지식 맵을 위한 캐싱 기법
(A Caching Mechanism for Knowledge Maps)

정 준 원 † 민 경 섭 †† 김 형 주 †††
(Jun-Won Jung) (Kyung-Sub Min) (Hyoung-Joo Kim)

요 약 데이타를 효과적으로 다루기 위한 방법으로 데이타에 부가정보를 추가하는 TopicMap이나 RDF같은 지식맵에 대한 연구가 늘고 있다. 하지만 기존의 연구는 정보표현과 기술, 응용방안에 대한 연구가 주를 이루고 있으며 구현과 서비스에 대한 연구는 부족한 상태이다.

본 논문에서는 TopicMap 시스템에서의 캐쉬 관리 기능의 구현을 통해 실질적인 지식맵 서비스를 지원하기위해 고려해야 할 부분 중에서 지식맵의 효과적인 접근을 지원하기 위한 방법을 제안하였다. 먼저 기존 탐색방법의 장점을 최대한 수용하는 탐색 기법을 제안하고 이러한 환경하에서 지식맵 전송 효율을 향상시키고자 지식맵이 가지는 정보를 이용하는 캐쉬기법을 제안하였다. 본 논문에서 제안한 캐쉬기법은 어플리케이션의 접근 형태에 따른 물리, 논리적 단위로 정보를 캐쉬하는 기존의 방식과 달리 사용자가 지식을 접근하는 관점에서 효율을 높이고자 하였다. 즉 지식맵이 이미 자신에 대한 부가 정보뿐만 아니라 다른 지식간의 연관관계와 같은 정보를 가지고 있으므로 이러한 정보를 클러스터링 요소로 이용, 실제 사용자가 지식맵을 탐색하는데 있어 접근확률이 높도록 캐쉬집합을 생성하도록 하였다. 또한 캐쉬집합을 교체하는 방법에 있어서도 지식맵의 그래프 관계와 같은 정보의 연관성을 이용, 필요한 부분만을 전송함으로써 효율을 높이는 방법을 제안하였다.

키워드 : XML, 지식맵, 시멘틱 웹, 캐쉬, TopicMap

Abstract There has been many researches in TopicMap and RDF which are approach to handle data efficiently with metadata. However, No researches has been performed to service and implement except for presentation and description.

In this paper, We suggest the caching mechanism to support an efficient access of knowledgemap and practical knowledgemap service with implementation of TopicMap system. First, We propose a method to navigate Knowledgemap efficiently that includes advantage of former methods. Then, To transmit TopicMap efficiently, We suggest caching mechanism for knowledgemap. This method is that user will be able to navigate knowledgemap efficiently in the viewpoint of human, not application. Therefor the mechanism doesn't cash topics by logical or physical locality but clustering by information and characteristic value of TopicMap. Lastly, we suggest replace mechanism by using graph structure of TopicMap for efficiency of transmission.

Key words : XML, KnowledgeMap, Semantic Web, Cache, TopicMap

1. 서 론

1.1 연구동기

컴퓨팅환경의 급속한 발전에 따라 정보가 기하급수적으로 늘어나는 상황에서 기존에는 정보의 처리능력에 관심이 집중되었으나 이제는 정보를 효과적으로 접근하고 관리하는 것에 대한 연구가 활발히 이루어지고 있다. 그 예로서 W3C에서는 기존의 웹에 의미적 정보를 추가하여 효율적으로 웹 데이타를 관리할 수 있는 시멘틱 웹인 RDF(Resource Description Framework)를 제정하였으며, ISO에서는 기업이나 조직의 정보를 효율적으로 구축하기 위한 지식맵인 TopicMap을 제정하였다. 또한 온톨로지 구축을 위한 DAML+OIL, OWL에 대한 제정과 연구도 활발하게 이루어지고 있다. 이 같은 데이타의 효율적 관리에 대한 연구와 함께 정보접근환경도 변하고 있다. 네트워크의 발달로 지역내의 정보접근에서

· 본 논문의 연구는 ITRC와 BK사업의 지원하에서 이루어졌음
† 학생회원 : 서울대학교 전기.컴퓨터공학부
jwjung@oopsla.snu.ac.kr
†† 비 회 원 : 서울대학교 인지과학과
ksmin@oopsla.snu.ac.kr
††† 종신회원 : 서울대학교 전기컴퓨터공학부 교수
hjkim@oopsla.snu.ac.kr
논문접수 : 2003년 2월 19일
심사완료 : 2004년 2월 25일

원격지의 정보접근이 주를 이루게 됨에 따라 정보의 전송과 서비스방식이 전체 시스템 성능을 좌우하는 중요한 요소가 되었다. 이에 반해 현재 지식맵에 대한 연구는 지식맵에 대한 명세나 정보를 효과적으로 기술하는 방법에 주로 집중되어 있는 상황이다. 따라서 현재 네트워크 환경에서의 지식맵 서비스는 여러가지 문제점을 가진다. 그 중 하나는 바로 지식맵을 전송하는데 있어서 효율적인 전송 방법이 없다는 것이다. 현재 가장 TopicMap 규격을 잘 구현하고 있는 Techquila(社)의 TM4J와 같은 지식맵 처리 엔진의 경우도 전체 TopicMap을 전송하는 방식을 사용한다. 하지만 이렇게 전체 TopicMap을 전송하는것은 작은 메모리의 모바일 기기에서의 지식맵 서비스를 어렵게 할 뿐만 아니라 높은 네트워크 비용과 전체 지식맵이 전송되는 긴 시간을 필요로 한다. 또한 각각의 Topic을 전송하는 경우는 응답성능이 낮아지게 된다. 본 논문에서는 이와같은 문제점을 해결하기 위해 연관 정보를 중심으로 탐색하는 방법을 제안하고 이와 같은 환경에서 지식맵이 가지는 정보의 연관성을 이용해 탐색가능성이 높은 정보를 캐쉬하는 방법을 제안하였다. 그리고 이와 같은 기법들을 지식맵의 일종인 TopicMap 서비스에 적용함으로써 메모리 요구량을 줄이고 네트워크 비용을 감소시킴으로써 네트워크 환경에서 지식맵 어플리케이션을 효율적으로 지원할 수 있음을 보이고자 한다.

1.2 논문의 구성

본 논문의 구성은 다음과 같다. 먼저 2장에서는 배경지식으로 지식맵의 효용성에 대해 설명하고 지식맵의 일종으로 본 논문의 시스템 구현에 사용한 TopicMap에 대해서 설명한다. 3장에서는 TopicMap과 관련된 연구들을 살펴보고 기존 연구와의 차이점을 설명하며 4장에서 시스템 구조를 보이고 5장과 6장에서 구체적인 캐쉬 생성과 교체기법에 대해 설명한다. 그리고 이렇게 제안한 기법이 정보를 고려하지 않는 기존의 캐쉬 기법보다 성능 향상이 있음을 7장에서 실험을 통해 보이고 8장에서는 전체적이 결론을 정리하는 것으로 본 논문이 구성된다.

2. 배경지식

지식이나 정보를 효율적으로 관리하기 위한 연구는 지식맵 이전에 이미 여러 분야에서 이루어졌다. 문헌정보학에서는 색인, 용어해설, 시소러스를 통해 책에 있는 정보간의 관계를 표현하려 하였으며, 인공지능 분야에서는 인간의 정보개념을 기계가 인식할 수 있는 체계로 표현하기 위해 개념들간의 관계를 나타내는 'Semantic Network', 'Associative Network' 등을 이용하였다[1]. 이러한 기법들의 공통점은 정보들의 연관 관계를 통해 구조화함으로써 지식의 체계를 구축한다는 것이며, 이와 같은 개념들을 지식관리 시스템에 적용한 것이 바로 지식맵이다.

지식맵이 개념관계를 표시하는 Semantic Network과 같은 기존의 방식과 다른점은 기존의 방식이 개념의 표현과 의미의 구축에 중점을 두었다면 지식맵은 Occurrence와 같이 정보에 대한 위치를 가짐으로써 개념과 실제 데이타간의 연관성을 표현할 수 있게 된 것이다. Semantic Network뿐만 아니라 기존 OODB의 개념적 모델도 지식맵과 비슷하게 개념을 모델링한다는 점에서는 유사성을 가지지만 OODB의 개념적 모델의 목적은 DB를 위한 개념적 설계에 있는 반면 지식맵은 구조를 갖지 않는 데이타에 연관관계를 추가하여 구조화하기 위함이 목적인 이라는 점에서 다르다. 실제 웹상의 정보를 구조화하기 위한 노력이 W3C의 시멘틱웹을 통해 나타나고 있으며, ISO에서 제정한 TopicMap은 더욱 광범위한 정보 기술능력을 가지기 때문에 웹뿐만 아니라 지식관리 시스템에 이르기까지 다양하게 적용이 가능하다.

TopicMap은 기본적으로 정보나 개념을 나타내는 Topic과 Topic간의 연관 관계를 나타내는 Association, 그리고 Topic에 해당하는 정보에 대한 위치를 나타내는 Occurrence로 구성된다. 또한 세부적으로 Type, Role, Scope등 다양한 부가 정보를 기술하는 항목들 통해 정보와 정보간의 관계를 구체적으로 표현할 수 있고 다시 이를 통해 새로운 지식을 얻을 수 있는 구조를 제공한다. 특히 정보의 위치를 명시하는 Occurrence에서 URL뿐만 아니라 Xpath, 데이타베이스의 특정 레코드까지 다양한 표현이 가능하므로 다양한 정보에 대한 조직화가 가능하다[2].

TopicMap이나 시멘틱웹과 같은 지식맵 위에 더욱 고차원적인 처리를 지원하기 위해 온톨로지가 쓰일 수 있다. 지식맵이나 시멘틱웹은 정보와 의미를 조직화하고 기술하는데 중점을 두고 있기 때문에 제약조건, 논리와 같은 고차원적인 정보를 기술하고 처리할 수 있는 온톨로지로 확장이 가능하다[3,4]. 온톨로지란 기존에 철학분야에서 특정 개념을 나타내기 위해 사용되는 단어의 집합을 의미했으나 현재 지식처리 분야에서의 온토롤지는 위와 같은 기능을 수행하는 정보의 집합을 의미한다. 지식맵을 온톨로지로 확장함으로써 소프트웨어 에이전트를 이용한 지능적인 정보 처리가 가능해지며 대표적인 온톨로지 기술언어로 DARPA(Defense Advanced Research Projects Agency)에서 제정한 DAML+OIL과 W3C의 OWL이 있다.

본 논문에서는 지식맵 수준에서의 데이타 처리까지만 다루도록 하며 TopicMap이 가지는 강력한 정보 구성 능력과 표준이라는 범용성을 들어 TopicMap을 지식맵

으로 사용하여 연구를 수행하였다. 따라서 본 장에서는 TopicMap에 대한 기본적인 개념을 소개하고자 한다.

2.1 Topic

Topic은 사물, 사람, 개체, 개념 등 기술하고자 하는 대상을 의미한다. 즉 Topic은 정보를 기술하는 사람이 나타나고자 하는 대상을 표현할 수 있는 단어로 구성되어야 한다. 다음의 그림 1은 OO대학교 OOO 연구실[1] 홈페이지에서 추출한 토픽의 예이다. 기술하고자 하는 대상으로서 '홍길동', '김철수', 'Caching Mechanism'이 Topic으로 지정되었다. 또한 각 Topic은 TopicType을 통해 분류된다. 그림에서 '홍길동' Topic은 '학생'이라는 Topic Type을 가지며 '김철수' Topic은 '교수'라는 Topic Type을, 'Caching Mechanism' Topic은 '논문'이라는 Topic Type 가진다. 이와 같이 Topic은 Topic Type을 통한 범주를 나타낼 수 있으며 이는 객체지향 모델의 Class와 Instance의 관계와 유사하다. 또한 '학생'과 '교수' Topic Type이 '구성원'이라는 Topic Type에 속하는 것처럼 Topic Type도 Topic Type을 통해 분류될 수 있다.

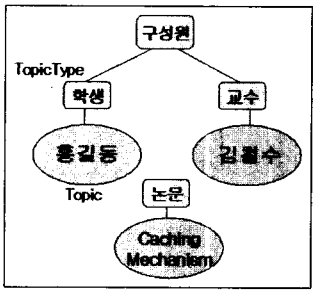

그림 1 Topic과 Topic Type

2.2 Association

Association은 Topic간의 연관 관계를 표현한다. 이것은 Topic Type과 같은 상하관계를 표현하는 것이 아니라 둘 이상의Topic들간의 관계를 정의하는 것이다. Topic간의 관계가 Association으로 표현됨과 동시에 그 관계가 어떤 것인지는 Association Type을 통해 나타내며, 이 관계에서 각 토픽이 어떤 역할을 하는지는 Role을 통해 기술된다. 그림 2는 연구실 홈페이지에서 추출한 Topic들 간의 Association을 나타내고 있다. 먼저 '홍길동' Topic과 'Caching Mechanism' Topic은 '작성'이라는 관계를 가지고 있으며, 이 관계에서 '홍길동' Topic은 '작성자'의 Role을 가지며 'Caching Mechanism' Topic은 '논문'의 Role을 가진다. 이것은 '홍길동이 Caching Mechanism이라는 논문을 작성했다.'라는 사실을 표현하고 있다. 이와 같은 방법으로

1) 서울대학교 객체지향시스템 연구실

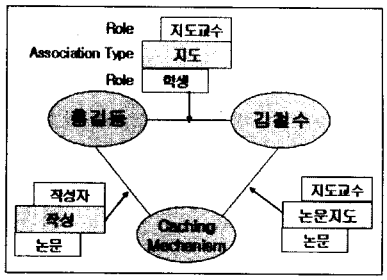

그림 2 Association과 Association Type, Role

Association은 Topic간의 관계를 상세하게 기술할 수 있으며 Topic간에는 여러 개의Association을 가질 수 있다.

2.3 Occurrence

Topic은 자신이 기술하는 정보가 가리키는 하나 이상의 실질적인 참조에 대해 Occurrence를 통해 기술한다. 그림 3은 연구실과 관련된 Topic들과 Occurrence를 나타내고 있다.

그림 3에서 보면 '홍길동' Topic은 홍길동이라는 사람과 관련된 정보가 있는 홈페이지의 URL을 Occurrence를 기술하고 있다. 마찬가지로 'Caching Mechanism' Topic은 해당 논문이 위치하는 URL을 Occurrence로 기술하고 있다. 이와 같이 Occurrence는 Topic에 대한 정보의 위치를 나타내는데 URL뿐만 아니라 데이타베이스 테이블의 레코드에 이르기까지 다양한 형태의 위치 정보를 기술할 수 있기 때문에 여러 가지형태의 정보를 TopicMap을 통해 통합할 수 있는 강력한 기능을 제공한다.

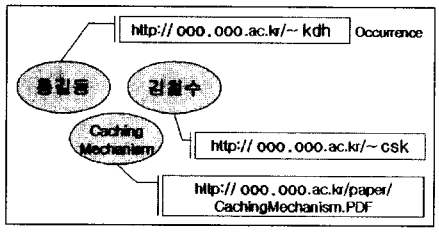

그림 3 Topic과 Occurrence

지금까지 살펴본 Topic, Association, Occurrence 외에도 TopicMap 표준안은 다양한 요소들을 통해 정보를 효과적으로 구조화할 수 있도록 한다.

3. 관련 연구

TopicMap에 대한 표준안은 2000년에 제정되었기 때문에 다른 분야만큼 많은 연구가 진행되어있지는 않은 상태이다. 대부분의 연구는 TopicMap 규격에 대한 연

구[5-7]와 응용 방안에 대한 구상[2]이 주를 이루고 있으며, 실질적인 서비스의 문제에 대한 연구는 적은 상태이다. TopicMap서비스와 관련하여 TopicMap의 저장, 처리, 전송, 표현으로 나눈다면 현재 그나마 연구가 되고있는 부분은 TopicMap의 표현에 관련된 부분이다. 현재 TopicMap 탐색방식과 관련하여 가장 일반적인 형태로 그래프 형태의 탐색과 트리 형태의 탐색[8] 그리고 텍스트 기반의 질의 탐색이 제안되어 있으나 모두 표현 능력과 효율성 면에서 문제가 남아있는 상태이다.

TopicMap을 처리하는 분야에 대해서는 TopicMap을 그래프로 모델화하고 다루는 방법이 주를 이룬다[9]. 즉 Topic을 노드로, Association을 간선으로 보고 TopicMap을 그래프 화 하여 TopicMap이 그래프 모델로 변환 될 수 있으며 그래프 기법들을 적용하여 처리할 수 있다는 것을 보였다. TopicMap이 그래프적인 구성을 가진다는 것을 생각할 때 그래프로 모델화가 가능한 것은 당연하게 보인다. 하지만 일반적인 그래프 기법을 적용하는 것에 대해 그 결과의 효용성에는 의문이 생긴다. 왜냐하면 TopicMap은 Topic과 Association의 연결이 전부가 아니라 TopicType과 같이 간선으로 간주되지 않지만 그래프 구조상 멀리 있는 Topic까지도 연결하는 개념을 가지고 있기 때문이다. 또한 TopicMap은 Topic과 Association외에 Role, Scope 등 Topic에 대한 정보를 기술하는 다양한 요소들을 가지지만 Topic과 Association 만으로 구성된 그래프는 이와 같은 정보를 모두 반영하지 못한다. 예를 들어 [9]의 논문에서 그래프의 거리를 이용하여 클러스터링된 결과는 단순히 물리적 거리가 인접한 집합을 만들어 줄 뿐 실제로 인접한 정보라고는 할 수 없다. 예를 들어 연구실에 '정준원'과 '이한준'이라는 사람이 있다고 하자. 이 둘이 연구실 TopicMap의 Topic으로 존재할 때 직접적으로 연결된 Association이 없을 수 있다. 하지만 '연구실 맴버'라는 TopicType을 가짐으로써 두 Topic은 같은 연구실 동료라는 정보를 가지게 되는 것이다. 이 경우 단순히 그래프 정보에 기반한 클러스터링을 수행할 경우 두 Topic이 가까운 정보 인접성을 가짐에도 불구하고 서로 다른 클러스터링 집합으로 분류될 수도 있는 것이다. 따라서 TopicMap의 그래프적인 모델링과는 별개로 TopicMap의 정보를 고려한 처리 방법이 필요하다.

TopicMap을 그래프 구조로 보고 캐쉬하는 방법은 웹 캐쉬 기법중 응답성을 높이기 위한 prefetching[10,11] 기법과 유사할 수도 있으나 웹 캐쉬 기법을 위한 그래프 구조는 단순히 페이지를 노드로, 링크를 간선으로 간주해 구성된 그래프이기 때문에 앞서 설명한 바와 같이 TopicMap에 적용하는데 있어서는 TopicMap의 부가정보를 고려하지 못한다.

기존의 DB분야에서도 시멘틱정보를 이용한 캐쉬기법[12]이 존재한다. 이 기법도 클라이언트 서버 환경에서 클라이언트가 캐쉬된 데이타에 대한 시멘틱정보를 유지하고 질의가 발생하면 시멘틱정보를 통해 현재 캐쉬에 없는 데이타만을 서버에 요구한다. 또한 페이지 교체시에도 연관데 데이타들의 집합인 의미적 영역(semantic region)을 이용해 튜플단위 캐쉬방법에서 교체를 위한 정보저장 오버헤드를 감소시킨다. 하지만 이 연구는 시멘틱데이타 자체를 캐쉬하는 것이 아니라 데이타 존재에 대한 정보를 시멘틱데이타로 기술하고 처리한다는 데서 본 논문과 다르며 구조에 있어서도 클라이언트가 서버에게 요청할 질의를 생성해내는 반면 본 논문은 클라이언트의 오버헤드를 최소화하기 위해 캐쉬 생성이 서버측에서 이루어진다는 점에서 다르다.

본 논문에서는 TopicMap처리를 위해 기존의 연구와 같이 TopicMap을 그래프로 모델링 하였지만, 그 처리에 있어서는 그래프의 구조 뿐만 아니라 TopicMap이 가지는 다양한 부가 정보를 이용함으로써 TopicMap의 정보를 고려한 결과가 생성되도록 하였다.

4. 시스템 구현

4.1 시스템의 구조

본 논문에서 가장 중점이 되는 TopicMap 캐싱 기법은 실제 사용자가 정보를 탐색하는데 있어서 관심이 있는 정보를 중심으로 탐색해 나간다는 전제로부터 시작한다. 따라서 전체 TopicMap을 전송하고 탐색하는 것이 아니라 탐색이 시작되는 Topic으로부터 사용자가 탐색할 가능성이 높은 Topic들을 캐쉬하여 네트워크 비용과 메모리 사용량을 줄이고 Topic 접근에 대한 응답속도를 높이는데 그 목적이 있다. 그러기 위해서는 먼저 TopicMap 탐색환경부터 관심있는 정보를 중심으로 탐색할 수 있는 기능이 지원되어야 한다. 따라서 기존의 탐색 기법을 기반으로 연관성에 의한 탐색을 강화한 탐색환경을 제안하였다. 그것은 바로 트리 탐색기법에 Keyword기반 탐색 기능을 추가하고 부가 정보의 표현을 이용한 연관 관계의 자유로운 탐색을 지원하도록 하는 것이다. 먼저 본 논문에서 제안하는 탐색 기법은 다음과 같다. TopicMap에 대해 Topic Type으로 분류 된 트리 구조를 표현한다. 그리고 트리구조에 대한 계층적 검색과 함께 Keyword입력을 받을 수 있도록 하여 검색을 원하는 Topic을 바로 검색할 수 있도록 한다. 이와 같은 방법으로 트리 탐색기법의 계층적 분류에 대한 특징을 지원하면서 Keyword기반 질의 기법에 의한 즉각적 정보검색의 장점을 수용하여 경로를 따라 탐색해야 하는 트리 탐색기법의 단점을 해소할 수 있게 된다. 또한 각 정보를 선택하면 Association, Role, Type, 및 연

관된 Topic들을 표현하고 이 결과들에 대해서도 다시 검색이 가능하게 함으로써 TopicMap의 그래프적인 구조 표현과 탐색을 지원하도록 한다. 그리고 Occurrence 정보가 가지는 자료를 함께 나타내도록 함으로써 정보의 연관성을 빠짐없이 표현 한다. 이러한 탐색환경은 연관된 정보를 중심으로 탐색하기 때문에 본 논문에서 구현하고자 하는 캐쉬기법이 적용되기에 적합하다. 그림 4는 이와 같은 방법으로 OO대학교 OOO연구실에서 개발한 'KBOX'라는 TopicMap 브라우져이다. 이와 같이 연관정보를 중심으로 탐색을 수행하는 탐색환경을 전제한 후에 연관성에 의한 TopicMap 캐쉬을 수행하게 된다. 이때 캐쉬와 관련해서는 크게 캐쉬 생성부와 캐쉬 교체를 담당하는 캐쉬 관리부로 나뉜다. 캐쉬 생성과 교체 기법에 대한 자세한 설명은 다음 절에서 이루어진다. 지금까지 설명한 시스템의 구조를 표현하면 그림 5와 같다.

전체 시스템은 기본적으로 TopicMap문서에 대해 TopicMap 서비스를 제공하는 Server단과 TopicMap을 탐색하는 Client단으로 나뉘어진다. 서버측에서 KBOX는 TopicMap 탐색을 지원하는 어플리케이션이며 Client는 Java 가상머신을 지원하는 브라우져를 통해 접속하게 된다. 서버의 하단에는 RDB로 구성된 저장소가 존재하여 TopicMap을 저장하게 된다. RDB로 저장된 TopicMap은 TopicMap Processing Engine인 RDB TM4J를 통해 TopicMap을 조작할 수 있도록 한다. RDB TM4J는 Ontopia사에서 Opensource로 공개한 TopicMap engine을 RDB상에서 수행되도록 개선한 것이다. 이 RDB TopicMap engine과 이를 사용하는 어플리케이션인 KBOX사이에 본 논문에서 구현한 캐쉬 생성기와 관리자가 존재한다. 캐쉬 생성기는 TopicMap정보에 기반한 클러스터링과 휴리스틱기법으로 캐쉬를 생성하고 캐쉬 관리자는 캐쉬 교체를 담당한다. KBOX에서 Topic에 대한 요청을 캐쉬 관리자에게 전송하면 캐쉬 관리자는 캐쉬 생성기에게 새로운 캐쉬 생성을 요청한다. 캐쉬 생성기는 DB에 있는 Topic들을 가져와 본 논문의 알고리즘을 통해 캐쉬를 생성하고 캐쉬 관리자에게 돌려준다. 캐쉬 관리자는 사용자별로 전에 전송했던 캐쉬 정보를 가지고 있다가 기존 캐쉬와 새로 생성된 캐쉬중 사용자 캐쉬와의 차집합을 계산하여 새로 추가된 부분만을 추가하고 캐쉬의 용량에 맞게 조절한 후전송할 캐쉬를 만들어 Client에게 전송한다.

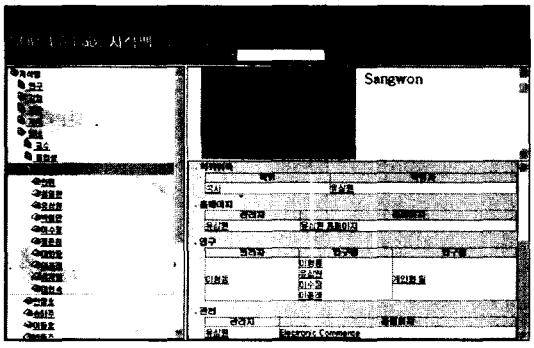

그림 4 KBOX TopicMap 브라우져

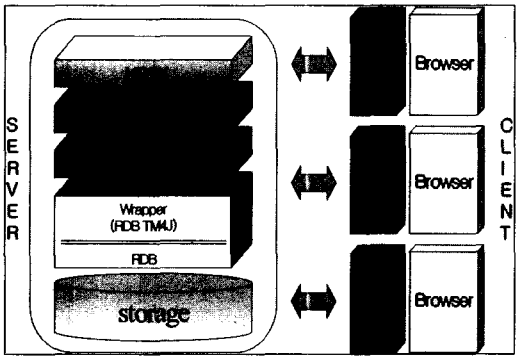

그림 5 전체 시스템 구조도

5. 캐쉬 생성

TopicMap은 비 구조적데이타에 구조적 정보를 추가하기 위한 메타데이타라는 의미 외에도 그 자체로서 정보간의 관계를 가지는 데이타이다. 특히 TopicMap의 특성상 TopicMap에 대한접근은 무작위적 접근이나 지역적인접근 보다는 Topic과 Association, 그리고 Type 등의 부가 정보를 따라 연관성에 의해 이루어지게 된다. 따라서 본 논문에서는 캐쉬를 생성함에 있어 하드웨어나 OS분야에서 데이타 접근의 지역성에 기반해 캐쉬를 생성하는 것과 달리 정보의 연관성에 기반해 캐쉬를 생성하고자 한다. 따라서 TopicMap의 정보를 이용해 한 Topic과 연관되어 탐색될 가능성이 높은 Topic들을 선정해야 하는데 그 방법으로서 클러스터링과 휴리스틱을 적용한다. 클러스터링은 Topic들의 특성을 데이타로 분석하여 연간 관계가 높은 Topic을 찾아내고자 함이며, 휴리스틱은 TopicMap의 탐색 특성을 고려할 때 연관성을 높아 탐색될 가능성이 분명하지만 클러스터링에 사용되는 factor로는 같은 클러스터에 포함될 가능성이 적은 Topic들을 추가하여 보완하기 위함이다. 이때 사용되는 클러스터링이 기존의 연구[9,13]와 다른점은 기존의 연구에서는 클러스터링 factor로서 그래프상의 거리를 이용하지만 본 논문에서는 TopicMap의 특성 정보를 factor로 사용한다는 것이다. TopicMap의 정보를 이용한 클러스터링과 휴리스틱은 다음과 같이 이루어진다.

5.1 K-means 알고리즘

K-means 알고리즘[14]은 비계층적인 클러스터링 기법으로서 hard clustering의 특성(한 데이타는 한 클러스터에만 포함)을 가지며, 개체간의 거리(Euclidean Distance)를 통해 클러스터를 생성하는 대표적인 기법으로서 처리가 효율적이고 우수한 성능을 나타내는 것으로 알려져 있다. 본 논문에서는 K-means 알고리즘을 통해 Topic을 클러스터링하므로 이에 대해 간단히 설명한다.

먼저 x라는 데이타가 n개의 factor를 가질 때 다음과 같은 백터로 표현된다.

$$x=[x_1, x_2, ..., x_n]$$

그리고 데이타간의 거리를 구하기 위해 다음과 같은 Euclidean Distance를 사용한다.

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

다음은 이를 통한 K-means 알고리즘의 각 단계를 나타낸다.

step 1. 클러스터의 수(K개) 선정 및 중심점(center) 선정.

step 2. (1) 모든 입력 백터과 각 클러스터의 중심점과의 거리 비교

(2) 가장 가까운 중심점이 속한 클러스터의 element로 등록

step 3. 클러스터별로 기존의 element와 새로 편입된 element를 포함하여 중심점을 다시 계산

step 4. (1) 모든 클러스터의 중심점에 변화가 없을 때까지 step 2 수행

(2) cluster들의 중심점에 아무런 변화가 없으면 종료

5.2 클러스터링

캐쉬를 생성하기 위한 첫번째 과정으로 클러스터링을 수행한다. 이때 클러스터링은 그래프의 거리적 특성이나 노드의 가중치를 이용하는 그래프 클러스터링 기법이 아니라 TopicMap의 Type, Accociation Type, Topic 간 거리와 같은 TopicMap의 특성값들을 factor로 데이타의 유사도 측정에 널리 사용되는 K-means algorithm을 이용한다.

클러스터링을 수행하는 과정은 다음과 같다.

- 현재 요구가 들어온 Topic을 중심으로 TopicMap을 그래프 구조로 볼 때 distance 6의 범위를 클러스터링 범위로 설정한다.(여기서 distance란 그래프 이론에서의 distance를 의미하는 것으로 한 노드와 다른 노드 간의 거리를 의미한다.) 이때 전체 TopicMap에 대해서 클러스터링을 수행하지 않는 이유는 사용자가 캐쉬를 벗어나는 탐색을 할 때마다 해당 Topic에 해당하는 클러스터를 동적으로 생성하기 위함이다. 또한 클러스터링 범위를 지정함으로써 클러스터링에 소요되는 시간을 줄일 수 있다. 여기서 distance 6은 실험을 통해 가장 적절한 범위를 선택하였다.
- K(클러스터링에 사용될 중심점 개수)및 중심점설정
  현재 요구가 들어온 Topic과 distance 6 이내의 Topic들로 Association의 개수를 가중치로 순위 2까지의 Topic을 중심점으로 택한다. 즉 K값이 3이 되는 것이다.(K값을 3으로 하는 이유는 distance 6에 대해 클러스터링했을 때의 실험결과 가장 적절한 크기의 cluster가 만들어지기 때문임) 이때 각 중심점간의 거리는 distance가 최소 3 이상이어야 한다. 이것은 나중에 클러스터링된 결과를 distance 3범위 내에서 취하기 위함이며 선택된 중심점이 이 조건을 만족시키지 못하면 다음 가중치 순위의 Topic을 선택한다.
- TopicMap의 정보 기반 클러스터링 factor 추출
  클러스터링 factor로는 Topic의 특성을 나타내거나 현재 요구가 들어온 Topic과의 연관성을 나타낼 수 있는 정보를 이용한다. 본 논문에서 클러스터링 factor로 사용한 특성은 다음과 같다.
  - Topic의 Type
  - Topic Type의 Type (상하 계층관계 고려)
  - Topic이 가지는 Association Type
  - Topic이 참여하는 Association에서의 Role
  - 요구가 들어온 Topic과의 거리
- K-means clustering 수행
- 요구가 들어온 Topic이 속한 클러스터만을 취함
- 위의 결과로 생성된 클러스터 중 요구가 들어온 Topic을 중심으로 distance 3 이내에 해당하는 Topic만을 취한다. 그림 6은 클러스터링을 통해 distance 1~3의 캐쉬를 생성하는 모습을 나타낸다.

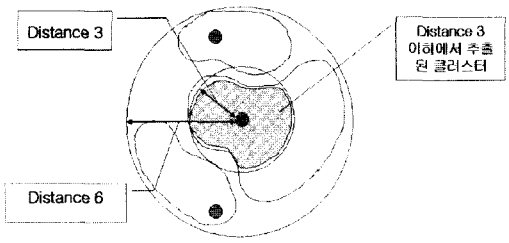

그림 6 K-means algorithm에 의한 클러스터링

5.3 휴리스틱

distance 1~3까지 클러스터링을 수행하여 캐쉬를 생성한 후 distance 4~6까지에 대해서는 정보의 연관성에 근거한 휴리스틱으로 캐쉬를 추가한다. 이렇게 두 단계로 범위를 나누는 이유는 클러스터링은 특성값을 중심으로 수행되기 때문에 외각부분으로 갈수록 TopicMap 정보의 연관성이 떨어질 가능성이 크기 때문이다. 따라서 외각 부분은 TopicMap의 개연성과 관련해 직관적으로

연관이 높다고 생각되고, 탐색에 이용될 확률이 큰 Topic들을 추가한다. 예를 들어서 같은 Type에 속하는 Topic들은 특성값이 다르고 먼 거리에 존재하더라도 탐색에 이용될 확률이 매우 높으므로 캐쉬에 추가하는 것이 높은 효율을 보일 것이다. 이와 같이 직관적인 방법으로 distance 4~6에 적용되는 휴리스틱은 다음과 같다.

- 요구가 들어온 Topic의 인접 Topic은 기본적으로 추가한다(distance 1).
- 요구가 들어온 Topic과 같은 Type을 가지는 Topic은 기본적으로 추가한다.
- 같은 Type을 가지는 Topic에서 distance 1인 Topic들을 추가.
- 위의 Topic과 같은 Type을 가지는 Topic을 추가
- 요구가 들어온 Topic과 distance 1인 Topic과 같은 Type을 가지는 Topic추가

그림 7에서 보면 이와 같은 휴리스틱의 근거를 찾아볼 수 있다. 요구가 들어온 Topic이 '정준원'일 경우 Paper 4와 같은 Topic은 바로 인접하므로 기본적으로 탐색의 결과 표현에 사용된다.

또한 직접 Association이 없는 '민경섭', '이한준'이라는 Topic도 'OOPSLA member'라는 같은 Type을 가지기 때문에 탐색될 확률이 높다. 그림 8은 지금까지 설명한 TopicMap 정보기반 클러스터링과 휴리스틱을 통해 캐쉬가 생성되는 것을 나타낸다.

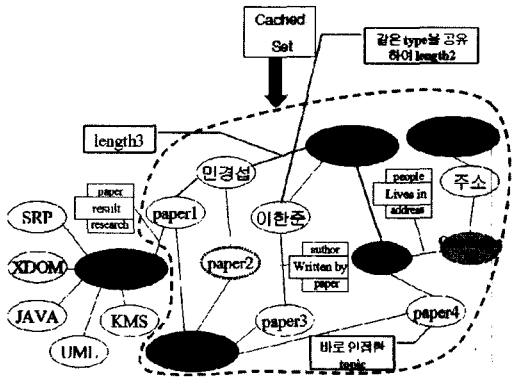

그림 7 휴리스틱의 예제

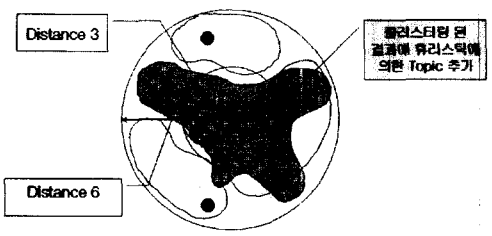

그림 8 클러스터링과 휴리스틱을 통해 생성된 캐쉬

6. 캐쉬 교체 기법

요구가 들어온 Topic에 대해서 사용자의 캐쉬에서 찾아보고 해당 정보가 없으면 fault가 발생한다. 이때 다음과 같은 캐쉬 교체 과정이 발생한다.

- 클라이언트는 현재 fault가 발생한 Topic과 함께 현재 캐쉬 정보를 서버의 캐쉬 관리자에게 전송한다.
- 서버는 요구가 들어온 Topic에 대해 캐쉬 생성자를 통해 새로운 캐쉬를 생성한다.
- 캐쉬관리자는 기존에 사용자가 탐색하던 캐쉬와 비교 후 차집합만을 캐쉬에 추가한다.
- 이때 캐쉬가 캐쉬메모리 용량을 초과하면요구가 들어온 Topic을 중심으로 가장 먼 거리에 있는 Topic부터 삭제하여 캐쉬메모리 용량에 맞추어 전송한다.

본 논문에서 제안하는 캐쉬 교체 기법 역시 기존에 다른 분야에서 사용되던 교체 알고리즘과 다르게 Topic-Map의 그래프적인 관계에 기반해서 수행한다. 접근 빈도에 기반하는 기존의 캐쉬 교체 알고리즘들을 이용할 경우 TopicMap의 구조적 정보에 의한 연관성이 손상될 가능성이 커지기 때문이다. 또한 그래프 구조의 캐쉬 데이타들의 연관성을 감안할 때 새로 생성된 캐쉬를 중심으로 역탐색 방향으로 중복된 캐쉬가 생성되기 때문이다. 따라서 그림 9와 같이 새로 생성된 캐쉬와 기존의 캐쉬의 차집합 만을 전송함으로써 캐쉬의 효율을 보존하면서 네트워크비용을 절약할 수 있다.

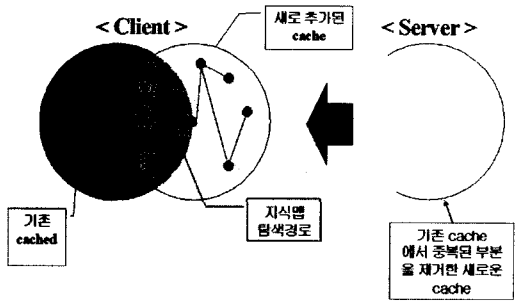

그림 9 그래프 기반 캐쉬교체기법

7. 실험 및 분석

7.1 실험 환경

실험을 위해 OO대학교 OOO연구실 홈페이지의 내용을 기반으로 TopicMap을 구축하고 TopicMap 서비스를 제공하는 서버와 탐색을 수행하는 클라이언트를 이더넷으로 연결하는 실험 환경을 구축하였다. 그리고 캐쉬 기법의 성능 평가는 다음과 같이 이루어 졌다. 먼저 캐쉬가 적용되지 않는 기존의 TopicMap 서비스 기법과

의 비교는 TopicMap 전체를 전송하고 탐색을 하는 환경, 그리고 Topic에 대한 요구시마다 Topic을 각각 전송하는 방법과의 메모리 사용량과 응답 시간을 비교하였다. 두번째로 본 논문에서 제안한 캐쉬 기법의 효율성을 보이기 위해 TopicMap의 특성을 이용하지 않고 일정 거리의 Topic을 캐쉬하는 경우, 그리고 LRU 알고리즘을 사용하는 경우에 대해서 캐쉬 용량 별로 Hit ratio를 측정하였다. 이러한 측정은 각 알고리즘에 동일한 경로에 대한 100번의 탐색을 통해 측정하였다. 이 100번의 탐색은 유형에 따라 지역적 접근, 연관성에 의한 접근, 무작위적 접근, 그리고 실제 탐색 유형과 유사하다고 생각되는 조합적 접근에 대해 측정하였다. 무작위적인 접근이란 토픽의 연관성에 의한 접근이 아닌 Association이 없는 Topic들을 탐색하는 것이다. Type분류를 따라 건너뛰면서 탐색하는 것이 그 예이다. 전반적인 탐색을 하거나 관심의 범위를 이동할 때 발생하는 탐색 유형이다. 지역적 반복이란 일정 범위에서 Topic들을 반복해서 탐색하는 경우이다. 일정 범위의 정보를 비교하기 위해 탐색하는 경우의 탐색 유형이라 할 수 있다. 연관적 탐색이란 한 Topic으로부터 연관된 정보를 따라 탐색해 나가는 경우를 의미한다. 즉 탐색 결과에 대해 계속해서 탐색해 나가는 탐색유형이다. 조합적 접근은 실제 사용자가 정보에 접근하는 패턴을 가정한 것으로, 최초 TopicMap에 접근하는 사용자가 관심 Topic을 중심으로 주변정보를 검색(지역적 접근)하다가 상세하게 알고 싶을 경우 연관정보를 따라 검색하게 되고(연관적 접근), 원하는 정보를 모두 찾거나 원하는 정보를 찾지 못하는 경우 다른 부분으로 가서(무작위적 접근) 지역적인 접근과 연관적 접근을 수행하는 탐색 패턴을 의미한다. 조합적 접근에서 실제 사용자의 탐색패턴은 주로 연관검색과 지역적 검색이 주를 이루고 완전히 다른 정보로 이동하는 무작위적 접근은 적은 빈도로 이루어진다고 가정하였다. 이와 같이 다양한 탐색 유형에 대해서 고려함은 물론 탐색 경로에 속하는 토픽의 연결 차수에 있어서도 다양한 차수를 가지는 Topic들이 동일한 빈도로 탐색되도록 하였다.

7.2 실험 결과

먼저 캐쉬를 사용하지 않은 기존 서비스 환경과 메모리 사용량, 응답속도를 비교한 결과는 그림 10과 같다.

이때 측정한 평균 응답시간이 대체적으로 긴 이유는 실험 Topic중에 worst case로서 1:N 연관의 N이 매우 큰 Topic도 포함시켰기 때문이다. 위의 결과에서 요구시마다 Topic을 전송해오는 방식은 평균 10KB가량의 메모리를 사용하는 반면 네트워크 접속과 DB접속에 대한 시간이 늘어나므로 9530ms의 긴 응답시간을 보여준다. 반면 메모리에 모두 전송하고 서비스하는 경우에는

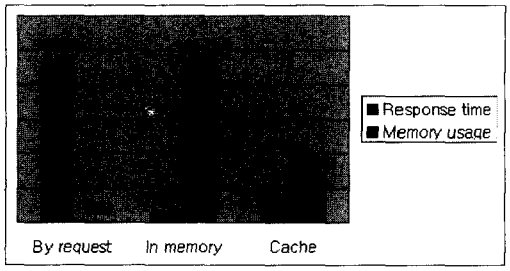

그림 10 캐쉬 적용과 비적용시의 성능

570ms라는 빠른 응답을 보여주지만 전체 TopicMap 크기만큼의 메모리를 필요로 한다. 이 경우 TopicMap의 크기가 늘어날수록 메모리 요구량도 늘어나고 토픽맵 전체를 전송하는데 시간이 길어지므로 초기 응답시간이 길어진다. 또한 이 같은 메모리 요구량은 모바일 기기와 같이 적은 메모리를 가지는 기기에서는 사용되기 어렵다. 마지막으로 300KB의 캐쉬를 적용한 경우 캐쉬 사이즈만큼의 메모리를 사용하면서, 1010ms의 응답 속도를 가진다. 이와 같은 메모리 사용량과 응답속도를 감안할 때 기존의 전송방식보다 효용이 높음을 알 수 있으며 모바일 네트워크 환경과 같이 적은 메모리를 요구하고 네트워크 비용이 높은 환경에도 유용하게 적용될 수 있으리라 보인다. 다음에는 기존 캐쉬기법 그리고, TopicMap의 특성값을 이용하지 않은 캐쉬 기법과의 비교결과를 살펴보자.

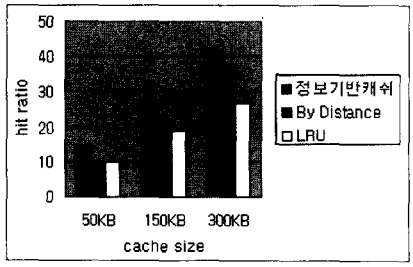

그림 11 무작위적접근에 대한 성능 비교

그림 11은 무작위적인 접근에 대한 결과를 나타내고 있다. 모든 캐쉬기법의 효율이 떨어지고 있으며 실험에 사용한 TopicMap용량(1014KB)의 30%인 300KB 정도가 되어야 그나마 30~40%정도의 hit ratio를 보여준다. 특히 무작위적 접근일 경우에는 접근 빈도에 의한 캐쉬 기법의 성능이 크게 떨어지며, 접근 빈도보다는 그래프적인 거리나 여기에 TopicMap정보를 이용하는 캐쉬기법이 효율적임을 볼 수 있다.

그림 12는 지역적인 접근에 대한 결과를 나타낸다. 지역적인 접근일 경우 캐쉬 용량과 상관없이 LRU가 우수

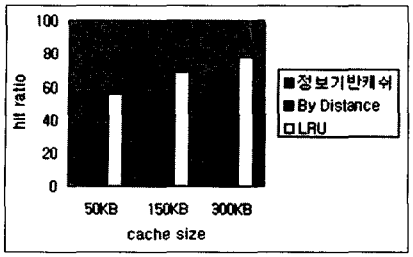

그림 12 지역적인 접근에 대한 성능 비교

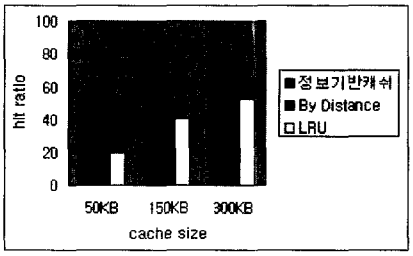

그림 13 연관적 접근에 대한 성능 비교

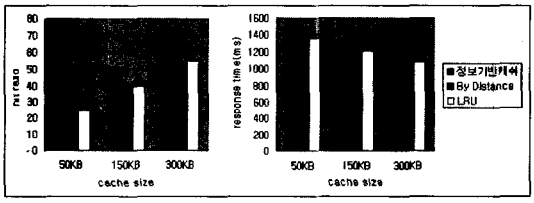

그림 14 조합적 접근에 대한 성능 비교(hit ratio, 응답시간)

한 성능을 보여주고 있다.

그림 13에서 나타나듯이 연관 관계에 의한 접근은 LRU기법이 가장 낮은 성능을 보이고 있으며 TopicMap의 특성을 이용해 캐쉬를 생성하는 것이 단순 거리에 의한 방법보다 우수함을 보인다. 특히 50KB캐쉬에서와 같이 적은 용량의 캐쉬에서 큰 차이를 보임으로써 TopicMap의 특성값을 이용하는 것이 유용함을 알 수 있다. 즉 연관 관계에 의한 탐색에 있어서 LRU와 같이 접근 빈도를 고려하거나 그래프 상의 거리만을 고려하는것 보다 Topic간의 정보적 연관성을 고려하는 것이 의미있음을 알 수 있다.

무작위적, 지역적, 연관적, 접근이 조합되어 실질적으로 사용자가 탐색하는 환경을 가정한 조합적 접근은 그림 14와 같은 결과를 보인다. 여기서 보면 그래프상의 거리나 LRU기법에 의한 방법이 비슷한 효율을 보이고 TopicMap의 특성을 이용하여 캐쉬를 생성하는 방법이 가장 좋은 효율을 보인다. 특히 적은 캐쉬용량에서도 큰 차이를 보임으로서 본 논문에서 제안한 캐쉬 기법이 기존의 접근빈도에 기반한 방법들보다 TopicMap 탐색환경에서 유용함을 알 수 있다. 응답시간 그래프를 보면 50~150KB의 적은 메모리에서 정보기반 캐쉬기법이 LRU보다 긴 응답시간을 나타낸다. 이와같은 문제는 본 논문에서 제안한 페이지 교체 기법에서 차집합을 계산하는데 필요한 시간 때문이다. 하지만 캐쉬 크기가 늘어나고 hit ratio가 높아지면서 LRU의 응답속도보다 짧아지고 있다. 본 논문에서 제안한 교체기법이 다른 기법보다 계산에 더 많은 시간을 필요로 하지만 hit ratio의 증가에 따라 다른 기법과 비교할 때 충분한 응답속도를 보인다. 특히 모바일 환경과 같이 통신속도가 느리고 통신 비용이 큰 환경에서는 hit ratio를 높여 전송 횟수를 줄이고, 전송되는 캐쉬의 크기를 최소화하는 것이 효율적이다.

8. 결 론

본 논문에서는 지식맵의 효율적인 서비스 방안을 제안하고자 지식맵의 일종인 TopicMap을 이용한 탐색 및 캐쉬를 이용한 정보전송기법에 대해서 제안하였다. 기존의 TopicMap에 대한 연구는 네트워크 환경에서의 서비스에 대해서는 고려하고 있지 않으므로 네트워크 환경으로 구현할시 전체 TopicMap을 전송하거나 요구시마다 Topic을 전송하는 방식으로 구현하게 된다. 이 경우 심각한 네트워크비용의 낭비와 응답속도의 저하가 발생하게 된다. 따라서 본 논문에서는 사용자가 전체 TopicMap을 탐색하는 것이 아니라 관심을 가지는 부분에 대해서 탐색한다는 점, 그리고 그 탐색이 주로 정보의 연관성에 의해 이루어진다는 데에서 착안 정보의 연관성 중심으로 탐색을 지원하는 환경을 제안하고, 이 환경에서 연관된 정보의 캐쉬를 생성하여 전송하는 방법을 제안하였다. 특히 캐쉬를 생성함에 있어 기존에 다른 분야에서 사용되던 접근빈도에 의한 캐쉬가 아닌 TopicMap이 가지는 정보를 이용함으로써 TopicMap 탐색에 더욱 유용한 캐쉬를 생성할 수 있음을 증명하였다. 또한 캐쉬를 사용하지 않았을 경우에 비해 메모리 효율적인 사용과 빠른 응답속도를 가짐을 확인하였다. 특히 최근 모바일 네트워크환경이 급속히 확산되는 추세인데, 본 논문의 캐쉬 기법은 TopicMap의 크기에 상관없이 캐쉬 용량 만으로 TopicMap탐색을 지원하기 때문에 적은 메모리 기기를 가지는 모바일 장치를 효과적으로 지원할 수 있으며, 전체 TopicMap중 탐색에 관련된 부분만을 전송하기 때문에 고비용의 무선 네트워크비용을 절약할 수 있다는 점에서 모바일 네트워크 환경에서도 TopicMap 서비스 지원을 위한 유용한 기법이라고 생각한다. 캐쉬의 효율에 대해서는 더 많은 연구를 통한 개선이 이루어질 수 있다. 본 논문에서 실험에 적용한 클러스터

링 범위, 적용된 휴리스틱의 종류는 실험을 통해 정확도가 높은 캐쉬 생성과 캐쉬 생성 시간간의 절충점에 의해 결정된 것이다. 따라서 클러스터링 기법의 변환이나 factor 개선, 새로운 휴리스틱의 제안을 통해 캐쉬의 효율을 높일 수 있을 것이다. 또한 현재 본 논문의 캐쉬 교체기법에 있어서는 그래프적인 구조로 간주하여 외각 부분을 제거하는 기법을 사용하였으나 교체에 있어서도 정보의 연관성을 고려하는 기법을 생각해 볼 수 있다. 또한 캐쉬 생성 및 교체에 있어 정확도를 높임과 동시에 계산시간을 최소화하는 기법들에 대해서 아직은 충분히 고려되어있지 않으므로 이에 대한 연구가 필요하다. 이와 같이 전송과 관련된 캐쉬의 효율을 높이는 연구와 더불어 지식맵의 실질적인 서비스 구현을 위해서는 저장, 처리, 탐색에 이르기까지 전반적이 연구가 이루어져야 할 것이다.

참 고 문 헌

[1] Rafal Ksiezyk, "Answer is just a question [of matching Topic Maps]," XML 2000 Conference & Exposition.

[2] Steve Pepper, "Navigating Haystacks and Discovering Needles: Introduce the New Topic Map Standard," Markup Languages:Theory & Practice (1.4), 1999, pp.41-68.

[3] S. Staab, H.-P. Schnurr, R. Studer and Y. Sure, "Knowledge Processes and Ontologies," IEEE Intelligent Systems, 16(1), 2001.

[4] Dieter Fensel, Ian Horrocks, F. van Harmelen, D. McGuinness and P.F. Patel-Schneider, "OIL: An Ontology Infrastructure for the Semantic Web," IEEE Intelligent Systems, 2001.

[5] Michel Biezunski, Martin Bryan, Steve Newcomb, ISO/IEC 13250 TopicMaps.

[6] Steve Pepper, Graham Moore, "XML Topic Maps (XTM) 1.0," TopicMaps.Org.

[7] Hans Holger Rath, "Making Topic Maps more colourful," XML 2000 Conference & Exposition.

[8] Le Grand, B.,Soto, M, 2000, "Information management Topic Maps visualization," XML Europe 2000.

[9] Pascal Auillans, "Graph clustering for vary large topic maps," XML Europe 2001.

[10] Jia Wang, "A Survey of Web Caching Schemes for the Internet," ACM SIGCOMM '99.

[11] GeneSys, "World-Wide Web Caching," IEEE Comm. Mag. '97.

[12] Shaul Dar, Michael J. Franklin, Bjorn T. jonsson, Divesh Srivastava, Michael Tan, "Semantic Data Caching and Replacement," VLDB, 1996.

[13] J.S. Deogun, D.Kratsch and G.Steiner, "An Approximation algorithm for clustering graphs with domination diametral paths," Information Processing.

[14] J. A. Hartigan and M. A. Wong. A, "k-means clustering algorithm," Applied Statistics Vol 28, pp.100-108, bibtex, 1979.

정 준 원
정보과학회논문지 : 컴퓨팅의 실제
제 10 권 제 1 호 참조

민 경 섭
정보과학회논문지 : 컴퓨팅의 실제
제 10 권 제 1 호 참조

김 형 주
정보과학회논문지 : 컴퓨팅의 실제
제 10 권 제 1 호 참조