프레몬트,가드월코먼34305

(74) 대리인 한양특허법인

심사관 : 손영태

(54) 계층 복합 문서로부터의 정보 검색

요약

문서가 계층으로 편제되어 있는 문서 저장부 내의 문서들에 탐색질의(30)가 적용된다. 탐색엔진은 계층을 탐색하여 질의어와 직접 혹은 간접적으로 일치하는 문서들을 다시 보내준다. 탐색엔진의 특정 실시예는 질의어를 개별 부제어로 편제하고, 문서들에 대해 부제어를 일치시켜, 전체 탐색 질의어와 간접적으로 일치하며 질의 부제어 중 적어도 하나와 직접적으로 일치하는 문서들만을 돌려준다.

대표도

도 1

명세서

배경기술

본 발명은 전자문서 저장 및 관리 분야에 관한 것이다. 특히, 본 발명의 일 실시예는 복합 문서를 저장하고 저장된 복합 문서를 탐색하는 시스템을 제공한다.

최근에 정보는 부족한 유용한 것에서 과잉의 유용한 것으로 옮아가고 있다. 부족한 유용한 것에 있어서는 유용한 것을 얻는데 중점을 두었으나 과잉의 유용한 것에 있어서는 이를 보다 유용하게 위해서 유용한 것들을 걸러내는데 중점을 두고 있다. "인터넷"이라고 알려진 전세계 망들의 상호망 성장에 기인한 폭발적인 정보는 이러한 현상의 주 예이다. 망 및 인터넷에 접속된 컴퓨터는 데이터 패킷을 소스 노드에서 목적지 노드로 신뢰성있게 전달하기 위해서 TCP/IP(전송 제어 프로토콜/인터넷 프로토콜)을 사용하여 데이터를 전달한다. 디지털 데이터의 객체를 전송하기 위해서 TCP/IP의 최상위에 다양한 상위레벨의 프로토콜이 사용되는데, 특정한 프로토콜은 객체의 본질에 따른다. 예를 들면, 전자메일은 단순 메일 전송 프로토콜(SMTP) 및 포스트 오피스 프로토콜 3(POP3)를 사용하여 전송되며, 파일들은 파일 전송 프로토콜(FTP)을 사용하여 전송된다. 하이퍼텍스트 문서 및 이들에 관련된 이펙트는 하이퍼텍스트 전송 프로토콜(HTTP)을 사용하여 전송된다.

많은 하이퍼텍스트 문서가 다른 하이퍼텍스트 문서에 연결되어 있을 때 이들은 총괄적으로 "웹" 구조를 형성하게 되므로 HTTP를 사용하여 전송될 수 있는 하이퍼텍스트 문서 전체를 "월드 와이드 웹"(통상 "WWW" 혹은 "웹"이라 약칭함)이라 명명하게 되었다. 물론, 하이퍼링크는 HTTP를 사용하여 전송되는 문서에서 필요한 것은 아니다. 사실, HTTP 요건에 따르기만 한다면 어떠한 객체도 HTTP를 사용하여 전송될 수 있다.

HTTP의 전형적인 사용을 보면, 브라우저는 균일 자원 로케이터(URL)를 웹 서버에 보내고 웹 서버는 브라우저가 표시할 수 있게 하이퍼텍스트 마크업 언어(HTML) 문서를 보내온다. 브라우저는 HTTP 클라이언트의 한 예이며 보내온 하이퍼텍스트 문서를 표시하고 이 문서 내에 인용된 다른 하이퍼텍스트 문서들을 사용자에게 선택 표시할 기회를 주게 하므로 그와 같이 명명되었다. 웹 서버는 HTTP 클라이언트에 의해 요구된 하이퍼텍스트 문서를 보내주는 인터넷 노드이다.

일부 웹 서버는 정적 문서를 서비스하는 것 외에도 동적 문서를 보낼 줄 수 있다. 정적 문서는 문서가 요청되기 전의 웹 서버에 존재하는 문서로서 이 문서에 대해 웹 서버는 단순히 요청이 있을 때 정적 문서를 내보낸다. 정적 페이지 URL은 통상 "host.subdomain.domain.TLD/path/file" 등의 형태로 되어 있다. 이러한 정적 페이지 URL은 "host.subdomain.domain.TLD"의 도메인 이름을 갖는 머신 상의 경로 "/path/"에 있는 "file"이라고 명명된 문서를 말한다. 실제 도메인인 "www.yahoo.com"은 ".com" 최상위 도메인(TLD) 내에 도메인 "yahoo"에서 "www"로 지정된 머신(혹은 머신들)을 말한다. 반대로, 동적 문서는 서버가 동적 문서에 대한 요청으로서 확인하는 특정 URL을 수신하였을 때 웹 서버에 의해 발생되는 문서이다.

많은 웹 서버는 한 사람 혹은 엔터티에 의해 제어되는 링크된 하이퍼텍스트 문서 전체를 제공하는 "웹 사이트"를 운영한다. 웹 사이트는 한 사람 혹은 엔터티에 의해 제어되기 때문에, 본 명세서에서 "웹 페이지"라고도 하는 하이터텍스트 문서는 일관된 외양 및 주제를 갖는다. 특히 상품 및 서비스를 판매하는 상업적인 이해관계로 게시된 웹 사이트의 경우, 웹 사이트를 형성하는 하이퍼링크가 된 문서는 이해 당사자에 의해 제어되지 않는 페이지에 있다고 해도 거의 링크는 없다. "웹 사이트" 및 "웹 페이지"라는 용어는 흔히 구별 없이 사용되나, 여기서는 "웹 페이지"는 웹 사이트의 일부를 형성하는 단일의 하이퍼텍스트 문서를 말하며 "웹 사이트"는 특정 기사에 관해 사이트를 게시하기 위해서 함께 작업하는 단일 엔터티 혹은 일군의 엔터티들에 의해 제어(즉, 수정될 수 있는)되는 하나 이상의 웹 페이지 전체를 지칭한다.

수백만의 인터넷 사용자가 이들의 웹 서버를 통해 이용할 수 있을 모든 많은 사이트 및 페이지에 의해서는 특정 페이지를 찾거나, 특정 기사에 관한 정보를 어디서 찾을 것인가 결정하기가 어렵다. 누구든 그들의 웹 서버에 무엇이든 놓아둘 수 있고 이를 공식 에이전시에 보고할 필요도 없고 웹이 매우 빠르게 변하기 때문에, 이용할 수 있을 것을 목록한 "공식적인" 목록은 없다. 공식적인 "내용물 테이블"이 없으므로, 웹을 색인하는 몇 가지 방법이 제안되어 있다.

한 방법은 모든 곳에서 발견되는 모든 웹 문서를 색인하는 것이다. 이 방법은 드물게 다루어지는 기사나 보기 드문 이름이나 성을 가진 사람에 대한 조회에 관한 문서를 찾는데 유용하지만, 이 방법은 "히트(hit)"수가 너무 과다하게 된다. 또 다른 방법은 웹 문서를 요약하여 분류하고 카테고리로 요약된 것을 탐색할 수 있게 하는 것이다.

어느 경우이든, 전형적인 탐색엔진은 각각의 후보 목록 내의 탐색어들을 찾고 탐색기준을 충족하는 문서 목록을 보내준다. 불행하게도, 문서들간 상호관계로부터 얻어질 정보가 상실된다. 전술한 바로부터 문서간 상호관계를 고려하는 개선된 탐색 시스템이 필요하다.

< 발명의 개시 >

링크에 걸쳐 탐색함으로써 문서간 상호관계를 고려하는 개선된 탐색 시스템이 본 발명에 의해 제공된다. 본 발명의 일 실시예에서, 문서는 키워드 및 주제 탐색에 사용되는 계층 문서 저장부 내의 문헌들이다. 탐색질의는 계층에 적용되는데, 이것은 질의어 모두 혹은 그 일부와 일치하는 페어런트(parent) 문서로부터 계층으로 차일드(child) 문서인 것에 의해서 탐색 질의어와 직접 일치하거나 탐색 질의어와 간접적으로 일치하는 문서를 보내준다. 바람직한 실시예에서, 보내온 문서는 질의어 중 적어도 한 부제어와 직접적으로 일치한다.

본 발명의 한 이점은 계층내 데이터 원소들간 관계를 고려하여 탐색이 수행되게 하면서 계층 데이터를 유용하게 저장할 수 있게 하는 것이다.

본 발명의 특징 및 이점에 대해서 명세서의 나머지 부분과 첨부한 도면을 참조하여 이해할 수 있다.

도면의 간단한 설명

도 1은 본 발명의 일 실시예에 따른 탐색엔진을 구비한 클라이언트-서버 시스템의 개략도이다.

도 2는 문서 저장부의 부분들에 대응하는 문서의 트리 그래프이다.

도 3은 도 1에 도시한 클라이언트-서버 시스템의 요소들에 대한 보다 상세한 도면으로, 문서 저장부, 단어 색인 및 탐색엔진의 상세히 도시한 도면이다.

도 4(a)-(c)는 도 3에 도시한 탐색엔진이 사용하는 일치목록의 예이다.

도 5는 본 발명의 일 실시예에 따른 탐색결과의 브라우저 디스플레이를 찍은 스크린이다.

도 6은 탐색엔진에 의해 수행되는 AND 연산의 흐름도이다.

도 7은 탐색엔진에 의해 수행되는 OR 연산의 흐름도이다.

실시예

본 발명은 상당히 많은 문서를 포함하는 문서 저장부에서 선택된 문서를 찾기 위한 장치에 관한 것이다. 본 발명의 일 실시예에 따른 탐색엔진은 탐색표현식을 수신하고, 그 탐색표현식에 기초하여, 문서 저장부 내에 일치하는 문서들을 찾고 일치하는 문서들 혹은 이들 일치하는 문서 각각에 대한 참조목록을 내보낸다. 탐색표현식이 2개 이상의 부제어로 된 복잡한 논리함수인 경우, 탐색엔진은 단지 간접적으로 부제어의 일부와 일치하는 문서들을 내보낼 것이다. 예를 들면, 탐색표현식은 3개의 부제어의 "AND"일 수 있다. 모든 3개의 부제어를 포함하는 문서들만을 내보내는 대신에, 탐색엔진은 루트 노드까지 문서로부터 계층경로을 따라 문서들 내에 어느 곳이든 나머지 부제어가 발견된다면, 단지 하나 이상의 부제어만을 갖는 문서라도 내보낼 것이다. 일부 변형예에선 모든 부제어에 대해서 단지 간접 일치만이 된 문서들을 내보낸다. 그러나 바람직한 실시예에선, 내보내지는 문서는 적어도 하나의 부제어와 직접적으로 일치해야 한다.

본 발명에 대해서 특정 유형의 문제에 관련하여 여기서 기술할 것이나 여기 기술된 본 발명 및 실시예는 많은 다른 형태의 문서에도 사용할 수 있다는 것을 알아야 한다.

여기 주 예에서 기술된 문서는 탐색 데이터베이스에 있는 레코드이다. 탐색 데이터베이스는 카테고리의 계층구조 및 사이트 참조로서 편성되어 있다. 이 구조는 자동적으로 발생될 수도 있겠으나, 야후! 탐색 데이터베이스로서 알려진 실시예에서 카테고리 및 사이트 참조는 사이트 제출인으로부터 경험과 제안을 사용하여 편집부원에 의해 계층 내의 적합한 위치에 배치된다.

카테고리 및 사이트 참조를 포괄하여 구조의 노드라고 칭한다. 일부 카테고리 노드는 패어런트 노드인데, 이들 노드는 패어런트 노드로 나타낸 카테고리의 더 특정한 서브카테고리를 나타내는 다른 카테고리 노드(차일드 노드)를 가리키기 때문이다. 사이트 노드는 (특정 사이트가 복수의 카테고리 내에 목록될 수 있으므로 몇몇 서브트리 내의 차일드 노드일 수도 있어도) 카테고리 노드로부터의 차일드 노드이다.

여기서, 노드는 패어런트, 차일드, 선조 혹은 또 다른 노드의 후손 노드로서 기술될 수 있다. 노드 N에 대한 패어런트 노드는 계층 내에서 노드 N의 한 레벨 위의 노드이며, N의 차일드 노드들은 계층 내에서 노드 N의 한 레벨 이하의 노드들이고, N의 선조 노드들은 노드 N의 임의의 레벨 위의 노드들이고, N의 후손 노드는 노드 N의 임의의 레벨 밑의 노드들이다. 통상, 계층은 선조 노드들이 없고 모든 다른 노드들을 후손 노드로서 갖는 루트 노드이다.

여기 기술된 실시예에서, 카테고리 노드는 카테고리 노드나, 사이트 노드 혹은 차일드 노드들로서 이들 모두를 가질 수 있으나, 사이트 노드는 차일드 노드를 갖지 않는다. 일부 카테고리 노드는 차일드 노드를 전혀 갖지 않을 수도 있으나 이러한 공 카테고리는 삭제되거나 은닉되는 것이 바람직하다. 또한, 카테고리 노드 모두가 차일드 노드를 가질 필요는 없는 것이지만 공 카테고리는 불필요한 혼란을 피하기 위해서 삭제되거나 은닉되는 것이 바람직하다.

도 1은 이러한 탐색 데이터베이스가 조회되는 클라이언트-서버 시스템(10)을 도시한 것이다. 시스템(10)은 인터넷(16)을 통해 탐색서버(14)에 접속된 HTTP 클라이언트(12)를 포함하는 것으로 도시되었다. 탐색서버(14)는 문서 저장부(20) 및 단어 색인(22)에 결합되며 탐색 요구(30)에 탐색 결과(23)로서 응답한다.

이러한 특정한 예에서, HTTP 클라이언트(12)는 브라우저이지만, 이 대신에 탐색 후위(back-end) 프로세서 등의 다른 HTTP 클라이언트를 사용할 수도 있다. 또한, 시스템(10)은 HTTP 클라이언트(12)와 탐색서버(14) 간에 대안이 되는 통신 채널로 대치된 인터넷(16)으로 구현될 수도 있다. 더구나, 탐색서버(14)가 HTTP 서버일 때 HTTP 클라이언트(12) 혹은 이의 대용물이 상이한 프로토콜을 이해하는 한 전적으로 다른 프로토콜을 사용하여 요구를 처리할 수도 있음을 알아야 한다. 간략화를 위해서, 단지 하나의 HTTP 클라이언트, 하나의 요구 및 하나의 응답만이 도시되었지만 실제로는 많은 클라이언트가 각각의 하나 이상의 탐색 요구하면서, 거의 동시에 탐색 서버(14)를 액세스하고 있을 것임을 알아야 한다. 사실, 보증된다면, 탐색서버(14)의 작업들은 복수의 머신들에 나눌 수도 있다. 작업이 복수의 머신들에 분배되면, 바람직한 구성으로서는 복수의 머신들을 클라이언트에게 단일의 논리적인 머신으로서 나타나게 하여 클라이언트 액세스를 단순화시키는 것이다.

동작을 살펴보면, 브라우저 사용자, 혹은 다른 HTTP 클라이언트는 탐색 표현식을 포함하는 요구(30)를 탐색서버(14)에 보내고 이 탐색서버(14)는 탐색 표현식을 분해하는데, 이 탐색 표현식이 유효한 형식으로 되어 있으면 탐색표현식을 사용해서 문서 저장부(20)에서 탐색표현식과 일치하는 문서들을 찾는다. 탐색서버(14)는 일치하는 문서 목록 혹은 그 문서들 자체로 응답한다. 단어 색인(22)은 문서 저장부(20) 내에 문서들에 대한 탐색을 가속화하는데 사용된다.

도 2는 문서 저장부(20) 내에 문서가 어떻게 논리적으로 구성되어있는지를 도시한 것이다. 이 예에서, 문서들은 원하는 WWW 사이트를 찾는데 사용되는 탐색 데이터베이스의 원소들이다. 각각의 문서는 주제 카테고리 혹은 사이트를 나타내며, 각각의 문서는 계층구조 내의 레코드(38)가 다른 레코드들과는 패어런트 혹은 차일드 관계에 있는 것으로서 도시되어 있다. 각각의 레코드(38)는 문서의 번호(40)와 내용(46)을 갖는 것으로 도시되었다. 카테고리인 문서의 경우, 내용(46)은 카테고리의 타이틀 및 은닉 키워드, 동의어, 서술 등의 다른 텍스트(도시없음)이며, 사이트라고 하는 문서의 내용은 타이틀, URL, 서술, 은닉 키워드, 동의어 등을 포함한다. 물론, 이들 원소 일부는 적합한 곳이나 원하는 곳이 블랭크일 수 있다. 앞서 설명한 바와 같이, 야후! 탐색 데이터베이스에서, 문서는 편집부원에 의해 계층구조 내에 배치된다. 전형적인 과정에서, 사이트 발기인은 사이트 타이틀, 사이트 URL, 계층내 제안 위치, 서술 등의 사이트 정보를 편집부원에게 제시할 것이다. 이 때 편집부원은 제안을 평가하여, 보다 적합한 위치가 존재한다면 제시된 위치를 변경하고, 필요할 때 링크를 연결시키고, 경우에 따라서는 은닉 키워드, 동의어 및/또는 문서에 중요도 가중치를 부가한다.

보다 일반적인 주제(패어런트 노드)를 보다 구체적인 주제 혹은 사이트 참조(차일드 노드)에 각각의 링크가 연결된 상태로, 레코드들간 링크를 도 2에 도시하였다. 예를 들면, 번호 5의 문서는 "고우(Go)"에 관계된 WWW 사이트에의 사이트 참조인데, 이 "Go"는 보드 게임이며 따라서 "보드 게임"의 부주제이고, 이것은 "게임" 주제의 부주제이고, 이것은 "레크리에이션" 주제의 부주제이고, 등등이다.

번호 5 및 6 문서 등의 일부 사이트 참조는 리프(leaf) 카테고리(즉, 차일드 카테고리 노드를 전혀 갖지 않는 것)의 노드 오프(nodes off)인 반면, 번호 21의 문서와 같은 다른 문서들은 논-리프 카테고리의 노드 오프이다. 총괄적으로, 링크들은 트리와 서브트리를 정의하는데, 이들은 후술하는 바와 같이 임의의 서브트리 내의 문서들이 서브트리의 최상위에 문서의 문서번호 다음에 연속적으로 번호가 매겨지도록 번호가 할당된다.

도 3은 도 2에 도시한 정보를 다른 형태로 도시한 것이다. 도 3은 도 2의 트리 구조의 노드에 대응하는 데이터 테이블 내의 레코드(38)를 갖고 저장될 정보를 나타낸다. 도 3의 데이터 구조는 많은 가능한 데이터 구성 중 하나를 나타내고 있음을 알아야 한다. 단지 소수의 레코드(38)만을 나타내었고 실제로는 수백만개의 레코드가 있을 수 있다.

레코드(38)에 대해 보인 필드는 문서의 번호(40), 서브트리 내에 마지막 노드를 가리키는 서브트리 포인터(42)(필요로 할 때 도중에서 저장되어 발생되거나 메모리 어레이로부터 얻어질 수 있음), 패어런트 노드를 가리키는 패어런트 포인터(44), 레코드로 나타낸 문서의 텍스트(여기서는 타이틀(46) 및 서술(47)로 나타내었음), 문서에 관련된 하나 이상의 키워드의 선택적 세트(48), 및 레코드가 카테고리용인지 아니면 사이트용인지에 대한 부울린 표시(50)이다. 도 2에 도시한 바와 같이, 일부 노드는 WWW 사이트를 가리키며 다른 노드들은 사이트 요소가 하나 이상의 카테고리 요소에 연관되어 있는 계층 주제 카테고리 구조 내의 카테고리들을 나타낸다. 이 예는 주제 및 사이트 참조의 트리구조에 대한 것이지만, 여기 기술된 시스템은 보다 많은 복잡한 문서를 탐색할 수 있다는 것이 본 명세서로부터 명백하다.

카테고리 노드에 대해서, 레코드(38)는 타이틀(46)과, 카테고리 서술(47), 아마도 한 세트의 은닉 키워드(48)를 포함한다. 사이트 노드에 대해서, 레코드는 타이틀, 사이트에 대한 서술(블랭크일 것임), 및 참조된 사이트/페이지를 가리키는 URL을 포함한다. 아울러, 문서의 번호(40), 서브트리 포인터(42) 및 패어런트 포인터(44)는 레코드들간의 연계를 기술한다. 예를 들면, 번호 2 문서는 이의 서브트리 포인터로서 "8"을 가지며, 3(이 문서의 번호에 하나를 더한)부터 8(서브트리 포인터 값)까지 번호가 매겨진 모든 문서는 문서 2 이하의 서브트리에 있음을 나타내며, 번호 2 문서의 패어런트 포인터로서 "1"은 문서 1이 문서 2의 패어런트 문서임을 나타낸다. 번호 2 문서에 대한 레코드(38)의 다른 필드는 이의 내용물이 "게임"임을 나타내며, 아무 키워드도 목록되어 있지 않으며 이것은 카테고리이다(사이트 참조와는 반대임). 서브트리에서 마지막 문서의 번호만을 사용하여 전체 서브트리를 명시할 수 있는데 이것은 문서의 번호의 순서가 특정하게 할당되어 있기 때문이다.

문서 저장부(20)는 문서 번호로 요구된 문서들을 다시 보내는데 필요한 처리 로직을 포함하며, 문서 저장부(20)나 탐색 엔진(36)은 질의어와 일치하는 필드값의 보기에 대한 레코드를 탐색하는 처리 로직을 포함한다.

도 3에 도시한 단어색인(22)에 대해서, 단어색인(22)의 내용물의 소 부분집합이 도시되었다. 단어색인(22)은 문서 저장부(20)의 문서들에 나오는 단어당 하나의 레코드를 갖는 복수의 레코드로서 편제되어 있으며, 단어의 알파벳 순서로 분류되어 있다. 단어색인(22) 내의 각각의 레코드(52)는 단어를 식별하는 태그와 문서 번호 목록이 이어지는 것으로 도시되어 있다. 이들 문서 번호들은 단어를 포함하는 문서 목록을 나타낸다. 단어가 "the"와 같이 한정된 유용성의 탐색어로 일반적인 경우 이의 레코드는 그 단어를 포함하는 모든 문서를 목록하지 않으며 그 단어는 무시될 것임을 나타낸다.

바람직한 실시예에서, 탐색엔진(36)은 탐색속도를 개선하기 위해서 문서 프로파일 어레이(49)를 사용한다. 통상, 어레이(49)는 신속한 액세스를 위해서 메모리에 저장된다. 어레이(49)는 문서상 하나의 레코드를 가지며 각각의 레코드는 문서 번호, 문서 저장부 포인터, 시간 스탬프, 차일드들의 범위 및 중요도 가중치용 필드를 포함한다. 문서번호는 문서 저장부(20)의 문서 번호에 대응하며 문서 저장부 포인터는 문서 저장부(20) 내의 문서의 물리적인 디스크 위치에 대응하므로, 어레이(49)는 문서 저장부(20) 자체의 액세스를 필요로 하지 않는 문서에 대한 어떤 조작을 수행하는데 사용될 수 있다. 시간 스탬프는 문서가 마지막으로 수정된 시간을 확인한다. 차일드 필드의 범위는 계층에서 어느 레코드가 현 문서 밑에 있는지를 나타내므로, 탐색엔진(36)은 빈번히 문서 저장부(20)를 참조할 필요없이 일치목록을 신속하게 구축할 수 있다.

중요도 가중치는 자동적으로 혹은 편집부원에 의해서, 특정 카테고리 혹은 사이트가 다른 카테고리 및 사이트에 대해 얼마나 가치가 있고 관계되어 있는가를 나타내도록 설정되는 값이다. 레코드의 중요도 가중치는 외부 이벤트 혹은 사이트 중요성에 근거하여 조정될 수 있다. 예를 들면, 현재 뉴스에 있는 특정 그룹에 관계된 사이트에 높은 가중치가 부여될 수 있고, 혹은 사이트가 인기가 있고 잘 설계되어 있는 것으로 편집부원이 판정한 경우엔 그 사이트에 높은 가중치가 부여될 수 있

다. 도 3에 도시한 어레이(49)에서 카테고리 레코드가 가중치 부여된 카테고리가 아닐지라도, 가중치가 부여된 카테고리가 사용될 수 있을 것이다. 예를 들면, 보우팅 시즌동안에는 번호 9 문서(카테고리 "보우팅")에 대한 가중치가 증가될 수 있다. 레코드의 가중치는 탐색결과로서 복수의 문서가 표시되고 있을 때, 이들의 가중치 순으로 문서가 표시되는 것으로 작용하게 된다. 물론, 후술하는 바와 같이, 다른 가중치 부여 인자가 중요도 가중치 부여에 앞서거나 이에 결합하여 전체 가중치 부여를 형성할 수 있다.

문서 저장부(20) 및 단어 색인(22) 사용에 대하 예를 들어 기술한다. 이 예에서, 사용자는 문서를 탐색하고 있으며 "고우(Go) 게임"이라는 질의 스트링으로 탐색 요구를 하고 있다. 탐색엔진(36)은 단어색인(22) 내에 각각의 어를 찾아본다. "the" 및 "of"는 일반적인 것이기 때문에 탐색엔진(36)에 의해 무시되거나 단어색인(22)은 전술한 바와 같이 이들 단어를 무시하는 명령을 다시 보내온다. 탐색엔진(38)은 각 어에 대한 일치목록을 생성하고 도 6에 관련하여 후술하는 바와 같이 일치목록에 "AND" 연산자를 적용한다.

일치목록은 질의어가 무엇인가에 따라, 직접 혹은 간접적으로 목록의 일치 태그를 포함하는 모든 문서의 목록이다. 도 2는 간접일치가 왜 중요하지를 예시하고 있다. 본 명세서 전체를 통해 사용된 예는 흑백 마커를 사용하는 잘 알려진 보드게임인 "고우" 사이트에 관계된 카테고리 및 사이트 탐색이다. 게임명은 영어에서 일반어와 동일할 수 있기 때문에, "고우"를 탐색하게 되면 관계없는 너무 많은 일치로 될 것이다. 그러나, 알 수 있는 바와 같이, 동일 문서에 "고우" 및 "게임"을 탐색하게 되면 일치는 없게 될 것이다. 그러므로, 각각의 문서는 이의 모든 선조 문서의 모든 탐색가능한 요소(탐색가능한 요소는 이 경우 단어임)를 포함하는 것처럼 탐색될 필요가 있다. 물론, 모든 선조 문서의 내용물은 이의 서브트리의 후손 문서 각각에 삽입될 수 있으나, 트리가 큰 경우, 이러한 방법은 낭비적이고 실제적이지 못하다.

도 2-3에서, 일치목록 상의 각각의 어는 직접일치의 경우 단일 문서를 언급하며, 혹은 간접일치의 경우 한 범위의 문서를 언급한다. 도 4는 일치목록의 몇 가지 예를 도시한 것이다. 먼저, 도 4(a)는 도 2-3에 도시한 특정문서에 대응하는 일치목록(60)이다. 일치목록(60)은 3개의 어, 혹은 일치 레코드를 포함한다. 첫 번째는 번호3 문서가 일치태그와 일치하는 것을 나타내는 직접 일치 레코드이며 두 번째는 번호4 내 번호 8 문서가 일치태그와 일치하는 것을 나타내는 간접 일치 레코드이다.

이 예에서, 일치 태그는 텍스트이기 때문에, 문서 내용 중에 스트림 혹은 서브스트링으로서 일치태그가 포함되어 있을 때 "일치"가 발생한다. 경우에 따라서는, 인식된 서브스트링의 유일한 형태가 "우측에 와일드카드" 서브스트링으로서 "단어*"의 형태를 갖는다. 도 2-3으로부터 알 수 있듯이, 번호 3 문서는 실제로는 일치목록(60) 중에 일치태그인, "보드(board)"와 직접 일치한다. 번호 4 내지 8의 문서들은 "보드"라는 단어를 직접적으로 포함하지 않지만, 이들은 그 단어를 포함하는 문서/노드로부터 차일드 문서/노드이다. 이들은 그 단어를 포함하는 패어런트로부터 차일드들이기 때문에, 따라서 이들은 간접일치이다.

일치목록(60)은 제3 일치 레코드인 '널(null)'을 가지며, 이것은 단순히 일치목록의 끝을 나타낸다. 목록 끝에 널 어를 사용하는 것은 공지된 콤퓨팅 기술이며 여기 기술된 특정한 것 대신으로 많은 다른 목록 처리 기술을 사용할 수 있다.

일치 목록 내의 직접 일치 레코드는 단어색인(22)으로부터 온다. 간접 일치 레코드는 문서 저장부(20) 내의 문서 레코드나, 각각의 직접 일치에 대해 어레이(49) 내 문서 요약 레코드를 검토하여 얻어진다. 문서가 서브트리를 갖는다는 것을 직접 일치 문서 레코드가 나타내는 경우, 서브트리 내 문서 범위에 대해 간접 일치 레코드가 생성된다. 서브트리 내 문서가 또한 직접 일치인 경우엔, 간접 일치범위(2개의 간접 일치 레코드에 대해 범위가 분할될 수 있음)로부터 제외된다. 각각의 직접 일치가 일치 목록에 부가될 때, 일치목록 상에 이미 간접일치(범위)가 직접일치와 겹치는지 판정하기 위해서 일치목록을 체크한다. 이것은 일치어가 선조문서 및 후손문서에 모두 나타나는 경우에 발생한다. 선조문서가 이들의 후손보다 낮은 문서번호를 갖기 때문에, 선조문서가 먼저 처리된다. 선조문서에 대해 직접 일치 레코드가 생성되고 직접 일치 문서 이하의 일군의 후손문서에 대해 간접 일치 레코드가 생성된다. 후손 간접 일치문서가 처리될 때, 이 역시 직접일치 레코드에 목록될 것이며, 따라서 간접일치 범위에 포함되지 않게 된다. 단지 한 범위의 문서들만 관련된 각각의 간접 일치를 유지하기 위해서, 간접 일치 레코드는 후손 직접 일치 문서를 제외한 이의 각 측에 하나씩의 2개의 간접 일치 레코드로 분할된다. 물론, 후손문서가 간접 범위의 한 경계에 있다면, 단지 하나의 새로운 간접 일치 레코드만이 생성될 것이다. 이 하나의 새로운 간접 일치 레코드는 단순히 경계에서 한 문서 번호만큼 감소된 간접 범위가 될 것이다.

이것을 도 4(b)-(c)에 도시하였다. 도 4(b)에 도시한 일치 레코드(62)는 문서 12에 대한 직접 일치 레코드와, 번호 13 내지 17 문서에 대해 간접 일치 레코드를 갖는다. 번호 15 및 16의 문서가 직접적으로 일치태그를 포함하게 변경되었다면, 서브트리는 직접일치의 각 측에 하나씩의 2개의 간접일치로 표현될 것이다. 물론, 직접일치의 일측에 일치가 없다면, 단지 하나의 간접일치 레코드만이 필요하게 될 것이다.

여기서, 문서범위 중 하나인 "17-17"은 단지 하나의 문서만을 포함한다. 이것은 간접 일치 레코드와 직접 일치 레코드를 구별하는 것이다. 물론, 대안 구성을 사용할 수 있다. 예를 들면, 단순한 경우로, 문서 번호가 직접 혹은 간접 일치를 언급하는지 여부를 나타내는 경우, 각각의 일치 레코드는 플래그 및 문서번호만을 포함할 수 있을 것이다. 직접일치에 있어서, 단일의 번호는 일치하는 문서의 번호가 될 것이며, 간접 일치들인 경우, 번호는 범위의 마지막 문서의 번호이다. 이것은 범위의 첫 번째 번호가 바로 전의 직접일치 레코드의 번호 다음의 번호인 경우 모호하지 않다. 이것은 문서들이 이러한 특성을 갖도록 정렬되어 있기 때문에 문서번호가 빠지지 않는다면 그 경우가 될 것이다.

도 3에서, 어떤 시스템에선, 문서가 얼마나 자주 변경되고 탐색어가 얼마나 자주 사용되는지에 따라, 직접일치 및 간접일치를 미리 계산하여 레코드(52)에 저장해 둘 수 있다. 아니면, 이들은 필요할 때 도중에서 생성된다. 이것이 행해지면, 탐색엔진(36)은 단어색인(22)으로부터 얻어질 수 없는 문서 혹은 다른 필드정보 내의 단어들의 위치의 검사를 탐색질의가 필요로 하지 않는다면 문서 저장부(20)를 액세스할 필요가 없다. 단어색인(22)을 사용하는 이점은 탐색요구가 있을 때 신속한 탐색을 위해 정돈되어 있다는 것이다.

일단 위치목록이 탐색엔진(36)에 의해 얻어지거나 생성되면, 탐색엔진은 출력목록(31)을 내준다. 출력목록(31)은 목록된 문서 자체일 수 있거나, 혹은 단지 문서번호일 수 잇다. 출력목록(31)이 문서 자체이고, 적합하게 형식화되어 있다면, 출력목록(31)은 브라우저(12)로 보내지는 탐색결과(32)가 될 수 있다(도 1 참조).

도 5는 "The game of go"라는 질의 스트링으로부터 나올 수 있는 탐색결과의 디스플레이(53)의 예를 도시한 것이다. 디스플레이(53) 상에, 일치하는 카테고리 문서(54)는 위에 도시되었고, 일치하는 사이트 문서(58)는 이와는 분리되어 카테고리 트리를 통한 이들의 경로(56)와 함께 도시되었다. 도 5는 카테고리 구조 및 본 발명의 양수인인 야후!, 인크.의 사이트 목록을 통한 실제 탐색을 나타낸다. 명확하게 하기 위해서, 도 5에는 도시한 모든 일치가 다 다른 도면에는 나타내지 않았고 발견된 실제 177개의 사이트 일치 모두를 다 도 5에 도시하지는 않았다.

디스플레이(53)의 몇 가지 쉬운 사용 특징을 언급한다. "히트" 혹은 일치(54, 58) 각각은 일치된 곳으로 경로를 정하는 카테고리의 타이틀의 연결과 함께 도시되었다. 더 읽기 쉽게 하기 위해서, 일치하는 문서로부터 차일드 노드들인 일치하는 문서는 도시하지 않았다. 이들이 있다면, 일치하는 카테고리 문서(54) 밑의 모든 레코드들이 나타나게 될 것이다.

직접 및 간접 일치 레코드를 갖는 일치 목록, 및 이들의 생성에 대해 기술하였는데, 다른 일치목록을 형성하기 위해서 일치목록에 "AND", "OR", "ADD", "SUBSTRACT"와 같은 연산을 적용하는 것에 대해 기술한다. 이들 연산은 탐색 부제어 각각에 대해 일치목록을 얻을 수 있는 경우 복수의 탐색 부제어를 포함하는 탐색엔진이 복잡한 탐색표현식에 대한 일치목록을 발생할 필요가 있는 경우에 유용하다. 전술한 예를 계속하여, 탐색엔진은 AND 연산자를 사용하여 "go" 및 "game" 탐색어에 대한 일치목록들을 결합하여 "go AND game" 탐색표현식에 대한 일치목록(혹은 문서목록)에 이를 수 있다.

도 6은 2개 이상의 일치목록을 "AND"하여 새로운 일치목록을 발생하는 처리 흐름도이다. 명백하겠지만, 결과적인 일치목록은 탐색결과를 발생하는데 사용될 수 있으며, 혹은 일치목록 상의 후속되는 논리연산에 입력으로서 사용될 수 있다. 후속되는 논리연산이 행해지지 않을 예정이면, 출력은 단순히 문서들의 목록일 수 있을 것이다. 도 6에서, 단계에 S1, S2, 등을 표기하였고 일반적으로 단계의 실행 순서를 나타낸다. 본 설명을 읽어 명백하게 되겠지만, 단계들에 대한 다른 구성이 동일한 기능을 수행하여 동일한 결과를 달성할 수도 있다.

2개 이상의 일치목록을 "AND"할 필요성은 탐색 스트링이 "표현_A AND 표현_B AND..." 형태의 표현식을 포함하는 경우 일어날 수 있다. 하나의 일치목록은 그 부제어를 포함하는 문서를 나타내는 "표현_A"에 대해서, "표현_B"에 대해선 또 다른 것, 등이 얻어진다. 결과적인 일치목록은 "AND"된 모든 표현식들을 포함하며 탐색 부제어 중 적어도 하나는 직접적으로 포함하는 모든 문서의 목록이다. 이들 요건의 다른 변경예는 본 명세서를 읽은 후에 명백하게 될 본 처리의 수정에 의해서 다루어질 수 있다는 것이 명백할 것이다.

광의로, 도 6에 기술된 처리는 모든 일치목록에서 어느 문서가 발견되고 적어도 하나의 직접 일치 레코드에서 어느 문서가 발견될 것인가 찾기 위해서 복수의 일치목록을 정밀조사하는 유용한 처리이다. 이것을 행하기 위해서, 처리는 하나의 일치목록에서 직접 일치 레코드를 찾는 단계와 이어서 이들 목록에서 문서가 발견되는지 여부를 판정하기 위해 모든 다른 일치목록을 체크하는 단계를 포함한다. 하나의 일치목록이 그에 대한 문서를 갖지 않는 것으로 발견되었을 때, 문서 커서는 일치목록 내의 그 다음 문서로 증분된다.

도 6에서, 처리변수는 단계 S1에서 초기화된다. 루프 카운터인 LOOP_CNT는 제로로 초기화된다. 루프 카운터를 사용하는 것에 대해 이하 설명한다. 또한, 일치목록 중에 문서들을 가리키는 문서 커서(D_CUR)는 1로 설정되고 발견된 일치 개

수를 세는 수집 카운터(COLL_CNT)는 제로로 설정되며, 일치목록 중 하나를 가리키는 목록 포인터(L_PTR)는 일치목록 중 하나를 가리키도록 설정된다. L_PTR은 그럴 필요는 없으나, "AND"되는 제1 목록된 부제어에 대한 일치목록을 가리키게 할 수 있다. L_PTR이 가리키는 일치목록을 여기선 "현재 일치목록" 혹은 "현재 일치"라고 칭한다.

단계 S2에서, 현재의 일치 레코드는 현재의 목록으로부터 얻어진다. 이것을 여기선 "현재 일치 레코드"라 칭한다. 이 현재 일치 레코드는 가장 낮은 문서번호가 D_CUR 이상인 현재목록 내의 일치목록이다. 현재 일치목록이 비어있다면, 아무 문서도 발견되지 않을 것이므로, 처리는 단순히 종료된다. 단계 S2까지 1회에 D_CUR은 1로 될 것이므로 현재의 일치 레코드는 현재의 일치목록 내에 제1 목록이 될 것이다.

단계 S3에서, 현재 일치 레코드가 직접 일치인지 아니면 그룹 일치인지 판정하기 위해 현재 일치 레코드를 체크한다. 도 4에서 명백한 바와 같이, 간접, 혹은 그룹, 일치들은 하나 이상의 문서 번호들의 범위들로서 표현되기 때문에, 일치 레코드가 직접 일치인지 여부를 판정하는 것은 간단하다. 현재 일치 레코드가 직접 일치이면, 이것은 현재 일치 레코드로 나타내어진 문서에 대해 적어도 하나의 직접 일치가 있다는 요건을 만족한다. 제1 직접일치가 발견되기 전에, COLL_CNT는 제로이며, 이것은 처리가 일치목록들로부터 문서를 "수집하는 것"을 아직 시작하지 않았음을 나타낸다. COLL_CNT가 단계 S4에서 제로이고 직접 일치 레코드가 발견되면, COLL_CNT는 직접 일치가 발견되었음을 나타내도록 1로 설정된다(S5).

일단 제1 직접 일치가 발견되면, LOOP_CNT는 단계 S6에서 제로로 재설정되고(LOOP_CNT는 어떤 상황에서 발생할 수 있을 무한루프를 방지하는데 사용됨; 이의 사용에 대해선 후술한다), L_PTR은 다음 일치목록을 가리키도록 회전된다(S7). 이에 이어, 처리는 다시 단계 S2로 돌아가 다음 일치목록을 현재의 일치목록으로 한다. 단계 S2에서, 현재의 일치 레코드는 지금 현재의 일치목록 내에서 발견된다. 전술한 바와 같이, 발견된 레코드는 가장 낮은 문서번호가 D_CUR 이상의 것이고, 아무 것도 발견되지 않으면 처리는 종료한다.

COLL_CNT가 S4에서 제로가 아니면, 이것은 문서 수집 상태를 나타내는 것으로, 현재 일치 레코드의 문서번호에 대해서 D_CUR을 체크한다(S8). 이들이 동일하다면, 이것은 이전 일치목록으로부터 수집된 문서가 지금 현재의 일치목록에 대한 것과 동일하다는 것을 가리킨다. 이 경우에 해당하면, COLL_CNT는 증분되어(S9) 또 다른 일치가 발견되었음을 나타내게 된다. 모든 N 일치 목록 미만이 처리되었다면, COLL_CNT는 N 미만이 될 것이며, 따라서 처리는 단계 S6/S7로 계속되어 여기서 다음 일치목록은 현재의 일치목록으로 된다. 이것은 COLL_CNT가 N과 같게 될 때까지 계속될 수 있다.

COLL_CNT가 N이 될 때, 이것은 모든 N개의 일치목록 중에서 D_CUR과 동일한 문서 번호가 발견되었으며, 따라서 이 문서번호는 출력 일치목록 내에 있는 문서번호임을 의미한다. 결국, 현재의 일치 레코드는 출력(S10)이며, 처리는 단계 S6/S7로 계속된다(비록 처리는 현재 일치목록을 변경함이 없이 단계 S2로 되돌아가 계속될 수도 있다하더라도). 단계 S10에서, COLL_CNT는 문서번호 탐색의 다음 사이클 동안 제로로 리셋된다.

단계 S8에서, 현재의 일치 레코드의 문서번호가 D_CUR과 같지 않다면, 이것은 현재 일치목록이 D_CUR과 동일한 문서번호를 갖는 일치 레코드를 갖지 않았으며 더 큰 문서번호가 선택되었기 때문이다. 이 경우, 현재의 문서가 여전히 수집되고 있는 중이나, 이것은 새로운 더 큰 문서번호이다. D_CUR은 그 새로운 문서번호로 설정된다(S11). 얼마나 많은 일치목록이 이 새로운 문서번호를 갖는가를 기억하기 위해서, COLL_CNT는 1로 리셋되고(S5) 처리는 전술한 바와 같이 계속된다.

단계 S3에서, 현재의 일치 레코드가 직접 일치 대신 그룹 일치이면, 레코드 처리는 처리 상태, 즉 문서가 "수집"되고 있는지 여부에 따른다. 이것은 COLL_CNT를 체크하여 판정된다(S12). COLL_CNT가 제로가 아니면, 문서는 수집되고 있는 것으로, 이 경우 현재 일치 레코드는 D_CUR과 비교된다(S13). D_CUR이 현재 일치 레코드의 범위 내에 있다면(이 단계로 되기 위해서 그룹 레코드이어야 함), COLL_CNT가 증분되고(S9) 다음 목록이 전술한 바와 같이 체크된다.

단계 S12에서, COLL_CNT가 제로이면, 처리는 단계 S14에서 계속된다. 또한, 단계 S13에서, D_CUR이 일치 레코드의 범위 내에 있지 않다면, 처리는 어떠한 문서도 수집되고 있지 않음을 나타내도록 COLL_CNT를 제로로 설정한 후에 단계 S14로 계속된다. 단계 S14에서, LOOP_CNT는 증분되어 N과 비교된다(S15). LOOP_CNT가 N이 아니면, 처리는 단계 S7에서 다음 목록으로 계속된다. LOOP_CNT가 N이면, 이것은 모든 N개의 목록이 검토되었고 각각에서 일치가 발견되어 있으나, 일치의 어느 것도 직접 일치가 아니었음을 나타내며, 그렇지 않다면 COLL_CNT는 제로가 되지 않게 될 것이다.

LOOP_CNT가 N이면, 이것은 D_CUR을 포함하는 그룹(간접) 일치 레코드가 일치 목록 각각에서 발견되었으며 따라서 D_CUR에 대해 어떠한 직접일치도 없다는 것을 의미한다. 가장 낮은 끝에 있는 문서번호를 갖는 그룹을 찾기 위해서 D_CUR을 포함하는 각각의 그룹을 조사한다. 대안으로, 탐색엔진은 각각의 일치목록이 조사될 때 가장 낮은 끝 문서번호를 유지하고 있을 수 있다. D_CUR은 가장 낮은 끝 문서번호보다 하나 큰 번호로 설정되고(S16) 문서 탐색이 단계 S6에서 계속되며, 이 단계에서 LOOP_CNT는 제로로 설정된다. 단계 S6은 직접일치가 발견될 때 LOOP_CNT를 리셋하기 위해서 놓인

것으로 출력 레코드가 출력되거나 LOOP_CNT가 N으로 되어 D_CUR이 현재 그룹의 끝을 지나 이동됨으로써 무한 루프가 회피된다. 바람직한 실시예에서, 적어도 하나의 직접일치가 요구된다. 그러나, 직접일치가 요구되지 않는 실시예에서, 처리는 LOOP_CNT가 N이 이를 때 일치 레코드를 출력할 수도 있다.

이 처리에 이어 이의 종결로, 단계 S2를 거쳐 현재 일치목록의 끝에 도달하였을 때, AND 연산의 요건을 충족하는 문서에 대한 일치 레코드는 단계 S10을 거칠 때 출력되었을 것이다. 대안으로, 다른 논리연산도 행해지지 않을 것이면, 출력은 일치하는 문서의 문서번호의 목록이 될 수도 있을 것이다.

도 7은 "A OR B OR ..." 형태의 탐색 표현식과 일치하는 문서의 출력목록을, 부제어 A, B, 등에 대한 일치목록들로부터 생성하는 처리를 도시한 것이다. 광의로, 이 처리는 탐색 표현식을 이의 부제어들로 분해하고 각각의 부제어에 대한 일치목록을 확인한 후, 출력목록 상의 각각의 문서가 부제어 중 적어도 하나를 포함하는 출력목록으로 일치목록들을 결합하는 것을 포함한다.

바람직한 실시예에서, 출력목록 상의 각각의 문서가 적어도 하나의 직접일치를 갖는 부가적인 요건이 부가되어, 이 부가요건을 충족하는 문서가 반드시 OR 표현식을 직접적으로 일치해야 할 것이기 때문에 간접 일치는 전혀 없게 될 것이다. 바람직한 실시예에서, 출력목록은 각각의 관련 일치 카운트를 갖는 직접일치들의 목록이다. 일치 카운트는 OR 부제어 중 얼마나 많이 직접적으로 혹은 간접적으로 일치되는가를 나타내며, 따라서 특정문서의 상대적 관련성에 대한 표시가 된다.

도 7의 흐름도에서, 처리단계는 S30, S31 등으로 표기되었고 표시된 곳을 제외하고 숫자상의 순서로 실행된다. 처리는 단계 S30에서 시작하고 이 단계에선 탐색 표현식으로부터 부제어들을 추출하고, 일치목록 카운트 N은 부제어 개수로 설정된다. 단계 S31에서, 하나의 일치목록이 각각의 부제어에 대해 발생되며, 혹은 일치목록들이 이미 존재하는 목록들이면 이들을 불러들인다.

단계 S32에서, 제1 문서의 연관된 목록에서 이 제1 문서를 가리키는 커서를 각 일치목록에 대해 초기화한다. 단계 S33에서 각각의 목록으로부터 제1 문서를 N-멤버 힙에 부가한다.

다음에, 힙 내용은 문서번호로 정렬된다(S34). 바람직한 실시예에서, 힙이 특정 문서번호에 대한 직접일치와 그 동일 문서번호에서 시작하는 한 범위와 간접 일치를 포함하는 경우, 직접일치는 간접 일치 앞에 정렬된다. 힙이 같은 시작 문서번호를 갖는 하나 이상의 간접일치를 갖는다면, 이들은 이들의 끝 문서번호로 소트된다.

일단 힙이 소트되면, 최상위 힙 항목이 힙으로부터 제거된다(S35). 최상위 힙 항목이 직접일치 항목이고 그 직접 일치항목의 문서번호가 이미 출력목록 내에 없다면, 이것은 출력목록에 부가되고(S36) 처리는 바로 지금 제거된 항목의 일치목록으로부터 힙에 또 다른 항목을 부가함으로써 계속된다(S37). 일치목록이 더 이상의 항목을 갖고 있지 않다면, 어떠한 새로운 항목도 힙에 부가되지 않는다. 결국, 힙은 완전히 비어 있게 될 것이다. 단계 S38에서 힙이 비어있다면 처리가 행해져 종료된다. 그렇지 않다면, 처리는 다시 단계 S34로 복귀하여 여기서 힙이 다시 정렬된다.

제거된 항목이 출력목록 상에 이미 있는 문서의 문서번호를 가진 직접일치 레코드이면, 그 문서번호에 대한 일치 카운트가 증분되고(S39) 처리는 전술한 바와 같이 단계 S34로 계속된다. 제거된 항목이 간접 일치 레코드이면, 이것은 출력 목록에 배치되지 않고 간접일치 레코드의 문서범위 내에 있는 각각의 문서에 대해 일치 카운트가 증분되고(S39) 처리는 단계 S34으로 계속된다. 적어도 하나의 직접일치를 가져야 한다는 요건을 충족하는 간접 일치에 대한 문서범위 내에 어느 문서든지 이미 출력목록 상에 있게 될 것이므로 그 간접일치는 출력목록에 부가되지 않는다. 이것은 일치 레코드가 힙 상에 배치될 때 이들 일치 레코드는 이들의 일치목록으로부터 순서대로 취해지고 힙 상의 항목들이 또한 순서대로 제거되며, 직접일치는 동일한 번호에서 시작하는 간접일치들 앞에서 취해지기 때문이다.

힙이 비어있을 때, 출력목록은 OR 기준에 일치하는 모든 문서들을 포함할 것이다. 모든 출력 리스트 엔트리는 직접일치가 될 것이며 관련된 일치 카운트를 갖게 될 것이다. 각각의 일치가 적어도 하나의 부제어를 직접 포함해야 하는 요건이 부가되지 않는다면, 출력목록은 다음 처리에 적합한 일치목록 형태로 있을 수도 있다. 단독으로 혹은 중요도 가중으로, 관련성에 따른 문제를 정돈하도록 일치 카운트를 사용할 수 있다.

"ADD" 및 "SUBTRACT" 연산에 대해서, 이들은 훨씬 간단하다. "ADD" 연산에 대해서, 목록에 부가될 문서번호가 단순하게 삽입된다. 물론, 직접일치의 문서번호를 포함하는 간접의 그룹일치를 포함하는 목록에 직접일치가 부가될 것이면, 그룹일치 레코드는 전술한 바와 같이 분할된다. "SUBTRACT" 연산에 있어서, 일치 레코드는 단순히 일치목록으로부터 삭제된다. 문서가 그룹 일치 레코드의 범위 내에 있는 경우 문서번호가 감분된다면, 그룹 일치 레코드는 전술한 바와 같이 분할된다.

전술한 바는 예시적인 것이고 제한적인 것이 아니다. 개시된 바를 살펴보았을 때 이 기술에 숙련된 자들에서 본 발명의 많은 변형이 명백하게 될 것이다. 예를 들면, 문서의 계층구조는 전술한 계층 탐색구조 대신 인터넷 상의 문서들의 웹일 될 수도 있을 것이다. 그러므로, 본 발명의 범위는 상기 설명에 관련하여 정해지지 않으며 첨부된 청구범위의 전 범위의 등가물과 함께 이들 청구범위에 관련하여 정해지는 것이다.

(57) 청구의 범위

청구항 1.

문서 저장부에 저장되어 있고, 탐색가능 요소들을 포함하며 문서계층으로 편제되어 있는 문서들을 탐색하는 방법에 있어서,

한 세트의 탐색가능 요소들을 기술하는 논리함수인 탐색 표현식을 탐색엔진에 제공하는 단계;

직접일치 혹은 간접일치를 탐색하는 단계로서, 직접일치는 상기 탐색 표현식과 일치하는 문서이며, 간접일치는 계층에서 간접적으로 일치하는 문서의 선조문서들의 내용과 상기 간접적으로 일치하는 문서의 내용을 결합하였을 경우에만 상기 탐색 표현식에 일치하는 문서인, 탐색 단계;

상기 탐색 단계의 결과에 대하여, 복수 문서에 대해서, 복수의 문서들을 링크하는 계층 내의 경로로서 표현되는 적어도 하나의 일치의 목록을 발생하는 단계; 및

탐색결과로서 상기 목록을 출력하는 단계를 포함하는 것을 특징으로 하는 문서 탐색방법.

청구항 2.

제1항에 있어서, 상기 탐색가능 요소들은 단어 및 문서들이며 적어도 일부 텍스트를 포함하는 것을 특징으로 하는 문서 탐색방법.

청구항 3.

삭제

청구항 4.

제1항에 있어서, 상기 탐색단계는 요소 색인에서 상기 탐색 표현식의 구성요소를 탐색하는 단계를 포함하는 것을 특징으로 하는 문서 탐색방법.

청구항 5.

제1항에 있어서, 상기 문서계층은 계층 트리이고, 상기 방법은, 상기 계층 트리의 임의의 브랜치 내의 문서번호들이 연속하게, 상기 계층 트리 내의 각각의 문서에 상기 문서번호를 할당하는 단계를 더 포함하는 것을 특징으로 하는 문서 탐색방법.

청구항 6.

제1항에 있어서, 상기 탐색 표현식은 오퍼랜드와 연산자를 포함하는 식이며, 상기 오퍼랜드는 명시된 탐색가능 요소 혹은 와일드 카드를 포함하며 상기 연산자는 AND, OR, ADD 혹은 MINUS를 포함하는 것을 특징으로 하는 문서 탐색방법.

청구항 7.

제1항에 있어서, 문서는 특정 주제를 나타내는 스트링이며 상기 계층은 주제들의 계층인 것을 특징으로 하는 문서 탐색방법.

청구항 8.

계층 데이터를 효율적으로 저장하고 탐색하는 방법에 있어서,

계층 내 위치를 가지며 이 위치 위엔 선조 데이터 요소를, 혹은 이 위치 밑으로는 후손 데이터 요소를 혹은 이들 모두를 갖는 각각의 데이터 요소를 상기 계층 내에 편제하는 단계;

상기 데이터 요소의 번호를 임의의 선조 데이터 요소의 번호보다 크게 하고, 상기 선조 데이터 요소의 후손이 아니며 상기 선조 데이터 요소 번호보다 큰 데이터 요소 번호를 갖는 임의의 다른 데이터 요소의 번호 미만이 되게 각각의 데이터 요소에 데이터 요소 번호를 할당하는 단계; 및

데이터 요소 내용이 단독으로 탐색 표현식과 일치할 때 상기 탐색 표현식은 직접 일치되는 것이고, 데이터 요소가 직접 일치하지 않으나, 데이터 요소의 내용 및 적어도 하나의 선조 데이터 요소가 함께 탐색 표현식과 일치할 때 상기 탐색 표현식은 간접적으로 일치되는 것으로서, 탐색 표현식에 직접 혹은 간접적으로 일치하는 데이터 요소들을 식별하기 위해서 상기 계층에 탐색 표현식을 적용하는 단계를 포함하는 것을 특징으로 하는 계층 데이터 저장 및 탐색방법.

청구항 9.

문서 저장부에 저장되어 있고, 탐색가능 요소들을 포함하는 문서들을 탐색하는 방법에 있어서,

문서들을 문서 계층으로 수작업으로 편제하는 단계;

한 세트의 탐색가능 요소들을 기술하는 논리함수인 탐색 표현식을 탐색 엔진에 제공하는 단계;

복수 문서에 대해서, 복수의 문서들을 링크하는 계층 내의 경로로서 표현되는 적어도 하나의 일치의 목록을 탐색 단계의 결과들로부터 발생하는 단계; 및

탐색결과로서 상기 목록을 출력하는 단계를 포함하는 것을 특징으로 하는 문서 탐색 방법.

청구항 10.

제9항에 있어서, 직접일치 혹은 간접일치를 탐색하는 단계를 더 포함하며, 직접일치는 상기 탐색 표현식과 일치하는 문서이며, 간접일치는 계층에서 간접적으로 일치하는 문서의 선조문서들의 내용과 상기 간접적으로 일치하는 문서의 내용을 결합하였을 경우에만 상기 탐색 표현식에 일치하는 문서인 것을 특징으로 하는 문서 탐색방법.

청구항 11.

제9항에 있어서, 상기 각 문서는 카테고리와 사이트 참조 중 하나를 나타내는 것을 특징으로 하는 문서 탐색방법.

청구항 12.

제9항에 있어서, 문서가 카테고리를 나타내면, 상기 문서는 타이틀을 포함하고, 상기 문서가 사이트 참조를 나타내면, 상기 문서는 타이틀과 URL(Uniform Resource Locator) 중 하나를 포함하는 것을 특징으로 하는 문서 탐색방법.

청구항 13.

각 데이터 요소가 계층 내 위치를 가지고, 이 위치 위엔 하나 이상의 선조 데이터 요소를, 또는 이 위치 밑으로는 하나 이상의 후손 데이터 요소를, 또는 이들 모두를 갖도록 하는 트리 구조를 가지는 상기 계층으로 상기 데이터 요소들을 편제하는 단계; 및

데이터 요소 내용이 단독으로 탐색 표현식과 일치할 때 상기 탐색 표현식은 직접 일치되는 것이고, 데이터 요소가 직접 일치하지 않으나, 데이터 요소의 내용 및 적어도 하나의 선조 데이터 요소가 함께 탐색 표현식과 일치할 때 상기 탐색 표현식은 간접적으로 일치되는 것으로서, 탐색 표현식에 직접 혹은 간접적으로 일치하는 데이터 요소들을 식별하기 위해서 상기 계층에 탐색 표현식을 적용하는 단계를 포함하는 것을 특징으로 하는 계층 데이터 저장 및 탐색방법.

청구항 14.

제13항에 있어서, 상기 데이터 요소들을 편제하는 상기 단계는 편집부원에 의하여 수작업으로 행해지는 것을 특징으로 하는 데이터 저장 및 탐색방법.

청구항 15.

제13항에 있어서, 상기 각 데이터 요소는 카테고리와 사이트 참조 중 하나를 나타내는 것을 특징으로 하는 데이터 저장 및 탐색방법.

청구항 16.

제15항에 있어서, 상기 데이터 요소가 사이트 참조이면, 상기 데이터 요소는 상기 계층 내 선조 데이터 요소들만을 가지는 것을 특징으로 하는 데이터 저장 및 탐색방법.

청구항 17.

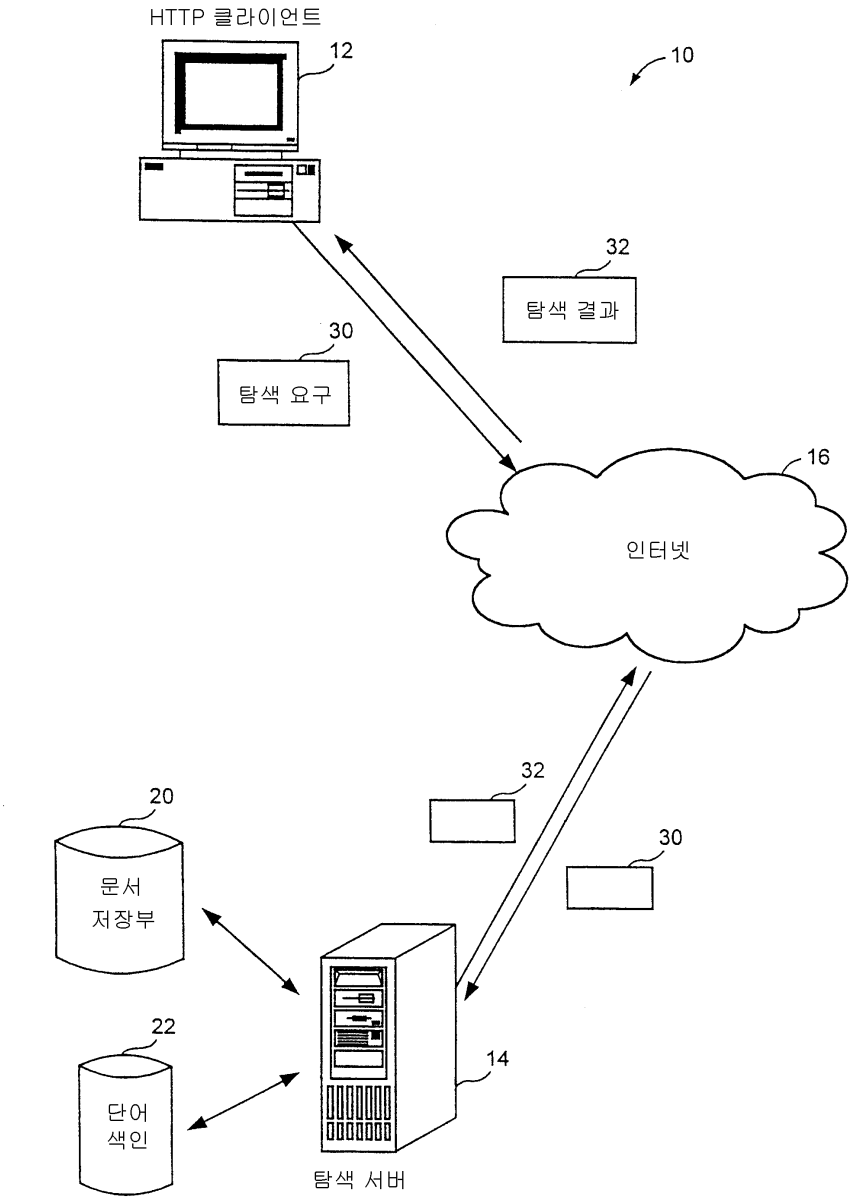
제15항에 있어서, 상기 데이터 요소가 카테고리이면, 상기 데이터 요소는 후손 데이터 요소들로서 하나 이상의 사이트 참조들 또는 하나 이상의 카테고리들 또는 모두를 포함하는 것을 특징으로 하는 데이터 저장 및 탐색방법.

청구항 18.

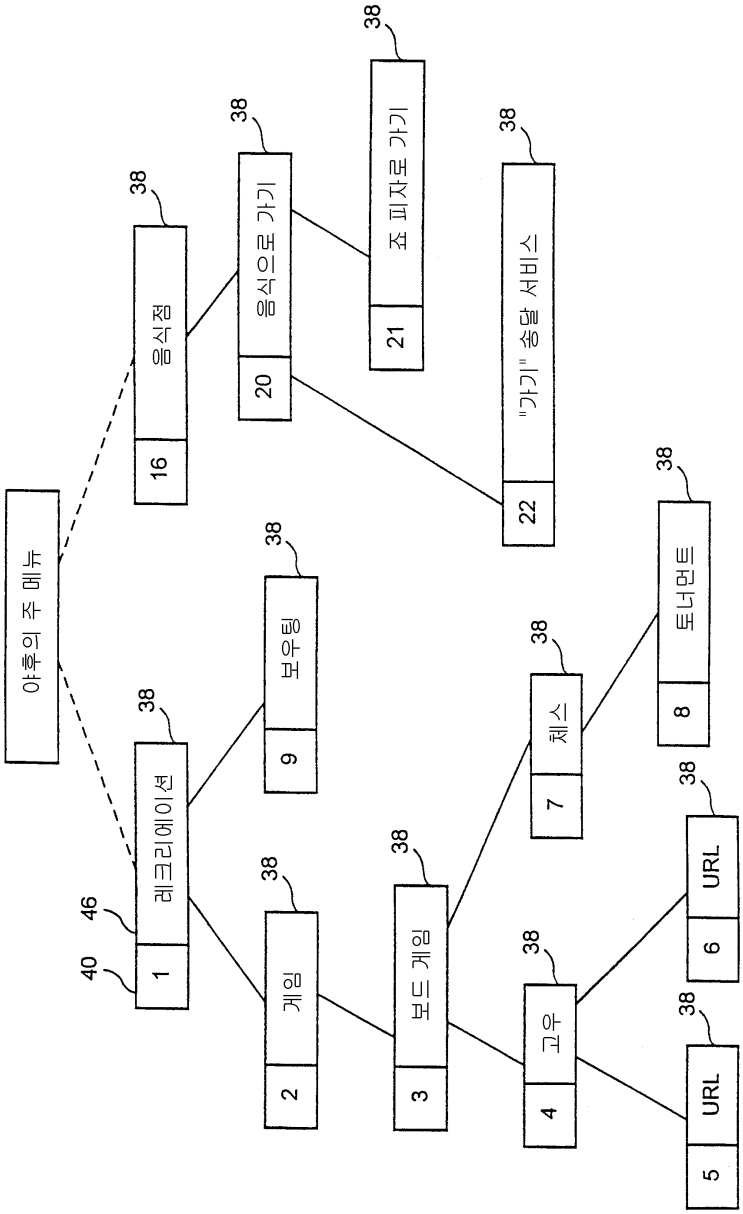
제13항에 있어서, 상기 계층 트리의 임의의 브랜치 내의 데이터 요소 번호들이 연속하도록, 상기 계층 트리 내 각 데이터 요소에 데이터 요소 번호를 할당하는 단계를 더 포함하는 것을 특징으로 하는 데이터 저장 및 탐색방법.

도면

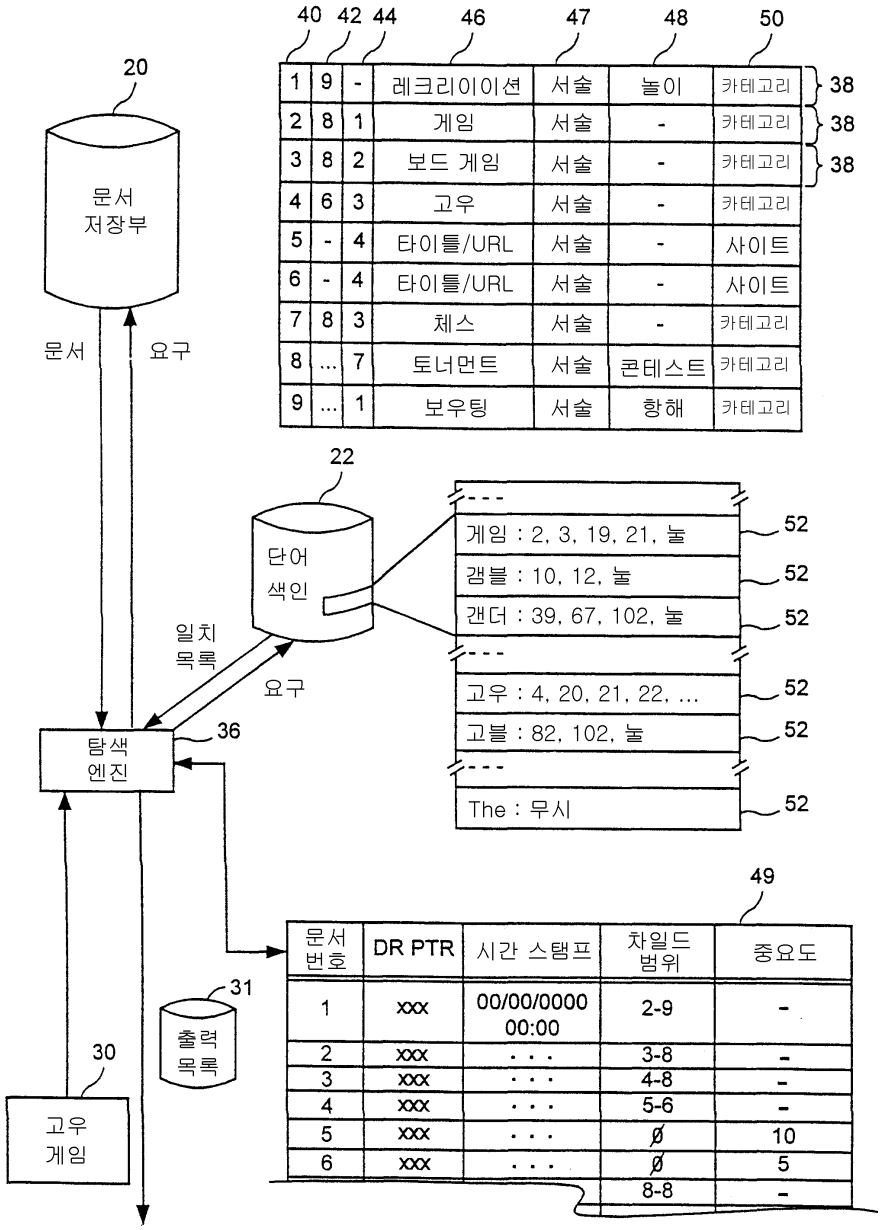
도면1


HTTP 클라이언트
12
10
32
탐색 결과
30
탐색 요구
16
인터넷
32
20
30
문서
저장부
22
14
단어
색인
탐색 서버

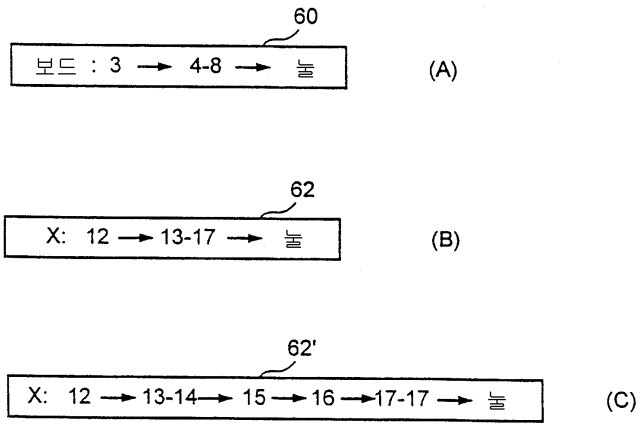
도면2


야후의 주 메뉴
레크리에이션
1
46
40
음식점
16
38
음식으로 가기
20
38
죠 피자로 가기
21
38
"가기" 송달 서비스
22
38
보우팅
9
38
게임
2
38
보드 게임
3
38
체스
7
38
토너먼트
8
38
고우
4
38
URL
6
38
URL
5
38

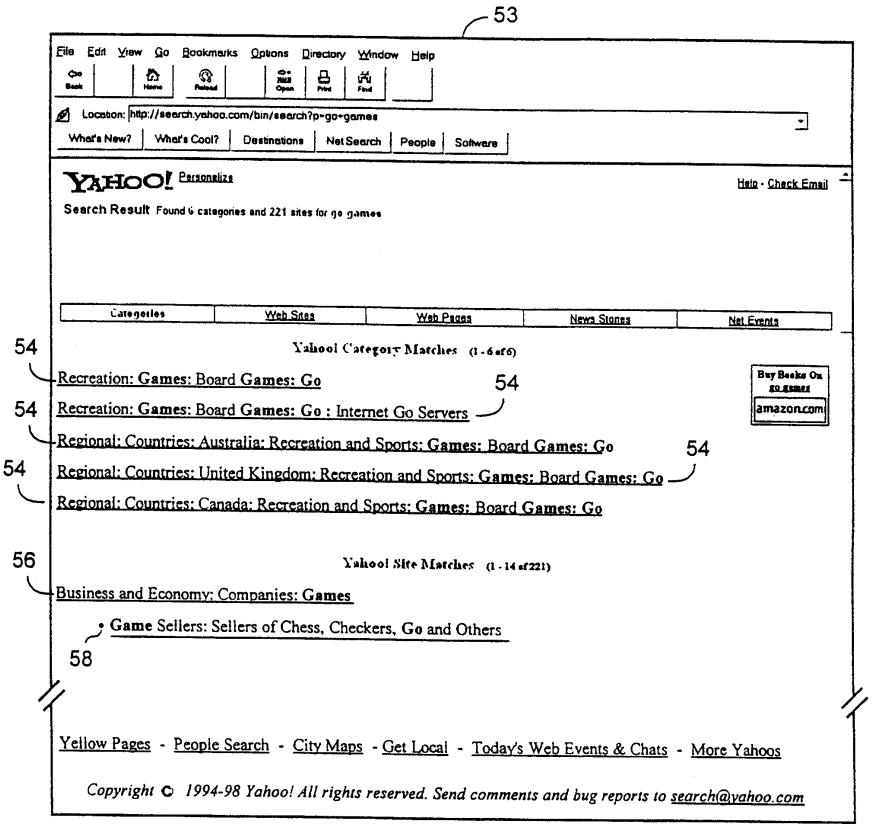
도면3


40 42 44 46 47 48 50
20
문서 저장부
문서
요구
1 9 - 레크리이이션 서술 놀이 카테고리 38
2 8 1 게임 서술 - 카테고리 38
3 8 2 보드 게임 서술 - 카테고리 38
4 6 3 고우 서술 - 카테고리
5 - 4 타이틀/URL 서술 - 사이트
6 - 4 타이틀/URL 서술 - 사이트
7 8 3 체스 서술 - 카테고리
8 ... 7 토너먼트 서술 콘테스트 카테고리
9 ... 1 보우팅 서술 항해 카테고리
22
단어 색인
게임 : 2, 3, 19, 21, 놀 52
갬블 : 10, 12, 놀 52
갠더 : 39, 67, 102, 놀 52
고우 : 4, 20, 21, 22, ... 52
고블 : 82, 102, 놀 52
The : 무시 52
일치 목록
요구
36
탐색 엔진
49
문서 번호 DR PTR 시간 스탬프 차일드 범위 중요도
1 xxx 00/00/0000 00:00 2-9 -
2 xxx · · · 3-8 -
3 xxx · · · 4-8 -
4 xxx · · · 5-6 -
5 xxx · · · ø 10
6 xxx · · · ø 5
8-8 -
31
출력 목록
30
고우 게임

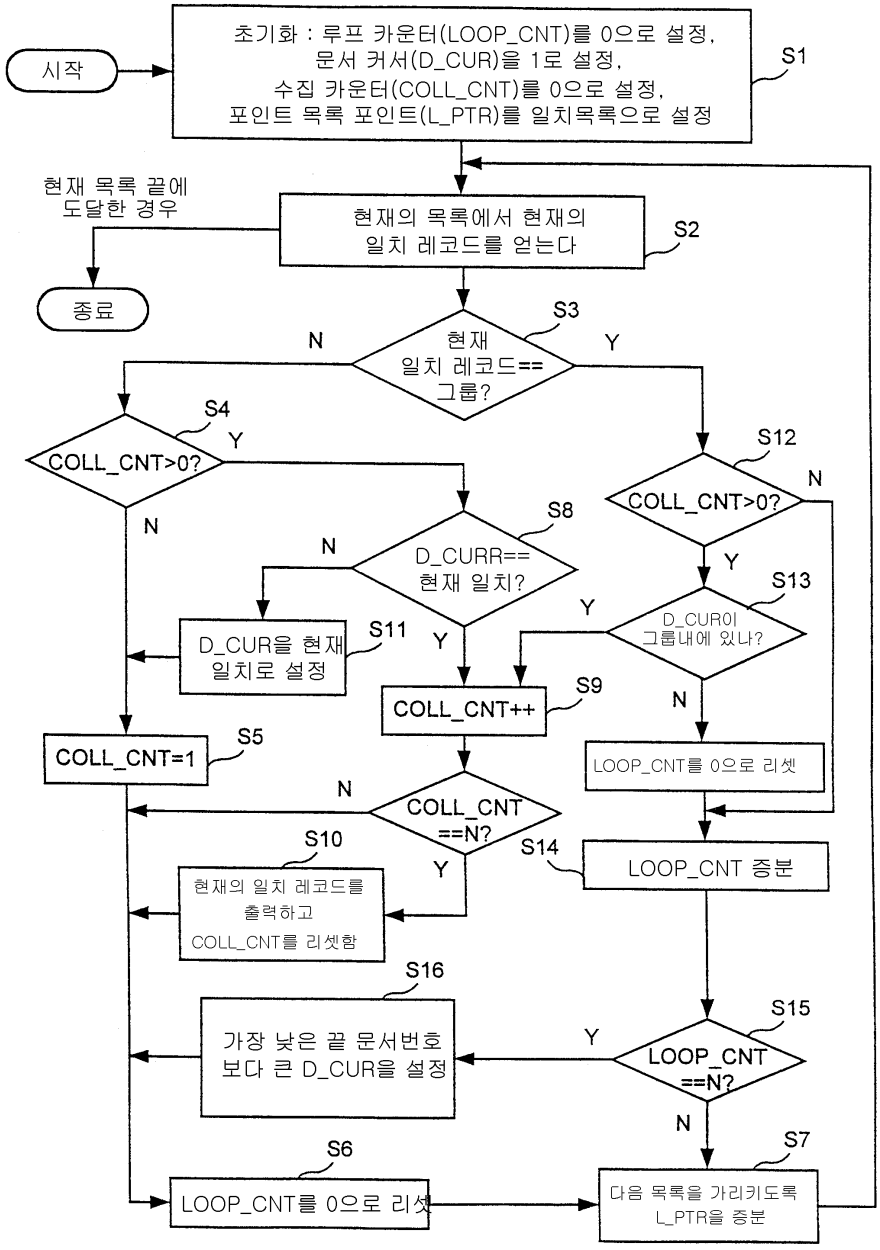
도면4


60
보드 : 3 → 4-8 → 놀
(A)
62
X: 12 → 13-17 → 놀
(B)
62'
X: 12 → 13-14 → 15 → 16 → 17-17 → 놀
(C)

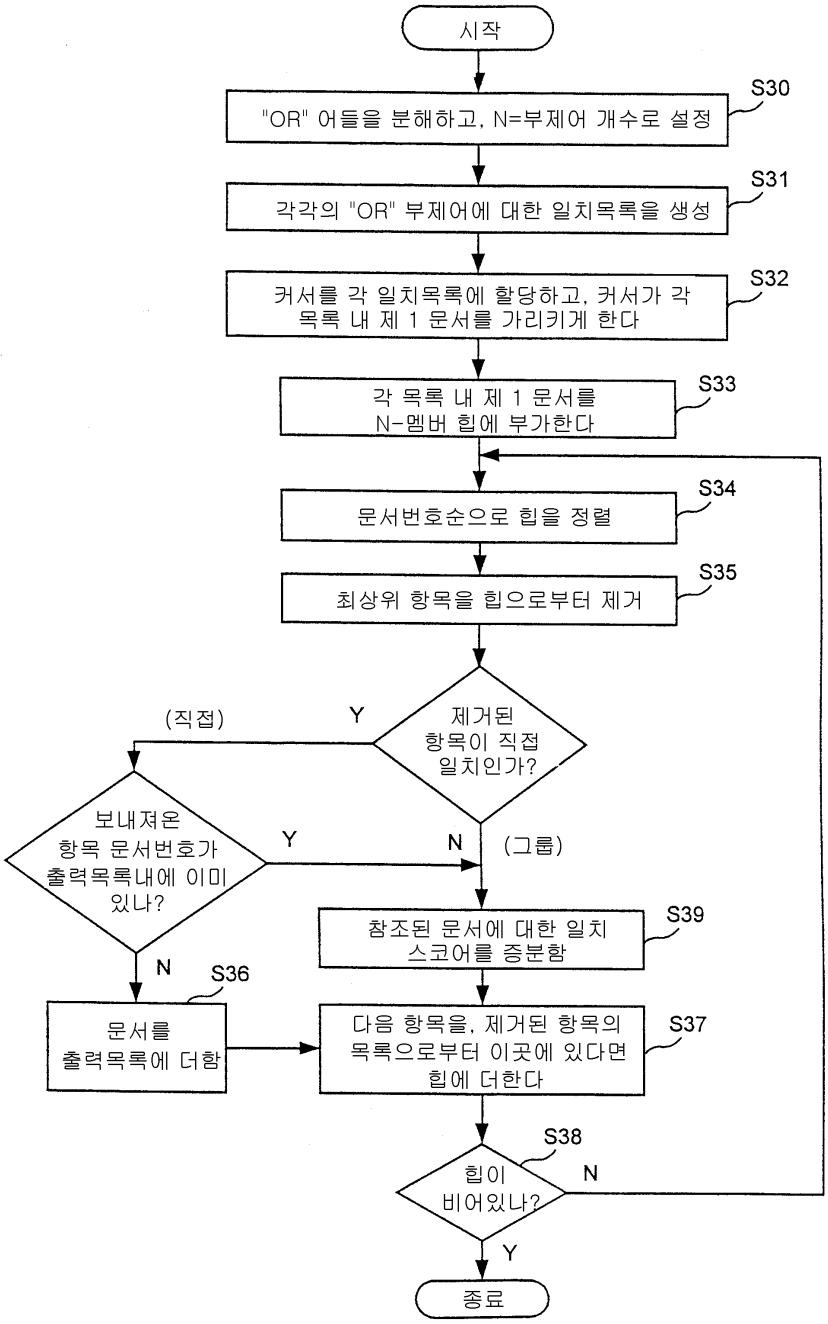
도면5


53
File Edit View Go Bookmarks Options Directory Window Help
Location: http://search.yahoo.com/bin/search?p=go+games
What's New? What's Cool? Destinations Net Search People Software
YAHOO! Personalize
Help - Check Email
Search Result Found 6 categories and 221 sites for go games
Categories
Web Sites
Web Pages
News Stories
Net Events
Yahoo! Category Matches (1 - 6 of 6)
54
Recreation: Games: Board Games: Go
54
Recreation: Games: Board Games: Go : Internet Go Servers
54
Regional: Countries: Australia: Recreation and Sports: Games: Board Games: Go
54
Regional: Countries: United Kingdom: Recreation and Sports: Games: Board Games: Go
54
Regional: Countries: Canada: Recreation and Sports: Games: Board Games: Go
Buy Books On go games
amazon.com
Yahoo! Site Matches (1 - 14 of 221)
56
Business and Economy: Companies: Games
Game Sellers: Sellers of Chess, Checkers, Go and Others
58
Yellow Pages - People Search - City Maps - Get Local - Today's Web Events & Chats - More Yahoos
Copyright © 1994-98 Yahoo! All rights reserved. Send comments and bug reports to search@yahoo.com

도면6


시작
초기화 : 루프 카운터(LOOP_CNT)를 0으로 설정,
문서 커서(D_CUR)을 1로 설정,
수집 카운터(COLL_CNT)를 0으로 설정,
포인트 목록 포인트(L_PTR)를 일치목록으로 설정
S1
현재 목록 끝에 도달한 경우
현재의 목록에서 현재의 일치 레코드를 얻는다
S2
종료
S3
현재 일치 레코드== 그룹?
N
Y
S4
COLL_CNT>0?
Y
N
S8
D_CURR== 현재 일치?
N
Y
D_CUR을 현재 일치로 설정
S11
COLL_CNT=1
S5
COLL_CNT++
S9
COLL_CNT ==N?
N
Y
S10
현재의 일치 레코드를 출력하고 COLL_CNT를 리셋함
S12
COLL_CNT>0?
N
Y
S13
D_CUR이 그룹내에 있나?
Y
N
LOOP_CNT를 0으로 리셋
S14
LOOP_CNT 증분
S15
LOOP_CNT ==N?
Y
N
S16
가장 낮은 끝 문서번호 보다 큰 D_CUR을 설정
S6
LOOP_CNT를 0으로 리셋
S7
다음 목록을 가리키도록 L_PTR을 증분

도면7


시작
"OR" 어들을 분해하고, N=부제어 개수로 설정
S30
각각의 "OR" 부제어에 대한 일치목록을 생성
S31
커서를 각 일치목록에 할당하고, 커서가 각 목록 내 제 1 문서를 가리키게 한다
S32
각 목록 내 제 1 문서를 N-멤버 힙에 부가한다
S33
문서번호순으로 힙을 정렬
S34
최상위 항목을 힙으로부터 제거
S35
제거된 항목이 직접 일치인가?
Y
(직접)
보내져온 항목 문서번호가 출력목록내에 이미 있나?
Y
N
(그룹)
N
S36
문서를 출력목록에 더함
참조된 문서에 대한 일치 스코어를 증분함
S39
다음 항목을, 제거된 항목의 목록으로부터 이곳에 있다면 힙에 더한다
S37
힙이 비어있나?
S38
N
Y
종료