

디지털 도서관 환경에서 일관성과 최근성을 고려한 메타데이터 관리 방법

(Metadata Management Method for Consistency and Recency in Digital Library)

이 해 민 * 박 석 **
(Haimin Lee) (Seog Park)

요약 디지털 도서관은 정보검색 시스템과 데이터베이스 시스템의 통합 시스템으로 볼 수 있다. 동적으로 질의와 갱신작업이 일어나는 디지털 도서관 환경에서 메타데이터에 대해 기존의 트랜잭션 관리 기법을 사용하면 질의 처리 시간의 증가와 최근성 반영을 보장하지 못하는 문제점이 발생하게 된다. 또한 소스데이터와 메타데이터 사이에 지켜져야 하는 일관성이 규정되지 않기 때문에 질의 결과에 대한 불확실성을 초래한다.

본 논문은 인터넷 자원을 기술하는 메타데이터인 Dublin Core에 기초하여, 이에 접근하는 작업을 질의 트랜잭션과 갱신 트랜잭션으로 모델링하고 이를 효과적으로 관리할 수 있는 기법을 제시한다. 특히 기존의 트랜잭션 관리에서 고려되어 온 일관성 기준에서 고려하지 못했던 디지털 도서관에서의 질의 트랜잭션의 결과와 소스데이터베이스와의 일관성을 고려한 메타데이터 일관성을 제시한다. 또한 최근성 만족을 위해 질의 트랜잭션의 뷰 시점을 분석하고 메타데이터 일관성을 지키는 범위 내에서 최근성을 보장하는 관리 기법을 제안한다.

Abstract The Digital Library is the integrated system of Information Retrieval System(IRS) and Database Management System(DBMS). In the Digital Library environment where dynamic query and update processes are required, however, the existing transaction management methods cause the following problems. First, since the traditional consistency criteria is too restrictive, it causes increment of query processing time and cannot guarantee the reflection of recency. Second, query result could be unreliable because the consistency criteria between source data and metadata is not defined.

This paper models the access to metadata based on Dublin Core as query transactions and update transactions, and gives the efficient method to manage them. Particularly, this paper describes the consistency criteria of metadata which takes it into consideration the consistency between the result of query transaction and status of source data in the Digital Library, that is different from the consistency criteria in traditional transaction management. It also presents analysis of the view point of query transaction to reflect recency and proposes metadata management to guarantee recency within metadata consistency.

1. 서론

디지털 도서관은 정보검색 시스템과 데이터베이스 시스템의 통합 시스템으로 볼 수 있다[1][2]. 디지털 도서관 환경에서 일어나는 갱신과 질의 작업을 효과적으로 다루기 위해서는 두 시스템의 특징을 분석, 통합시키는 작업이 필요하다. 디지털 도서관을 구현한 예를 살펴보면 크게 세 가지 방법으로 두 시스템을 통합시켰다. 첫

* 이 논문은 한국전자통신연구원 이 수행하는 "21세기 DBMS 핵심 요소 연구 사업"결과물의 일부인.

† 비 회 원 : 한국교육학술정보원 학술연구정보화실 연구원
hmlcc@ms.keris.or.kr

** 종 신 회 원 : 서강대학교 컴퓨터학과 교수
spark@dblab.sogang.ac.kr

논문접수 : 1998년 8월 7일
심사완료 : 1999년 10월 5일

째, 두 시스템의 독립성을 인정하고 두 시스템사이의 갱신 및 질의 작업에 대한 일관성 유지를 위해 서버를 두는 방법으로 대부분의 디지털 도서관이 이 방법을 사용하고 있다. 이러한 서버는 그 기능이 특별히 명시되어 있지 않으며 응용 프로그램의 목적에 따라 구현된다. 상용 소프트웨어로써 이러한 서버를 일반적인 경우에 적용하려는 노력도 있다. 예를 들어, 정보검색 엔진 BRS (Bibliographic Retrieval Service)를 개발해 온 Dataware에서는 디지털 도서관을 겨냥하여 정보검색 시스템과 데이터베이스 시스템의 통합 형태인 토탈리콜 (Total Recall)을 개발하였다. 그러나 관계형 데이터베이스 시스템의 독립성을 인정한 채 통합 시스템을 추구하였으므로 통합 이전의 문제점을 근본적으로 극복하지는 못하였다. 둘째, 두 시스템을 완벽하게 결합시켜 하나의 독립적인 시스템으로 구성하는 경우가 있을 수 있지만 아직까지 일반화되지 않은 방법이다. 셋째, 두 시스템 중 하나의 시스템에서 다른 하나의 기능을 중점적으로 이용하는 방법으로써, 정보검색 시스템에서 제공하는 데이터 관리 기능을 이용하는 방법, 혹은 데이터베이스 시스템의 인덱스 관리를 통해 정보검색 시스템의 기능을 대신하는 방법이 있으나 각각 데이터에 대한 안정성 결여, 검색 시간의 지연 등의 문제점을 안고 있다. 정보검색 시스템과 데이터베이스 시스템의 성공적인 통합을 위해서는 두 시스템의 통합정도에 독립적으로 소스데이터와 메타데이터 사이에 지켜져야 할 일관성 기준과 일관성을 유지할 수 있는 방법이 필요하다.

본 논문은 디지털 도서관 환경에서 정보검색 시스템과 데이터베이스 시스템의 통합 시 메타데이터를 소스데이터에 일관되게 관리할 수 있는 일관성 기준을 정의하고 응용 분야에 맞게 최근성을 만족시킬 수 있는 메타데이터 관리 기법을 제시한다. 메타데이터는 정보검색 시 가장 먼저 접근되어야 하는 정보이며, 방대한 양의 데이터에 대응하는 메타데이터를 효과적으로 다루기 위해 메타데이터베이스를 구축한다. 효과적인 디지털 도서관은 데이터베이스 시스템과의 연동도 뛰어나야 할 뿐더러, 통합 시스템으로서의 요구사항을 만족시켜야 한다. 즉, 정보검색에서 좋은 성능을 보여야 하고 안정된 형태의 데이터가 제공되어야 한다. 이에, 메타데이터베이스 관리를 위해 기존의 데이터베이스 시스템에서 사용하던 트랜잭션 관리를 메타데이터의 특징에 맞게 변형한다.

메타데이터 관리를 위해 트랜잭션은 메타데이터에 접근하는 목적에 따라 질의 트랜잭션과 갱신 트랜잭션으로 분류하고, 메타데이터 관리를 위해 다음 두가지 사항

을 고려한다. 첫째는, 일관성 문제이다. 트랜잭션과 메타데이터의 특징을 분석하여 일관성 기준을 잡아도 이것은 기존의 데이터베이스 관리에서 다루는 일관성 기준을 벗어나지 않는다. 또한 정보검색이 목적인 환경을 고려하면, 메타데이터와 소스데이터의 일관성 또한 검토되어야 한다. 둘째, 최근성 반영의 문제이다. 최근성은 질의 결과가 최근의 변화를 얼마나 반영하는가를 나타낸다. 이에 질의 트랜잭션의 결과값이 되는 질의 트랜잭션의 뷰를 세가지 시점으로 나누어 분석하고, 메타데이터 일관성을 지킬 수 있는 수준의 판독 시점 뷰를 제시한다.

본 논문의 제 2장에서는 연구 배경이 되는 디지털 도서관 프레임워크와 Dublin Core를 보이고, 제 3장에서는 트랜잭션 일관성 기준과 메타데이터 일관성 기준을 정의하며, 제 4장에서는 최근성 반영을 위해 질의 트랜잭션의 뷰 시점을 분석한 후 메타데이터 일관성을 보장하는 관리 기법을 제시한 후 기존 알고리즘과의 성능평가를 보인다. 제 5장에서는 결론을 맺는다.

2. 연구배경

본 논문은 디지털 도서관 환경을 가정한다. 디지털 도서관의 프레임워크를 설정할 때에는 메타데이터의 집합을 먼저 정의하고 시작한다. 본 논문은 현재 웹문서 표준 메타데이터로 자리잡은 Dublin Core를 사용한다 [3]. 또한 Dublin Core를 효과적으로 구성할 수 있는 객체지향 구조인 Warwick Framework [6] 개념을 사용한 에이전트 기반의 디지털 도서관 프레임워크를 배경으로 한다.

2.1 Dublin Core와 Warwick 프레임워크

디지털 도서관에서의 메타데이터란 객체, 즉 문서를 효율적으로 일관성 있게 검색하기 위한 데이터에 대한 설명 데이터이다. 단순한 정보의 인덱스가 아니라 메타데이터 자체로서 데이터베이스가 구성되어지고 일관성 있는 관리가 필요한 데이터이다.

본 논문은 디지털 도서관에서 사용할 메타데이터의 기본적 구성으로 Dublin Core를 택하였다. Dublin Core는 네트워크를 통한 검색, 여러 문서에 대한 검색 시 필요한 메타데이터에 대한 표준안으로서, 기존에 존재하던 단순 패턴 부합 방식을 개선하기 위해 제안된 것이다 [4]. 또한 현재 웹문서를 표현할 수 있는 꼭 필요한 요소를 가지고 있으며, 물리적인 단위가 아닌 논리적 단위이므로 확장성을 고려할 여지를 마련해준다. Dublin Core는 표 1과 같은 15개의 요소로 이루어져 있다 [5]. 이 요소들은 최소 단위의 정보만을 요구하며 기본적으로

모두 기술적인(descriptive) 내용을 제시한다.

표 1 Dublin Core 15개 요소

요소	의미
Title	제목
Author or Creator	저자
Subject and Keyword	주제를 나타내는 문장이나 키워드
Description	객체에 대한 텍스트 형태의 설명
Publisher	출판을 책임지는 기관
Other Contributors	저자에게 도움을 준 사람이나 기관
Date	문서가 만들어진 시점
Resource Type	장르
Format	구성의 형태
Resource Identifier	식별을 위한 문자열이나 숫자
Source	생성된 근본 객체를 나타내는 지시자
Language	주로 사용하는 언어
Relation	계층 등을 표현하는 다른 문서와의 관계
Coverage	시간적, 공간적 제약 사항
Right Management	저작권, 보급 서버 등에 관한 정보

Dublin Core는 가장 보편적이고 단순한 15가지 요소를 나타내는 표준 메타데이터를 제공해주었고, 이것을 통해 많은 메타데이터 집합이 단순화되었으며, 또한 검색에 있어 편리함이 제공되었다. 그러나 많은 사람들은 Dublin Core가 제공해주는 정보 이외의 다른 형태의 정보도 제공받기를 원하게 되었다. 따라서 기존에 존재하던 Dublin Core와 그 외의 많은 메타데이터의 표준들을 확장성과 융통성 있게 통합하는 연구가 활발히 이루어지게 되었고 그것에 대한 결과로서 나타난 것이 컨테이너/패키지 구조를 가지는 Warwick 프레임워크이다[6] [7].

Warwick 프레임워크는 논리적인 구조로서, 크게 두 가지 부분으로 이루어져 있다. 첫 번째는 패키지(package)로서 이것은 서로 관련되어 있는 일련의 메타데이터의 집합을 의미한다. 즉 서로 같은 의미를 가지는 메타데이터의 모임 하나를 하나의 패키지에 부합시킬 수 있는 것이다. 두 번째로 컨테이너(container)인데 이것은 서로 다른 메타데이터들의 집합인 패키지들을 하나의 단위 구조로 묶어 놓은 것이다. 이렇게 함으로써 서로 다른 메타데이터 집합들을 통합하게 된다.

이러한 컨테이너/패키지 구조는 그림 1에서 볼 수 있다. 그림 1은 Dublin Core와 MARC를 허용하는 구조

에서 그 외의 다른 메타데이터 집합까지 포함하는 구조를 나타낸다. 이러한 구조는 Warwick 프레임워크의 확장을 보장한다.

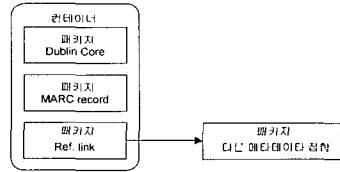


그림 1 컨테이너/패키지 구조

2.2 에이전트 기반의 디지털 도서관 프레임워크

이번 절은 Warwick 프레임워크에 기초한 에이전트 기반의 디지털 도서관 프레임워크를 설명한다. 메타데이터 집합은 Dublin Core를 사용하며, 그 개념적인 프레임워크는 Dublin Core를 잘 표현할 수 있는 Warwick 프레임워크를 도입하였다. 또한, 클라이언트/서버 중심에서 에이전트 중심으로 가는 변화에 맞추고 디지털 도서관의 확장성 보장 등의 장점을 취할 수 있도록 하기 위하여 그림 2와 같은 구조를 갖는 에이전트 기반의 프레임워크를 제시하였다[8] [9].

에이전트 기반의 디지털 도서관 프레임워크는 정보검색 시스템과 데이터베이스 시스템을 통합할 때 고려되어야 하는 논리적 모듈을 에이전트 형태로 구분하여 나타낸 것으로 특정 에이전트의 기능을 정보검색 시스템과 데이터베이스 시스템의 통합시 두 시스템 어느 곳에 포함되어야 하는가를 명시한 것은 아니다.

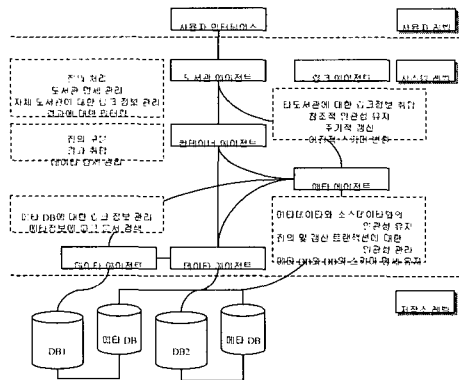


그림 2 디지털 도서관 프레임워크

메타에이전트의 기능 중에서 트랜잭션 관리를 중심으로 구조를 살펴보면 그림 3과 같다. 새로운 문서는 분석

기에 의해 정형화된 형태로 처리되어 갱신 트랜잭션으로서 트랜잭션 관리자에 들어간다. 여러 클라이언트로부터 들어오는 질의들은 문맥이나 의미 사진을 이용하여 정형화시키는 질의 처리기를 통해 질의 트랜잭션으로서 트랜잭션 관리자에 들어간다. 메타데이터는 사용자의 메타데이터 구성 도구에 의해 작성된다. 접근해야 하는 메타데이터 집합이 갱신, 질의 트랜잭션과 함께 트랜잭션 관리자에게 주어진다.

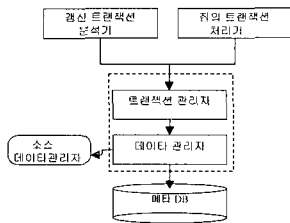


그림 3 메타에이전트의 트랜잭션 관리 구성도

트랜잭션 관리자는 병행 수행되는 질의와 갱신을 스케줄링하고 로크를 관리한다. 한 번 로크가 허용되면, 트랜잭션 관리자는 그 데이터에 대해 데이터 관리자에게 알린다. 데이터 관리자는 요구된 판독과 갱신 등의 연산을 메타데이터에 대해 수행하며 문서에 대한 처리 정보를 돌려준다.

그러면, 사용되는 중요 용어 정의를 내리고 메타데이터 관리를 위한 기본적인 가정을 한다.

- 소스데이터(source data) : 전통적으로 데이터라 부른다. 디지털 도서관 환경에서는 문서라고 볼 수 있고, 소스데이터는 소스데이터베이스에 구축된다.
- 메타데이터(metadata) : 메타데이터는 소스데이터에 대한 설명 데이터이다. 기본적으로는 주제어지만 소스데이터의 위치 등을 나타낼 수도 있다. 기존의 데이터베이스 시스템의 카탈로그도 메타데이터의 한 형태이다. 각 메타데이터의 요소는 소스데이터에 의존적으로 소스데이터에 대한 설명 데이터이다.
- 메타데이터 집합(metadata set) : 메타데이터 집합은 메타데이터가 가지는 요소를 말한다. 응용에 따라 메타데이터 집합에서 필요로 하는 요소가 다르다. 예를 들면, 멀티미디어 정보를 주로 가지는 소스데이터에 대한 메타데이터는 이미지 화일의 타입, 오디오 화일의 타입 등의 요소를 가져야 한다. 웹 문서가 소스데이터인 디지털 도서관에서는

소스데이터인 문서의 제목, 저자 등의 요소가 필요하다. 본 논문에서는 웹 문서 표준 메타데이터 집합인 Dublin Core를 사용한다.

- 질의 트랜잭션(query transaction) : 본 논문의 질의 트랜잭션은 메타데이터에 접근하는 질의 작업을 의미한다. 전체적 시스템에서 실행될 때 질의 작업은 사용자의 질의 생성부터 메타데이터에 대한 접근, 그 결과를 바탕으로 소스데이터에 대한 접근, 결과의 도출, 사용자에게 전달하는 과정을 모두 포함한다. 이 중 메타데이터에 대한 접근을 트랜잭션으로 모델링 한 것(그림 4)이 본 논문에서 다루는 질의 트랜잭션이다.

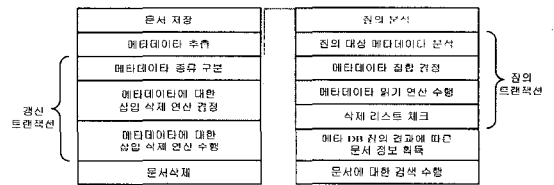


그림 4 메타에이전트의 트랜잭션 관리 구성도

- 갱신 트랜잭션(update transaction) : 질의 트랜잭션과 마찬가지로 전체 시스템에 대한 갱신 작업 중 메타데이터에 대한 접근을 트랜잭션으로 모델링 한 것(그림 4)이 본 논문에서 다루는 갱신 트랜잭션이다.

다음은 메타데이터 관리를 위한 기본적인 가정이다.

- 먼저, 갱신 트랜잭션과 질의 트랜잭션에 대한 가정을 내린다. 갱신 트랜잭션의 연산은 삽입(insert)과 삭제(delete)만 존재한다. 문서에 대한 새로운 첨가는 삽입연산이, 문서에 대한 삭제는 삭제연산이 대응되고 문서에 대한 변화, 즉 갱신은 삽입연산 후 삭제연산으로 이루어진다. 질의 트랜잭션은 판독 연산만이 존재한다. 이러한 두 트랜잭션의 빈도수가 매우 중요한 사항이다. 정보 검색 시스템에서 질의 트랜잭션의 빈도는 갱신 트랜잭션보다 훨씬 높다. 또한 질의 트랜잭션은 사용자의 임의적인 생성에 의해 어느 정도의 시간을 필요로 하는지 예측하기 힘들다. 그리고 질의 트랜잭션은 메타데이터베이스의 하나의 데이터에 대해 단 한 번 판독한다. 즉, 재판독이 존재하지 않는다. 또한 두 트랜잭션 모두 철회되지 않고, 질의 트랜잭션의 경우 일관성과 최근성을 보장해야 하

는 요구사항을 가진다. 두 트랜잭션에 대한 흐름은 그림 4와 같다.

- 메타데이터는 그 구조와 상관없이 종속성이 존재하지 않는다. 메타데이터는 다른 메타데이터로부터 유도된 데이터가 아니라는 의미이다. 이러한 비종속성은 메타데이터에 대한 관리 기법을 개발할 때 중요한 특징이 된다. 전통적인 도서관에 사용하던 표준 메타데이터인 MARC이나 다른 다양한 메타데이터는 저장소 레벨에서 하나의 엔터티가 되는데 각 엔터티는 다른 데이터로부터 유도되는 성격을 갖지 않는다. 응용 프로그램 레벨에서는 인덱스 생성 등으로 인한 의존성이 존재할 수 있으나 데이터에 대한 기술적인 데이터로서 저장소 레벨의 종속성은 존재하지 않는다.¹⁾

3. 일관성 기준

이번 장에서는 Dublin Core를 그 특성에 따라 두 가지로 분류하여 모델링 함을 보인다. 그리고 관리 기법의 기준이 될 수 있는 일관성에 대해 논한다. 기존에 연구되어 오던 일관성 기준을 트랜잭션 일관성이라 하고, 정보검색 시스템에서 메타데이터에 대한 질의 결과와 소스데이터 사이에 지켜져야 하는 일관성을 메타데이터 일관성이라 한다.

3.1 Dublin Core의 분류와 메타데이터 모델링

Dublin Core는 웹문서에 대한 표준 메타데이터 집합으로 15개의 요소를 가진다. 이를 그림 5와같이 크게 두 부분으로 분류한다.

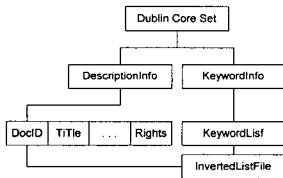


그림 5 메타데이터 분류

- DescriptionInfo : 문서를 정의하고 차별화하는 요소들의 집합이다. 이러한 메타데이터는 하나의 문서당 하나의 값을 갖는다.

- KeywordInfo : 여러 문서와 대응되는 요소들의 집합이다.

메타데이터를 두 가지로 분류하는 기준은 문서 정보와 메타데이터 요소 사이의 관계이다. "DescriptionInfo"에 해당하는 요소들의 경우, 하나의 문서에 의존된 데이터로서 문서의 식별자를 기본키로 갖는 테이블로 구성될 수 있다. "KeywordInfo"의 경우는 하나의 키워드에 여러 문서 식별자가 대응되는 경우로서 키워드를 중심으로 한 반전-리스트로 구성될 수 있다. 후자의 경우 반전-리스트로 구성하는 이유는 정보검색시 빠른 검색을 위함이다. 검색 시간의 단축을 위하여 15개의 모든 요소를 반전-리스트로 구성하는 것을 고려할 수 있으나 "DescriptionInfo"집합에 해당하는 요소들의 특성은 하나의 문서에 하나의 값이 대응하기 때문에 오히려 공간상의 낭비를 초래하여 반전-리스트를 구성하는 의미가 없어진다. "KeywordInfo"의 경우, 대부분의 디지털 도서관이 사이트마다 각각의 성격을 지니게 되므로 하나의 사이트에 존재하는 문서는 주제어 포함에 있어서 그 유사성이 높으므로 검색 속도가 빠르고 저장공간도 줄일 수 있다. Dublin Core를 분류하여 모델링 하는 것은 검색 성능 면에서도 빠르고 정확한 결과를 기대할 수 있으며 공간 절약 면에서도 중복을 피할 수 있다.

3.2 트랜잭션 일관성과 메타데이터 일관성

정보검색 시스템에서 소스데이터와 메타데이터 사이의 관계에 대한 정의를 내리기 위해 일관성 기준을 트랜잭션 일관성 기준과 메타데이터 일관성 기준으로 분류한다.

메타데이터 관리 기법에는 메타데이터에 접근하는 트랜잭션의 성격만을 고려해서는 안 된다. 메타데이터는 일반적인 데이터와는 그 성격이 다르기 때문이다. 위에서 밝혔듯이, 메타데이터는 문서에 대한 설명 데이터이다. 그러므로 메타데이터의 특성을 먼저 살펴보기로 한다.

데이터 관리시 가장 중요하게 살펴볼 점은 그 데이터가 어떤 종류이든 데이터 사이의 종속성이다. 메타데이터는 문서에 대한 설명 데이터이므로 서로간의 종속성이 존재하지 않는다. "DescriptionInfo" 집합에 속하는 메타데이터 요소는 특히 하나의 문서당 하나의 값을 가지므로 각 문서에 대한 종속성은 전혀 존재하지 않는다. "KeywordInfo" 집합에 속하는 메타데이터 요소도 서로간의 종속성이 없다. 다만, "KeywordInfo" 집합에 속하는 요소는 여러 문서에 중복적으로 존재하는 설명정보이다. 즉, 문서 사이의 관계성과 메타데이터 사이의 관계성은 없지만, 하나의 설명 정보에 여러 문서가 대응되

1) 종속성이 존재하는 데이터의 예는 다음과 같다: 기온을 시간순으로 기록하고 이의 변화량을 기록하는 응용의 경우, 변화량 데이터는 시간당 기록한 기온의 차에 종속되어 기록된다. 메타데이터의 경우 저장소 레벨에서 보았을 때 문서에 대한 기술적인 데이터이므로 서로간의 종속성이 존재하지 않는다. 즉, 다른 메타데이터에 의해 유도된 데이터가 아니다.

게 된다. 서로간의 연관성이 전혀 없는 데이터 사이의 관리기법은 최소한의 일관성 유지만을 요구한다. 그러나, 메타데이터 사이의 일관성 문제와 더불어, 메타데이터와 문서 사이의 일관성, 좀 더 자세히 말하면 질의 결과와 문서 사이의 일관성 문제를 해결해주는 관리 기법이 필요하다. 이에, 트랜잭션 일관성과 메타데이터 일관성에 대해 분석한다.

□ 트랜잭션 일관성

전통적으로 연구가 되어오던 일관성 기준을 정보검색 시스템의 메타데이터 관리에 필요한 일관성 기준과 구별하여 트랜잭션 일관성(transactional consistency)이라 한다. 트랜잭션 일관성 기준은 트랜잭션의 스케줄링 기법에 따라 각 트랜잭션이 가지는 데이터 값에 대한 일관성 기준이라고 볼 수 있다[10]. 메타데이터 관리를 위해서는 단계 0, 1, 2, 3 일관성 기준 중에 단계 0 일관성 기준을 사용한다. 단계 0 일관성의 경우 트랜잭션 사이의 종속성이 존재하지 않는 환경에서만 일관성이 지켜질 수 있다. 단계 1 일관성 기준은 기록-기록 종속성(write-write dependency)을 고려한 기준이다. 단계 2 일관성 기준은 기록-기록, 기록-판독 종속성(write-write, write-read dependency)을, 단계 3 일관성 기준은 기록-기록, 기록-판독, 판독-기록 종속성(write-write, write-read, read-write dependency)을 고려한 기준이다. 단계 3 일관성 기준은 가장 제약이 심한 일관성 기준으로서 트랜잭션이 서로의 미완료 데이터(dirty data)를 읽지 않는다. 그러나 이러한 일관성 기준은, 데이터 사이의 종속성이 존재하지 않는 메타데이터 관리에 사용하기에는 제약이 심하다. 그러므로 메타데이터에 접근하는 트랜잭션을 위한 일관성은 단계 0 일관성을 적용한다. 단계 0 일관성을 적용하는 근거는 다음과 같다.

메타데이터의 특성에서 밝혔듯이, 메타데이터 요소 사이에는 종속성이 없다. 또한, "KeywordInfo"의 경우, 각 주제어에 대한 반전-리스트는 증가, 감소의 과정을 거친다. 이러한 이유에서 트랜잭션의 고립(isolation) 특성은 고려할 필요가 있다. 고립 특성은 판독과 기록 사이의 종속성에 기초한다. 기록-기록, 기록-판독, 판독-기록 종속성에 대해 알아본다.

데이터 요소 사이의 종속성이 존재하지 않은 경우의 일관성 기준에 대해 [1]에서는 다음과 같이 트랜잭션 일관성 기준을 정의하고 있다.

- ▷ 기록-기록 종속성(write-write dependency) : 갱신 트랜잭션이 기록 전용이고, 역전-트리에 대한 갱신은 추가의 성격을 가진다. 이러한 갱신은 순

서에 상관없이 동일한 결과를 가진다(idempotent 하다). 그러므로 이러한 종속성은 무시될 수 있다.

- ▷ 기록-판독, 판독-기록 종속성(write-read, read-write dependency) : 이것은 질의와 갱신이 병행적으로 수행된다는 것을 의미한다. 그런데 다음 세 가지 이유로 기록, 판독의 종속성은 무시될 수 있다. 첫째, 갱신은 각 메타데이터에 대해 단위시간에 한 번 갱신을 행한다. 둘째, 질의는 메타데이터를 단위시간에 한 번 판독한다. 셋째, 메타데이터 사이의 무결성 제약은 존재하지 않는다.

그러므로 질의 트랜잭션과 갱신 트랜잭션의 병행적인 수행 순서는 무시될 수 있다.

이러한 고립 특성을 고려했을 때 메타데이터에 대한 관리기법은 단계 0 일관성 기준에서 행해진다. 즉, 단기로크(래치 : latch)로서 데이터의 최소한의 일관성을 지킨다. 그보다 더 강력한 일관성 기준은 메타데이터에 접근하는 트랜잭션과 메타데이터 자체의 특성을 고려하면 오히려 성능상의 저하를 가져옴을 알 수 있다. 그러므로 메타데이터 관리의 트랜잭션 일관성은 단계 0 일관성 기준에 따른다[1].

□ 메타데이터 일관성

전통적인 트랜잭션 일관성과 비교하여, 정보검색을 목적으로 하는 시스템에서 질의의 결과와 문서 사이에서 지켜져야 하는 일관성을 메타데이터 일관성(metadata consistency)이라 정의한다.

- ▷ 메타데이터 일관성(metadata consistency) : 질의 트랜잭션의 결과가 문서의 존재성을 보장할 때, 메타데이터 일관성을 가진다고 한다.
- ▷ 문서의 존재성(document existence) : 질의 트랜잭션의 결과가 소스데이터에 실제로 존재할 때 문서의 존재성을 가진다고 한다.

즉, 질의 트랜잭션의 결과가 메타데이터 일관성을 가진다는 의미는 질의 결과의 문서가 실제로 소스데이터베이스에 존재한다는 것이다. 존재하지 않는 문서에 대해서 존재한다고 하지 않아야 한다. 메타데이터 일관성을 가진다는 의미가 질의 트랜잭션의 결과와 소스데이터의 상태가 동일함을 의미하는 것은 아니다.

4. 메타데이터 일관성과 최근성을 보장하는 관리 기법

정보검색 시스템에서 메타데이터를 관리할 때 고려할 사항은 메타데이터 일관성 외에도 최근성이라는 것이 있다. 메타데이터 일관성은 질의 트랜잭션의 결과가 문서의 존재성을 보장한다는 의미이다. 그러나 메타데이터

트랜잭션의 경우 일관성과 최근성을 보장해야 하 일관성을 지키기 위해 최근성을 간과한다면 정보 검색 시스템의 결과로서 그 가치를 잃게 된다. 따라서, 최근성을 반영하는 질의 트랜잭션의 뷰 시점을 세 가지로 나누어 정의하고, 메타데이터 일관성을 지키면서 판독 시점 뷰를 가지는 관리기법을 제안한다.

4.1 질의 트랜잭션의 뷰 시점 분석

이제, 질의 트랜잭션 결과의 최근성을 고려한다. 최근성은 질의 트랜잭션의 뷰 시점(view point)에 관련한다. 다음 그림 6과 같이 질의 트랜잭션이 어느 시점의 뷰를 가져야 하는지 크게 세 가지로 분류한다.



그림 6 질의 트랜잭션의 뷰 시점

- 시작 시점(start point) : 질의 트랜잭션이 시작 시점의 뷰를 가지면 메타데이터 일관성을 지키기 위해 시작 시점 이후에 일어난 갱신이나 삭제를 막아야 한다. 이는 일관성을 만족시킬 수 있지만 최근성 반영을 전혀 고려하지 않은 결과를 얻게 된다.
- 완료 시점(end point) : 질의 트랜잭션이 완료 시점의 뷰를 가지면 가장 최근의 결과를 반영한다는 의미이다. 즉, 질의 트랜잭션의 완료 시점의 메타데이터의 상태를 반영하는 것이다. 그러나 완료시점의 작업부하와 완료작업 중에 일어나는 삭제연산을 막기 위한 로크가 필요하게 된다. 이러한 삭제를 고려하지 않으면 메타데이터 일관성을 지킬 수 없다.
- 판독 시점(read point) : 질의 트랜잭션이 데이터를 판독하는 시점의 값을 그 결과로 반영하면 판독 시점 뷰를 가진다고 한다. 즉, 어떠한 데이터라도 판독 시점의 값을 결과로 갖는다. 이후에 일어난 어떠한 삽입이나 삭제도 질의 결과에 반영되지 않는다.

판독 시점의 뷰는 가장 최근의 결과는 아니지만, 재판독이 없는 가정 하에 판독 시점의 뷰를 갖도록 하면, 시작 시점의 뷰보다는 훨씬 최근의 정보를 얻을 수 있고, 메타데이터의 삭제는 곧바로 반영되어 다른 트랜잭션의 최근성 반영을 방해하지 않고, 문서 데이터의 삭제 시점을 구분하여 메타데이터 일관성을 만족할 수 있다.

4.2 메타데이터 일관성과 판독 시점 뷰를 갖는 관리 기법

앞서 밝혔듯이, 트랜잭션 일관성 기준을 지키기 위해서 단계 0 일관성 기준에 따라 래치를 사용한다. 즉, 판독, 기록을 위하여 단기간 로크를 사용한다. 만약, 트랜잭션 일관성만을 고려한다면 래치만으로 관리할 수 있을 것이다. 대용량의 데이터베이스에 대해 질의 트랜잭션이 주로 일어나는 환경에서 트랜잭션 일관성만을 고려하는 경우는 단순 래치 방법을 사용하는 것이 제안된 바 있다[1][2]. 그러나 3절에서 분석한 디지털 도서관 환경에서는 메타데이터 일관성을 지키고 질의 트랜잭션의 결과에 최근성을 반영하기 위한 관리 기법이 필요하게 된다.

먼저, 메타데이터 일관성을 지키기 위해 갱신 트랜잭션의 연산을 삽입과 삭제로 구분한다. 그런데 최근성의 시점을 판독 시점이라고 했기 때문에 판독 이후에 일어난 삽입과 삭제를 고려해본다.

판독 이후에 일어난 삽입은 재판독이 없이 때문에 질의 결과에 반영되지 않는다. 그러나, 문서의 존재성에 영향을 끼치지 않기 때문에 갱신 트랜잭션의 삽입은 아무런 제약 없이 행해질 수 있다. 그러나 판독 이후에 일어난 삭제는 질의 결과의 메타데이터 일관성을 깨뜨리는 연산이기 때문에 특별한 고려가 필요하다. 단순히 래치만을 사용하는 관리기법이 그대로 적용될 수 없는 이유도 된다.

다음 세 가지 방법을 고려해본다.

(가) 단순 래치 방법

판독 이후의 삽입이나 삭제 모두를 허용하는 방법이다. 이러한 경우, 삭제연산은 판독 이후에도 소스데이터에 대해 반영하게 된다. 이는 앞서 밝힌 바와 같이 메타데이터 일관성을 지킬 수 없는 방법이다.

예1 : 다음 그림 7은 질의 트랜잭션 Q1이 데이터 a에 대한 판독을 한 후에 갱신 트랜잭션 U2의 삭제연산 d(a)를 허용한다. 그러므로 질의 트랜잭션의 완료 시점에서 자신이 판독한 결과에 대한 문서의 존재성을 보장할 수 없게 된다. ■

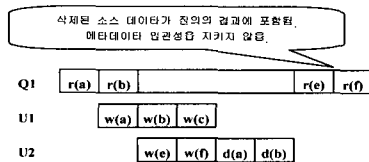


그림 7 단순 래치 방법

판독 이후의 삭제연산을 충돌이 일어나는 질의 트랜잭션의 완료시까지 지연시키는 방법이다. 그러나 정보검색 시스템을 살펴보면, 질의 트랜잭션에 비해 갱신 트랜잭션의 비율은 매우 떨어진다. 또한, 동적으로 질의 트랜잭션과 갱신 트랜잭션이 발생하기 때문에 갱신 트랜잭션과 충돌이 일어나는 질의 트랜잭션이 언제 끝날지 예상할 수 없으므로 지연을 언제까지 시킬 수 있는가의 문제점이 발생한다. 또한, 최근성의 문제점도 발생한다. 하나의 질의 트랜잭션과 충돌이 일어난다 하여 갱신 트랜잭션의 삭제연산을 지연시키면, 실제로 일어났어야 최근성을 반영할 수 있었던 갱신 트랜잭션 이후에 들어온 질의 트랜잭션의 최근성 반영을 방해하게 된다.

예2 : 그림 8을 보면 그림 7에서의 삭제연산을 충돌이 일어나는 질의 트랜잭션의 완료시까지 지연시킴을 볼 수 있다. 삭제연산을 지연시키면 판독 이후의 삭제연산이 이뤄지지 않기 때문에 메타데이터 일관성을 지킬 수 있으나 갱신 트랜잭션 U2는 언제까지 지연되어야 하는지 알 수 없게 된다. 질의 트랜잭션의 빈도수가 매우 높기 때문에 U2가 지연되고 있는 동안 또다른 질의 트랜잭션과 충돌이 일어나게 될 가능성이 많아진다. 또 하나의 문제점은 다른 질의 트랜잭션의 최근성 반영을 방해한다는 데에 있다. 질의 트랜잭션 Q1의 r(e) 직후에 생성된 질의 트랜잭션은 원래 U2의 삭제연산을 반영해야 최근성을 갖는데 이와 같은 삭제연산의 지연은 다른 질의 트랜잭션의 최근성 반영을 방해한다. ■

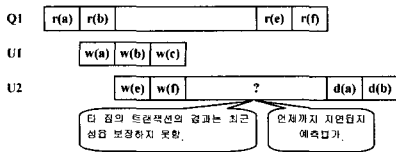


그림 8 삭제연산을 지연시키는 방법

(다) 메타데이터의 삭제 시점과 소스데이터의 삭제 시점을 달리하는 방법

메타데이터와 소스데이터의 삭제연산 시점을 달리한다. 즉, 삽입은 단순 래치에 기초하여 아무런 제약 없이 행한다. 단, 삭제연산의 경우, 메타데이터에 대한 삭제는 논리적으로 행하고, 실질적인 소스데이터에 대한 삭제는 (나)와 같은 시점에 행하게 한다. 여기에서, 메타데이터에 대한 삭제를 논리적으로 하는 이유는, "KeywordInfo" 집합의 경우, 반전-리스트로 구성이 되기 때문이다. 반전-리스트로 구성하면 검색 시간이 빨라지지만 삭제연산은 비용이 많이 들게 된다. 그러므로 제거-

리스트(purged-list)를 두어 마크만을 행한다. 즉, 삭제연산은 제거-리스트에 마크함으로써 행해졌다고 할 수 있다. 후에 시스템의 부하가 매우 적은 때나 메타데이터를 접근하고 있는 트랜잭션이 존재하지 않을 때 실질적인 메타데이터에 대한 삭제를 한다. 질의 트랜잭션은 래치에 기준하여 판독 연산을 행하고, 질의 결과를 돌려줄 때 제거-리스트 중 자신의 판독 이전에 일어난 삭제만을 반영한다.

문서가 분석되고 난 후 질의와 갱신 트랜잭션에 의해 판독되고 쓰여질 메타데이터는 알려져 있고, 메타데이터는 단지 한 번만 접근된다. 트랜잭션이 도착하였을 때, 트랜잭션 관리자는 트랜잭션에 의해 접근될 모든 데이터를 명시하고 모든 활성 갱신 트랜잭션(active update transaction)에 대한 정보를 유지한다. 질의 트랜잭션이 도착하였을 때, 모든 활성 갱신 트랜잭션과 충돌 체크를 행한다(이는 공통적으로 접근하는 메타데이터가 존재하는지 여부를 확인하여 확인된다). 만약, 질의와 병행적으로 수행되고 있는 갱신 트랜잭션 사이에 공통적인 메타데이터가 있다면, 그러한 갱신 트랜잭션의 식별자는 질의 트랜잭션의 충돌 집합에 추가된다.

트랜잭션 관리자는 순서대로, 갱신 트랜잭션의 연산을 각 데이터의 대기큐에 위치시킨다. 한 번 데이터가 갱신되고 기록 래치가 풀리면, 질의 트랜잭션은 그 데이터에 대한 래치를 얻을 수 있고, 판독한다. 이러한 방법은 단계 0 일관성에 기초하여 대기시간을 현저하게 줄이고 데드락이 발생하지 않는다는 장점을 가진다.

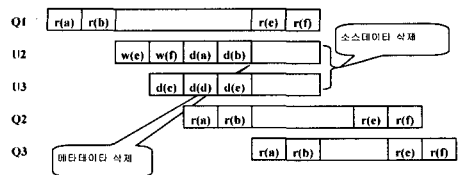


그림 9 메타데이터의 삭제 시점과 소스데이터의 삭제 시점을 고려한 방법

예3 : 그림 9와 같은 경우에 갱신 트랜잭션 U2와 U3의 삭제연산 d(a), d(b), d(c), d(d), d(e)는 제거-리스트에 마크만 된다. 질의 트랜잭션 Q1의 경우, 완료시에 데이터 a, b에 대한 판독 이후의 a, b에 대한 삭제는 반영하지 않는다. 질의 트랜잭션 Q2, Q3는 Q1에 의한 삭제연산의 지연으로 인한 최근성 반영 방해를 받지 않는다. 소스데이터에 대한 삭제 메시지는 충돌이 일어나는 질의 트랜잭션의 완료 이후로 미뤄진다. 그러므로 모든 질

있음을 보장할 수 있다. ■

4.3 비교분석 및 성능평가

제안된 메타데이터 관리 기법과 전통적인 트랜잭션 관리 기법, 단순 래치 기법을 성능평가로 비교한다. 성능평가 도구는 SLAMII[11]를 사용하였다. 성능평가는 다음과 같은 목표를 가진다. 첫째, 기존의 전통적인 트랜잭션 관리 기법보다 평균 대기 시간, 질의 응답 시간 면에서 절약을 할 수 있다. 둘째, 성능면에서 단순 래치 방법에 떨어지지 않으면서 메타데이터 일관성을 지킨다.

성능평가는 다음과 같은 사항에 대해 행한다.

- 평균 대기 시간
- 질의 응답 시간
- 메타데이터 비일관성 비율
- 잘못된 삭제 연산 비율

전통적인 트랜잭션 관리 기법과 단순 래치 기법, 그리고 제안하는 메타데이터 관리 기법을 평균 대기 시간, 질의 응답 시간 측정을 통해 다음과 같은 두 가지를 보인다. 첫째, 평균 대기 시간과 질의 응답시간 면에서 전통적인 트랜잭션 기법보다 현저히 높은 성능을 낸다는 것과 둘째, 메타데이터 일관성은 위한 부분이 첨가되었지만 단순 래치 기법에 비해 성능이 떨어지지 않는다는 것이다. 그리고 메타데이터 비일관성 비율과 잘못된 삭제 연산 비율을 측정하여 단순 래치 기법에서 문제점으로 지적한 사항에 대해 제안하는 관리 기법이 어느 정도의 효과를 나타낼 수 있는지 평가한다.

전통적인 트랜잭션 관리 기법은 두단계 로크 기법(2PL : 2 Phase Locking)을 사용한다.

성능평가에 필요한 매개변수는 표 2와 같이 일반적인 수치를 따르며[12], 갱신 트랜잭션의 연산 비율을 낮춤으로써 갱신의 비율이 적은 환경임을 반영하였다.

표 2 성능평가를 위한 매개변수

매개변수	설정된 값
각 연산이 CPU에 머무는 시간	1 ms
디스크 입출력에 걸리는 시간	2 ms
트랜잭션이 취소되는 데 걸리는 시간	100 ms
데이터베이스의 크기	1000 pages
갱신 트랜잭션의 기록 연산 비율	50 %
갱신 트랜잭션의 크기 min/max	4/20 ms
질의 트랜잭션의 크기 min/max	5/100 ms
트랜잭션의 개수	100
다중프로그래밍 레벨	100

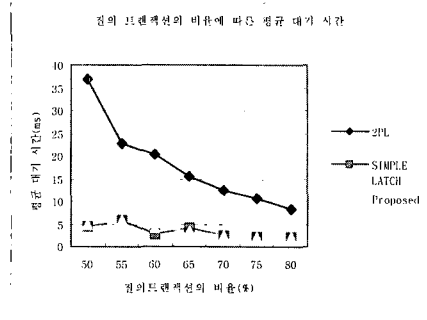


그림 10 질의 트랜잭션의 증가에 따른 평균 대기 시간

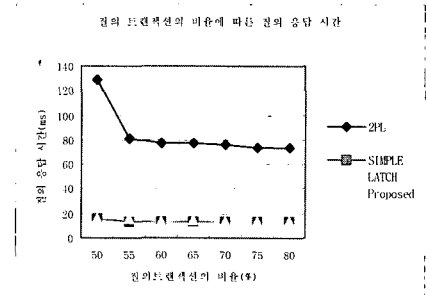


그림 11 질의 트랜잭션의 증가에 따른 질의 응답 시간

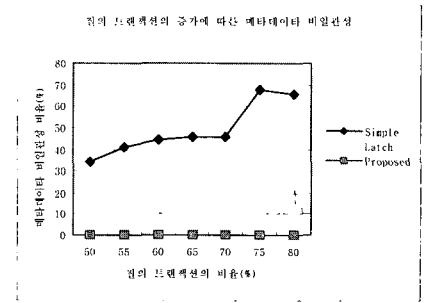


그림 12 질의 트랜잭션의 증가에 따른 메타데이터 비 일관성 비율

그림 10은 질의 트랜잭션의 증가에 따른 평균 대기 시간을 나타낸다. 질의 트랜잭션의 비율이 커짐에 따라 2PL의 평균 대기 시간은 현저하게 감소한다. 이는 질의 트랜잭션의 비율이 커지고 갱신 트랜잭션의 비율이 적어져서 데이터 충돌 횟수가 줄기 때문이다. 그러나 로크를 푸는 과정과 블락큐에서 대기하는 시간, 데드록 체크로 인해 단순 래치 기법이나 제안하는 알고리즘보다 오

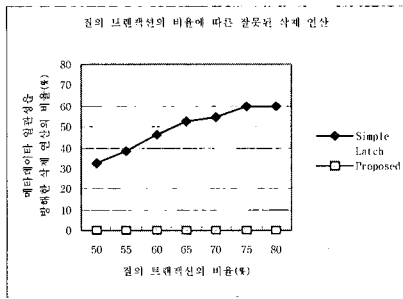


그림 13 질의 트랜잭션의 증가에 따른 잘못된 삭제 연산 비율

랜 시간이 걸린다. 단순 래치와 제안하는 알고리즘은 제거-리스트 체크와 메시지 발생 시간으로 인해 생기는 차이를 고려해도 크게 다른 점이 없다.

그림 11은 질의 트랜잭션의 증가에 따른 질의 응답 시간이다. 질의 트랜잭션의 비율이 커져도 2PL의 질의 응답 시간은 평균 대기 시간처럼 현저하게 줄지 않는다. 이는 아무리 데이터 충돌 횟수가 줄어도 기본적으로 2PL은 장기 로크를 사용하기 때문이다. 제안하는 기법은 평균 대기 시간과 마찬가지로 단순 래치를 이용하는 방법보다 크게 증가하지 않는 결과를 낸다. 두 성능평가의 결과를 살펴보면 질의 트랜잭션의 비율에 많은 관계를 나타내지 않는다. 이는 로크를 걸자마자 푸는 래치를 사용함으로써 데이터 충돌의 횟수에 민감하게 반응하지 않기 때문이다. 즉 제안하는 관리기법은 메타데이터 일관성을 지키면서 전통적인 트랜잭션 관리 기법에 비해 높은 성능을 나타내며, 메타데이터 일관성을 고려하지 않은 단순 래치 기법과 비슷한 성능을 낼 수 있으므로 성능은 높이고 일관성은 지킬 수 있음을 할 수 있다.

그림 12는 질의 트랜잭션의 증가에 따른 메타데이터 비일관성 비율을 나타낸다. 이것은 제안하는 기법과 단순 래치를 비교하기 위한 성능평가이다. 제안하는 기법은 메타데이터 일관성에 기초하여 만들어졌지만, 단순 래치 기법이 얼마나 메타데이터 일관성을 만족하지 못하는지 나타내는 그래프이다. 그림 12를 보면 단순히 래치만을 사용하면 질의 트랜잭션의 비율이 증가함에 따라 판독 이후 삭제된 문서를 결과에 반영한 질의 트랜잭션의 비율도 증가한다. 그림 13은 질의 트랜잭션의 증가에 따른 잘못된 삭제 연산 비율을 나타낸다. 단순히 래치만을 사용하면 질의 트랜잭션의 비율이 증가함에 따라 삭제 연산 중 메타데이터 일관성을 해치는 삭제 연산의 비율도 증가한다. 두 성능평가의 결과는 성능면

에서 비슷한 두 관리 기법이 일관성면에서 어느정도 차이를 나타내는지 나타낸다. 단순 래치 기법을 사용하면 질의 트랜잭션의 비율이 높아질수록 소스 데이터와 메타데이터 사이의 비일관성은 높게 증가하게 됨을 알 수 있고 메타데이터 일관성을 방해하게 된 삭제 연산의 비율또한 증가함을 알 수 있다. 디지털 도서관은 갱신 트랜잭션에 비해 질의 트랜잭션의 비율이 매우 높고 질의 트랜잭션의 결과가 정확함을 보장해야 하는데 단순 래치 기법만을 사용하면 질의 트랜잭션의 결과가 정확하지 않음을 성능평가를 통해 알 수 있다. 제안하는 관리 기법은 소스 데이터와 메타데이터 사이의 일관성을 보장하고 결과의 최근성을 보장하면서도 전통적인 트랜잭션 기법보다 높은 성능을 나타냄을 알 수 있다.

5. 결론

효과적인 디지털 도서관은 정보검색에서 좋은 성능을 보여야 하고 안정된 형태의 데이터가 제공되어야 한다. 이에 Dublin Core를 모델링하고, 메타데이터베이스 관리를 위해 기존의 데이터베이스 시스템에서 사용하던 트랜잭션 관리를 메타데이터의 특징에 맞게 변형하였다.

메타데이터 관리시 고려해야 할 사항은 일관성과 최근성 두 가지이다. 일관성 부분에서는 기존 데이터베이스 시스템에서 연구되어 오던 일관성 기준과 구분하여, 정보검색 시스템의 질의 결과가 실제 소스데이터와 가지는 일관성을 메타데이터 일관성이라 정의하였다. 메타데이터 관리 기법에서는 이러한 메타데이터 일관성을 지키면서 최근성을 잃지 않을 수 있는 기법을 고려하였다. 즉, 메타데이터베이스에 대한 질의 트랜잭션의 결과가 되는 질의 트랜잭션의 뷰시점을 세 가지로 분류하여 이 중 판독 시점 뷰를 제시하고 메타데이터 일관성을 지키기 위해 갱신 트랜잭션의 연산을 삽입과 삭제로 구분하여 이 중 삭제연산의 메타데이터베이스와 소스데이터베이스의 수행 시점을 달리하였다. 특히 메타데이터 일관성을 지키고 최근성을 보장하는 관리 기법은 매우 단순하여 응용 분야에서 구현되기 용이하고 높은 성능을 낼 수 있다.

일관성의 기준을 전통적인 일관성 기준과 달리 정보검색 시스템의 요구사항에 맞도록 정의한 연구는 앞으로 디지털 도서관을 설계하는데 있어 하나의 기준이 될 수 있는 사항이다. 디지털 도서관과 같이 정보검색 시스템, 데이터베이스 시스템, 분산 시스템 등의 통합 시스템의 성격을 갖는 응용 분야는 현재 활발히 연구가 진행 중이다. 본 논문은 이러한 응용 분야에서 메타데이터

와 관련된 모델링, 요구사항 도출, 일관성 기준을 제시하였다.

최근 메타데이터에 대한 연구는 전자상거래 부분에 응용되어 문서에 대한 식별자 연구와 함께 진행, 변화되고 있다. 이러한 예는 DOI(Digital Object Identifier), UKOLN(the UK Office for Library and information Networking)의 Squid 시스템, INDECS(Interoperability of Data in E-Commerce Systems) 프로젝트 등에서 볼 수 있으며 메타데이터 관리 기법은 문서에 대한 식별자 부여 시스템과 같이 개발되고 있으므로 이러한 디지털 도서관 응용에의 연구도 향후 진행되어야 할 것이다.

참 고 문 헌

- [1] Mohan Kamath and Krithi Ramamritham, "Efficient Transaction Support for Dynamic Information Retrieval Systems," *19th International ACM SIGIR International Conference on Research and Development in Information Retrieval(SIGIR '96)*, 1996, <http://www-ccs.cs.umass.edu/~kamath/publications.html>
- [2] Mohan Kamath and Krithi Ramamritham, *Efficient Transaction Management & Query Processing in Massive Digital Databases*, Technical Report 95-93 Dept. of C.S., University of Massachusetts, 1995, <http://www-ccs.cs.umass.edu/~kamath/publications.html>
- [3] Karen Kaye, "Metadata Formats and Standards : A World Wide Web Perspective," *Workshop on Improving International Exchange of Aerospace Information*, 1996, <http://www.nlr.nl/public/library/lib-work/paper-3/metadata.html>
- [4] Stuart Weibel and Eric Miller, "Dublin Core Metadata Element Set : Reference Description," *4th Dublin Core Metadata Workshop*, 1997, http://purl.oclc.org/metadata/dublin_core_elements
- [5] Traugott Koch, Anna Bramer, Devra Hiom, Marianne Peereboom, Alan Poulter, and Emma Worsfold, "The Role of Classification Schemes in Internet Resource Description and Discovery," *Development of a European Service for Information on Research and Education(DESIRE) project*, 1997, <http://www.ukoln.ac.uk/metadata/DESIRE/classification>
- [6] Carl Lagoze, "The Warwick Framework : A Container Architecture for Diverse Sets of Metadata," *D-Lib Magazine July/August*, 1996, <http://www.dlib.org/dlib/july96/lagoze/07lagoze.html>
- [7] Lorcan Dempsey, "The Warwick Framework : A Framework for the Deployment of Resource Description," *D-Lib Magazine July/August*, 1996, <http://www.dlib.org/dlib/july96/07weibel.html>
- [8] Michail Salamapasis, John Tait and Colin Hardy, "An Agent-Based Hypermedia Framework for Designing and Developing Digital Libraries," *Forum on Research and Technology Advances in Digital Libraries*, pp.5-13, 1996
- [9] 이성용, 이해민, 박 석, "디지털 도서관을 위한 에이전트 기반의 프레임워크 모델", 한국정보과학회 학술발표 논문집 제 24권 2호, pp.365-368, 1997
- [10] Jim Gray, Raymond A. Lorie, Gianfranco R. Putzolu, Irving L. Traiger, "Granularity of Locks in a Large Shared Data Base," *Very Large Data Base(VLDB)*, pp.428-451, 1975
- [11] Alan B. Pritsker, *Introduction to Simulation and SLAM II*, System Publishing Corp., pp.380-482, 1986
- [12] Vijay Kumar, *Performance of Concurrency Control Mechanism in Centralized Database Systems*, Prentice Hall Inc., pp.429-493, 1996



이 해 민

1996년 서강대학교 컴퓨터학과 학사.
1998년 서강대학교 컴퓨터학과 공학 석사.
1998년 ~ 현재 한국교육학술정보원 학술연구정보화실 연구원 재직중. 관심분야는 트랜잭션 관리, 메타데이터, 디지털 도서관.



박 석

1978년 서울대학교 계산통계학과(이학사). 1980년 한국과학기술원 전산학과(공학석사). 1983년 한국과학기술원 전산학과(공학박사). 1983년 9월 ~ 현재 서강대학교 전자계산학과 교수. 1989년 ~ 1991년 University of Virginia 방문교수. 1996년 ~ 현재 한국정보과학회 데이터베이스 연구회운영위원장. 관심분야는 실시간 데이터베이스 연구회운영위원장. 관심분야는 실시간 데이터베이스, 보완 데이터베이스, 멀티미디어 데이터베이스, 트랜잭션 관리, 데이터웨어하우스, 디지털 도서관.