히포크라테스 XML 데이타베이스: 모델 및 액세스 통제 방법

(Hippocratic XML Databases: A Model and Access Control Mechanism)

이 재 길 † 한 욱 신 †† 황 규 영 †††
(Jae-Gil Lee) (Wook-Shin Han) (Kyu-Young Whang)

요 약 최근에 Agrawal 등이 제안한 히포크라테스 데이타베이스(Hippocratic database)는 관계형 데이타베이스에 프라이버시 보호 기능을 추가한 데이타베이스 모델이다. 히포크라테스 데이타베이스는 관계형 데이타베이스에 기반한 모델이므로 최근에 널리 사용되는 XML 데이타베이스에 적용하기 위해서는 확장이 필요하다. 본 논문에서는 히포크라테스 데이타베이스 모델을 XML 데이타베이스에 적용할 수 있도록 확장한 **히포크라테스 XML 데이타베이스**(*Hippocratic XML database*) 모델과 이 모델에서의 효과적인 액세스 통제 방법을 제안한다. XML 데이타는 관계형 모델과 달리 트리 형태의 계층 구조를 가진다. 따라서, 히포크라테스 데이타베이스의 모델에서 제시한 개념들인 프라이버시 선호 및 정책, 프라이버시 권한, 데이타 레코드의 사용목적을 트리 형태의 계층 구조에 맞게 확장하며, 확장된 개념들을 정형적으로 정의한다. 다음으로, 본 모델의 액세스 통제 방법에 사용되는 새로운 방법인 다차원 인덱스를 사용한 **권한 인덱스**(*authorization index*)를 제안한다. 이 권한 인덱스는 최근접 질의(nearest neighbor search) 기법을 활용하여 가장 가까운 조상 엘리먼트에 부여된 권한에 의해 내포되는 권한을 효율적으로 찾을 수 있게 해준다. 합성 데이타와 실제 데이타를 사용하여 기존의 액세스 통제 방법과 질의 처리 시간을 비교하는 다양한 실험을 수행한 결과, 본 논문에서 제안한 액세스 통제 방법은 하향식(top-down) 액세스 통제 방법에 비하여 최대 13.6배, 상향식(bottom-up) 액세스 통제 방법에 비하여 최대 20.3배 성능을 향상시킴을 보였다. 본 논문의 주요 공헌은 1) 히포크라테스 데이타베이스 모델을 히포크라테스 XML 데이타베이스 모델로 확장하고 2) 제안한 모델 상에서 권한 인덱스와 최근접 질의 기법을 사용하는 효과적인 액세스 통제 방법을 제안한 것이다.

키워드 : 보안, 프라이버시, XML 데이타베이스, 히포크라테스 데이타베이스

Abstract The Hippocratic database model recently proposed by Agrawal et al. incorporates privacy protection capabilities into relational databases. Since the Hippocratic database is based on the relational database, it needs extensions to be adapted for XML databases. In this paper, we propose the *Hippocratic XML database* model, an extension of the Hippocratic database model for XML databases and present an efficient access control mechanism under this model. In contrast to relational data, XML data have tree-like hierarchies. Thus, in order to manage these hierarchies of XML data, we extend and formally define such concepts presented in the Hippocratic database model as privacy preferences, privacy policies, privacy authorizations, and usage purposes of data records. Next, we present a new mechanism, which we call the *authorization index*, that is used in the access control mechanism. This authorization index, which is implemented using a multi-dimensional index, allows us to efficiently search authorizations implied by the authorization granted on the nearest ancestor using the nearest neighbor search technique. Using synthetic and real data, we have performed extensive experiments comparing query processing time with those of existing access control

· 본 연구는 첨단정보기술연구센터를 통하여 한국과학재단으로부터 지원을 받았음
† 학생회원 : 한국과학기술원 전산학과/첨단정보기술연구센터
jglee@mozart.kaist.ac.kr
†† 종신회원 : 경북대학교 컴퓨터공학과
wshan@knu.ac.kr
††† 종신회원 : 한국과학기술원 전산학과 교수
첨단정보기술연구센터 소장
kywhang@mozart.kaist.ac.kr
논문접수 : 2003년 12월 23일
심사완료 : 2004년 8월 3일

mechanisms. The results show that the proposed access control mechanism improves the wall clock time by up to 13.6 times over the top-down access control strategy and by up to 20.3 times over the bottom-up access control strategy. The major contributions of our paper are 1) extending the Hippocratic database model into the Hippocratic XML database model and 2) proposing an efficient access control mechanism that uses the authorization index and nearest neighbor search technique under this model.

Key words : Security, Privacy, XML database, Hippocratic database

1. 서 론

최근 들어 사적인 데이타가 데이타베이스에 점점 더 많이 저장되고 있으며[1], 프라이버시(privacy)의 중요성이 크게 대두되고 있다[2]. 데이타베이스에 저장되어 있는 사적인 데이타를 제공한 데이타 제공자의 프라이버시를 보호하기 위해, Agrawal 등은 프라이버시 보호 기능이 데이타베이스 시스템의 주요 기능에 추가되어야 한다는 점을 강조하고, 관계형 데이타베이스에 프라이버시 보호 기능을 추가한 히포크라테스 데이타베이스(Hippocratic database)라는 새로운 개념을 제안하였다[3].

프라이버시 보호를 위해, 히포크라테스 데이타베이스는 프라이버시 메타데이타(privacy metadata)를 저장한다. **프라이버시 메타데이타**는 프라이버시 정책(privacy policy)과 프라이버시 권한(privacy authorization)으로 구성된다. **프라이버시 정책**은 데이타 수집 및 사용을 위한 데이타 관리자의 정책을 나타내는 정보로서, 데이타 테이블의 각 속성 별로 사용목적, 외부 수령자, 보유 기간을 포함한다. **프라이버시 권한**은 프라이버시 정책에 따라서만 데이타가 액세스되도록 통제하기 위해 데이타 관리자에 의해 데이타 사용자에게 부여되는 권한이다. 이 권한은 데이타 테이블의 각 속성 별로 사용목적과 허용된 사용자를 명시하며, 데이타 사용자가 프라이버시 정책에 명시된 사용목적에 따라서만 데이타 테이블의 속성을 액세스하도록 보장한다.

히포크라테스 데이타베이스는 데이타 수집 과정과 질의 처리 과정에서 프라이버시 보호를 위한 검사를 수행한다. 데이타 수집 과정에서 데이타 제공자는 우선 프라이버시 선호(privacy preference)를 명시한다. **프라이버시 선호**는 데이타 제공자의 의도를 나타내는 정보로서, 데이타 테이블의 각 속성 별로 사용목적, 외부 수령자, 보유 기간을 포함한다. 프라이버시 선호가 프라이버시 정책과 일치할 때만 히포크라테스 데이타베이스는 데이타 레코드를 프라이버시 선호에 포함된 사용목적과 함께 저장한다. 질의 처리 과정에서는 우선 스키마 레벨에서 프라이버시 권한에 명시된 사용목적을 검사한 후, 레코드 레벨에서 데이타와 함께 저장된 사용목적을 검사함으로써 데이타 사용자가 데이타 제공자의 의도에 따라서만 데이타를 액세스하도록 통제한다.

현재 XML 데이타의 프라이버시 보호와 관련하여 많은 연구가 진행 중이며, 대표적인 연구로는 Platform for Privacy Preference(P3P)[4] 표준과 XML 보안 모델[5,6,7]이 있다. P3P는 XML 데이타에 프라이버시 정책과 프라이버시 선호를 명시하고 비교하는 방법을 제공하지만, 이는 히포크라테스 데이타베이스의 데이타 수집 과정에 필요한 기능만을 제공한다. 한편, XML 보안 모델은 XML 데이타에 권한을 부여하고 권한을 통하여 데이타 액세스를 통제하는 방법을 제공하지만, 이는 히포크라테스 데이타베이스의 질의 처리 과정에 필요한 기능만을 제공한다. 그러므로, XML 데이타베이스에서는 히포크라테스 데이타베이스 모델과 같이 데이타 수집과 질의 처리 과정에서의 프라이버시 보호 기능을 제공하는 통합된 모델은 아직 제안된 바 없다.

본 논문에서는 **히포크라테스 XML 데이타베이스** (*Hippocratic XML database*) 모델을 정의하고 이 모델에서의 효과적인 액세스 통제 방법을 제안한다. 관계형 데이타는 평면(flat) 구조를 가지는 반면 XML 데이타는 트리 형태의 계층 구조를 가진다. 이러한 구조적인 차이점으로 인해 프라이버시 선호 및 정책, 프라이버시 권한, 데이타 레코드의 사용목적과 같이 히포크라테스 데이타베이스 모델에서 사용하는 프라이버시 정보들이 XML 데이타 트리의 임의의 엘리먼트에 명시될 수 있도록 확장되어야 한다. 본 모델에서 임의의 엘리먼트에 명시된 프라이버시 정보는 데이타 계층 구조의 후손 엘리먼트에 대한 프라이버시 정보를 묵시적으로 내포한다[4-6]. 이러한 내포 규칙에 따라 프라이버시 정보를 검사하는 방법도 검사하려는 엘리먼트 뿐만 아니라 그 조상 엘리먼트까지 검사하도록 확장되어야 한다.

히포크라테스 XML 데이타베이스에서는 효과적인 액세스 통제 방법이 필요하다. 왜냐하면, 프라이버시 권한은 인스턴스 레벨에서 부여될 수 있으며, 인스턴스 레벨의 권한 검사는 일반적으로 많은 처리 시간이 소요되므로 질의 처리 성능에 큰 영향을 미칠 수 있기 때문이다. 최근에 발표된 여러 가지 XML 액세스 통제 방법[5,6,8,9]들은 인스턴스 레벨의 권한을 지원한다. 그러나, 기존의 XML 액세스 통제 방법은 XML 데이타 트리를 탐색하면서 가장 가까운 조상 엘리먼트에 부여된 권한

을 검색하므로, 최악의 경우 XML 데이타 트리 전체를 액세스하는 비용이 소요된다. 이로 인해, 기존의 XML 권한 검사 방법은 시스템의 성능을 크게 저하시킬 수 있다.

이러한 문제점을 해결하기 위해, 본 논문에서는 권한 인덱스와 최근접 질의[10,11] 기법을 활용하여 부여된 권한을 효율적으로 검색하는 액세스 통제 방법을 제안한다. 권한 인덱스는 다차원 인덱스를 사용하여 구현된다. 제안하는 방법은 먼저 XML 데이타 트리에서 권한이 부여된 엘리먼트를 2-차원 공간상의 점으로 매핑시켜 다차원 인덱스에 저장한다. XML 데이타 트리의 특정 엘리먼트에 대한 권한은 가장 가까운 조상 엘리먼트에 부여된 권한에 의해 결정되므로 이 권한을 찾아내기 위해 최근접 질의 기법을 활용한다.

본 논문의 구성은 다음과 같다. 제2절에서는 프라이버시 보호와 관련된 기존의 연구를 설명한다. 제3절에서는 히포크라테스 XML 데이타베이스 모델을 정의한다. 제4절에서는 히포크라테스 XML 데이타베이스에서의 액세스 통제 방법을 제안한다. 제5절에서는 제안하는 액세스 통제 방법의 성능 평가 결과를 제시한다. 마지막으로, 제6절에서는 결론을 내린다.

2 관련 연구

본 절에서는 프라이버시 보호를 위한 기존의 모델과 검사 방법에 대한 연구를 요약한다. 제2.1절에서는 Agrawal 등이 제안한 히포크라테스 데이타베이스[3]를 설명한다. 제2.2절에서는 W3C에서 제정한 표준인 Platform for Privacy Preference(P3P)[4]를 설명한다. 제2.3 절에서는 기존의 XML 보안 모델[5,6,7]을 설명한다.

2.1 히포크라테스 데이타베이스

히포크라테스 데이타베이스[3]는 관계형 데이타베이스에 프라이버시 보호 기능을 추가한 데이타베이스 모델이다. 그림 1은 히포크라테스 데이타베이스 모델의 프라이버시 보호를 위한 아키텍처 및 구성 요소를 나타낸다. 이미 서론에서 히포크라테스 데이타베이스의 기본적인 개념을 설명하였으므로, 본 절에서는 히포크라테스 데이타베이스의 스키마의 예와 본 논문에서 중점을 두는 히포크라테스 데이타베이스의 질의 처리 방법을 집중적으로 설명한다.

그림 2[1]는 히포크라테스 데이타베이스의 스키마의 예를 나타낸다. 프라이버시 정책은 데이타의 사용목적, 테이블 이름, 속성 이름, 외부 수령자, 보유 기간을 포함하며, 그림 2(a)와 같이 프라이버시-정책 테이블에 저장된다. 프라이버시 권한은 데이타의 사용목적, 테이블 이름, 속성 이름, 허가된 사용자를 포함하며, 그림 2(b)와 같이 프라이버시-권한 테이블에 저장된다. 데이타 레코드의 사용목적은 그림 2(c)와 같이 데이타 테이블의 *purpose* 속성에 저장되며, 이는 인스턴스 레벨의 프라이버시 보호를 위해 사용된다.

질의 처리 과정에서 히포크라테스 데이타베이스는 속성 액세스 통제(Attribute Access Control)와 레코드 액세스 통제(Record Access Control)를 연속해서 수행한다[3]. 질의에는 질의를 수행하고자 하는 사용목적이 함께 명시되며, 이는 속성 액세스 통제와 레코드 액세스 통제에 사용된다. **속성 액세스 통제**는 질의에 명시된 테이블의 속성을 질의 사용목적을 위해 액세스할 수 있도록 허용하는 프라이버시 권한이 부여되어 있는지를 검사하는 것으로서, 스키마 레벨의 프라이버시 보호 방법이라 볼 수 있다. 스키마 레벨의 프라이버시 권한 검사는 관계형 데이타베이스 시스템에서 질의 컴파일 시점에 수행하는 권한 검사와 유사하다.

레코드 액세스 통제는 액세스하려는 레코드에 저장된 *purpose* 속성의 값이 질의 사용목적과 일치하는지를 검사하는 것으로서, 인스턴스 레벨의 프라이버시 보호 방법이라 볼 수 있다. 이는 일반적인 관계형 데이타베이스 시스템에서 수행하지 않는 과정으로, 히포크라테스 데이타베이스에서만 추가적으로 수행하는 과정이다. 레

1) 설명의 용이함을 위해 참고 문헌 [3]과 같이 테이블에 집합(set) 타입을 사용한다.

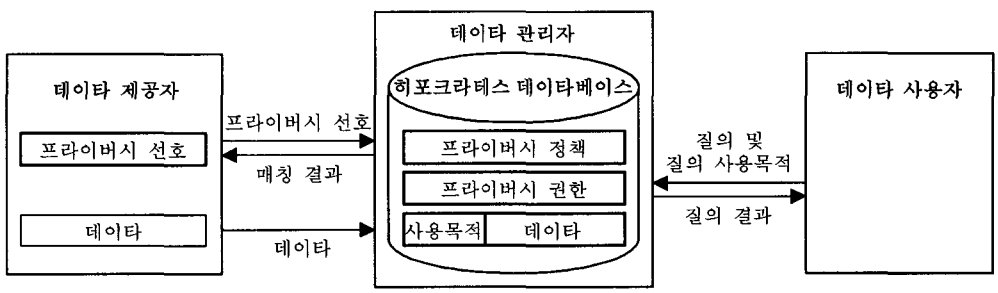


그림 1 히포크라테스 데이타베이스의 아키텍처 및 구성 요소

코드 액세스 통제는 질의에서 액세스하려는 모든 레코드들을 대상으로 수행하는 인스턴스 레벨의 검사이기 때문에 스키마 레벨의 권한 검사에 비해 시간이 많이 소요되며, 질의 처리 성능에 큰 영향을 미칠 수 있다.

예 1. 그림 2에서 *customer-service*라는 사용자가 *purchase* 사용목적을 위해 *Customer* 테이블에서 e-mail을 구하는 질의를 수행한다고 가정한다. 첫째, 속성 액세스 통제에서는 *customer-service* 사용자가 *purchase* 사용목적을 위해 *Customer* 테이블의 *email* 속성을 액세스할 수 있는지 검사한다. 이 경우, 프라이버시-권한 테이블의 첫번째 프라이버시 권한이 그 사용자에게 액세스를 허용하도록 부여되어 있다. 둘째, 레코드 액세스 통제에서는 액세스하는 레코드의 *purpose* 속성에 *purchase* 사용목적이 포함되어 있는지 검사한다. 이 경우, *Customer* 테이블의 첫번째와 두번째 레코드의 *purpose* 속성에는 *purchase* 사용목적이 포함되어 있으므로, 질의 결과로 Bob과 Alice의 e-mail이 반환된다. □

2.2 Platform for Privacy Preference(P3P)

Platform for Privacy Preference(P3P)[4]는 데이타 제공자가 자신이 방문한 웹 사이트에 개인 정보를 제공하는 것을 통제할 수 있도록 W3C에 의해 제정된 표준이다. 데이타 제공자가 P3P를 지원하는 웹 사이트에 접속할 때, P3P를 지원하는 웹 브라우저는 자동적으로 웹 사이트의 프라이버시 정책(privacy policy)을 읽어 제공자의 프라이버시 선호(privacy preference)와 비교한다. 프라이버시 정책과 프라이버시 선호가 일치하지 않으면 웹 브라우저는 경고 메시지를 표시하며, 데이타 제공자는 이를 보고 개인 정보를 웹 사이트에 제공할지를 결정할 수 있다.

P3P에서 프라이버시 정책과 프라이버시 선호는 임의의 DTD 엘리먼트에 명시될 수 있다. 특정 DTD 엘리먼트에 명시된 프라이버시 정책과 프라이버시 선호는 후손 엘리먼트에 대한 프라이버시 정책과 프라이버시 선호를 묵시적으로 내포한다. P3P는 히포크라테스 데이타베이스의 데이타 수집 과정에 필요한 기능만을 제공한다. 즉, P3P는 데이타 제공자가 자신의 개인 정보를 제공하기 전에 웹 사이트의 프라이버시 정책을 확인할 수 있는 방법을 제공한다. 그러나, P3P는 질의 처리 과정에서 데이타 제공자의 프라이버시를 보호하는 방법은 제공하지 않으며, 이는 제 2.3 절에서 설명하는 XML 보안 모델에서 제공한다.

2.3 XML 보안 모델

XML 데이타의 보안을 위하여 여러 가지 XML 보안 모델[5-7]이 발표되어 있다. XML 보안 모델은 데이타베이스에 저장된 XML 데이타에 권한을 부여 및 취소하는 방법과 부여된 권한을 통하여 XML 데이타에 대한 액세스를 통제하는 방법을 제공한다.

기존의 XML 보안 모델에서, 권한은 기본적으로 5-투플 (*s*, *o*, *a*, *sign*, *imply_option*)로 정의한다[5,6]. *s*는 권한이 부여된 사용자 혹은 사용자 그룹을, *o*는 권한에 의해 보호되는 XML 문서의 엘리먼트를, *a*는 수행이

purpose	table	attribute	external-recipients	retention
purchase	customer	email	*empty*	1 month
purchase	customer	credit-card-info	{ credit-card-company }	1 month
registration	customer	name	*empty*	3 years
...	...	...	...	...

(a) 프라이버시-정책 테이블

purpose	table	attribute	authorized-users
purchase	customer	email	{ shipping, customer-service }
purchase	customer	credit-card-info	{ charge }
registration	customer	name	{ registration }
...	...	...	...

(b) 프라이버시-권한 테이블

purpose	customer-id	name	email	...
{ purchase, registration }	0	Bob	bob@ibm.com	...
{ purchase }	1	Alice	alice@microsoft.com	...
{ registration }	2	Mallory	mallory@hotmail.com	...
{ registration, purchase-circles }	3	Trent	trent@oracle.com	...
...	...	...	...	...

(c) 데이타 테이블 : Customer 테이블

그림 2 히포크라테스 데이타베이스의 스키마의 예

허용 혹은 금지되는 연산의 종류를 나타낸다. *sign*은 + 과 - 중의 한 값을 가지며, 해당 연산의 허용 및 금지 여부를 나타낸다. *imply_option*은 권한이 후손 엘리먼트에 대한 권한을 묵시적으로 내포(imply)[2]하는지를 나타낸다.

권한은 스키마 레벨 혹은 인스턴스 레벨에서 부여될 수 있다. 스키마 레벨의 권한은 DTD에 부여되며 권한이 부여된 DTD를 따르는 모든 XML 문서에 적용된다. 인스턴스 레벨의 권한은 XML 문서에 부여되며 권한이 부여된 XML 문서로 국한된다.

XML 문서의 특정 엘리먼트에 대한 권한은 가장 가까운 조상 엘리먼트에 부여된 권한에 의해 내포될 수 있으므로, 권한 검사를 수행하기 위해서는 해당 엘리먼트 뿐만 아니라 조상 엘리먼트에도 권한이 부여되어 있는지를 검사해야 한다. 이를 위해, 기존의 XML 액세스 통제 방법[5,6]은 XML 문서의 루트와 검사하려는 엘리먼트 사이의 패스 상의 각각의 엘리먼트마다 권한이 부여되어 있는지 검사한다. 기존의 방법은 패스를 탐색하는 방향에 따라 루트로부터 검사하려는 엘리먼트의 방향으로 패스를 탐색하는 **하향식**(*top-down*) **방법**과 검사하려는 엘리먼트로부터 루트의 방향으로 패스를 탐색하는 **상향식**(*bottom-up*) **방법**으로 구분된다[5]. 두 가지 방법은 모두 권한이 부여되어 있지 않은 엘리먼트까지 불필요하게 액세스할 수 있다는 문제가 있다. 검사하려는 엘리먼트가 데이타베이스 전체에 분포되어 있으면, 최악의 경우 권한 검사를 위해 전체 데이타베이스를 액세스할 수 있다[5].

최근에 일반적인 XML 보안 모델을 다소 변형시킨 보안 모델에서의 액세스 통제 방법이 몇 가지 발표되었다. 참고 문헌 [8]에서는 권한이 후손 엘리먼트에 대한 권한을 내포하지 않는 보안 모델을 가정하였다. 이러한 종류의 보안 모델에서 모든 엘리먼트마다 권한이 명시적으로 부여되어야 하므로 명시적으로 부여된 권한의 개수가 많아지는 문제가 있다. 이와 같은 문제를 해결하기 위해 인접한 엘리먼트에 부여된 권한들을 압축하여 표현하는 방법을 제안하였다. 참고 문헌 [9]에서는 권한이 미리 지정된 일부 엘리먼트들에만 부여될 수 있는 특수한 보안 모델을 가정하고, 이러한 보안 모델에서 부여된 권한을 구할 때 상향식으로 탐색하는 패스의 길이를 최소화하는 방법을 제안하였다.

기존의 XML 보안 모델은 히포크라테스 데이타베이스 모델과 비교하여 다음과 같은 차이점이 있다. 첫째, 데이타 수집이나 데이타 제공자의 프라이버시 선호와 같은 개념이 존재하지 않는다. 둘째, 사용목적이라는 개념이 제공되지 않는다. 단, **읽기**(*read*) 혹은 **쓰기**(*write*) 등을 사용목적의 특수한 예로 간주할 수는 있다.

3. 히포크라테스 XML 데이타베이스 모델

본 절에서는 히포크라테스 XML 데이타베이스 모델을 제안한다. 히포크라테스 XML 데이타베이스 모델은 (1) 프라이버시 선호 및 프라이버시 정책 모델과 (2) 프라이버시 권한 모델로 구성된다. 첫번째 구성 요소는 데이타 수집 과정에서의 프라이버시 보호를 담당하며, 두 번째 구성 요소는 질의 처리 과정에서의 프라이버시 보호를 담당한다. 설명의 편의를 위해, 먼저 제 3.1 절에서 프라이버시 권한 모델을 정의하고, 제 3.2 절에서 프라이버시 선호 및 정책 모델을 정의한다. 이후부터는 구분을 위해 Agrawal 등[3]이 제안한 모델을 히포크라테스 관계형 데이타베이스라 부른다.

본 모델에는 데이타 계층 구조와 사용목적 계층 구조의 두 가지 계층 구조가 존재한다. 데이타 계층 구조는 XML 데이타의 트리 형태의 구조로 인하여 자연스럽게 발생한다. 또한, 사용목적은 일반적으로 계층적인 의미를 지닐 수 있으므로, 사용목적도 계층 구조를 가지도록 확장한다. 이와 같은 두 가지 계층 구조를 다룰 수 있도록, 프라이버시 선호, 프라이버시 정책, 프라이버시 권한, 데이타 레코드의 사용목적의 정의를 확장한다.

3.1 프라이버시 권한 모델

질의 처리 도중에 히포크라테스 XML 데이타베이스 모델은 Agrawal 등[3]에 의해 정의된 바와 같이 속성 액세스 통제와 레코드 액세스 통제를 수행함으로써 데이타 제공자의 프라이버시를 보호한다. 그러나, 두 가지 액세스 통제를 수행하는 방법에 있어서 본 모델은 히포크라테스 관계형 데이타베이스 모델과 차이가 있다. 히포크라테스 관계형 데이타베이스 모델에서는 속성 액세스 통제만이 권한 모델의 일부로서 정의되었고 레코드 액세스 통제는 그렇지 않다. 레코드 액세스 통제는 각 레코드의 *purpose* 속성 값을 검사하도록 정의되었다. 그러나, 본 모델에서는 데이타 레코드와 함께 저장된 사용목적을 권한으로 취급함으로써 속성 액세스 통제와 레코드 액세스 통제가 모두 권한 모델의 일부로서 정의된다. 이로서 두 가지 액세스 통제를 일관적인 형태로 정의하고 수행할 수 있다.

데이타 레코드와 함께 저장된 사용목적을 권한으로 취급함으로써, 레코드 액세스 통제를 수행함에 있어 사용목적 계층 구조뿐만 아니라 데이타 계층 구조를 쉽게 지원할 수 있다. 권한의 내포 규칙을 사용하여 특정 엘리먼트의 사용목적이 후손 엘리먼트의 사용목적을 내포하게 만들 수 있다. 또한, 권한의 내포 규칙을 사용하여

2) 대부분의 XML 보안 모델에서 전달(propagation)이라는 용어를 사용하지만 일관성을 위해 내포라는 용어로 통일한다.

특정 엘리먼트의 사용목적이 묵시적으로 하위 사용목적을 포함하도록 만들 수도 있다.

3.1.1 프라이버시 권한의 정의

정의 1과 2에서 두가지 종류의 권한인 **관리자 프라이버시 권한**(*administrator privacy authorization*)과 **제공자 프라이버시 권한**(*provider privacy authorization*)을 정의한다. 관리자 프라이버시 권한은 속성 액세스 통제에 사용되며, 제공자 프라이버시 권한은 레코드 액세스 통제에 사용된다.

정의 1. 데이타 관리자에 의해 부여되는 권한을 **관리자 프라이버시 권한**(*administrator privacy authorization*)이라 부르며, 3-투플 (s, o, p)로 정의한다.

- s: 데이타 사용자;
- o: DTD의 엘리먼트를 지정하는 경로식(path expression)[3];
- p: 권한 타입 혹은 데이타의 사용목적.

이 권한은 모델의 간결성을 위해 항상 후손 엘리먼트에 대한 권한을 내포한다고 가정한다.[4]

관리자 프라이버시 권한은 히포크라테스 관계형 데이타베이스 모델에서 정의된 프라이버시 권한을 XML 데이타에 적용할 수 있도록 확장한 것이다. 즉, 히포크라테스 관계형 데이타베이스 모델에서는 권한 객체가 데이타 테이블의 속성인데 반해, 본 모델에서는 권한 객체가 DTD의 임의의 엘리먼트가 될 수 있도록 경로식을 사용하여 정의를 확장한다. 관리자 프라이버시 권한은 DTD에 부여되는 프라이버시 정책에 포함된 사용목적을 반영하므로 기존의 XML 보안 모델에서의 스키마 레벨 권한[5,6]에 해당한다.

정의 2. 데이타 제공자에 의해 부여되는 권한을 **제공자 프라이버시 권한**(*provider privacy authorization*)이라 부르며, 2-투플 (o, p)로 정의한다.

- o: XML 문서의 엘리먼트를 지정하는 경로식(path expression);
- p: 권한 타입 혹은 데이타의 사용목적.

제공자 프라이버시 권한은 히포크라테스 관계형 데이타베이스 모델에서 데이타 레코드와 함께 저장되어 있는 사용목적을 나타내기 위한 것이다. 본 모델에서는 이러한 사용목적을 권한으로 간주하여 데이타 계층 구조와 사용목적 계층 구조에서의 권한 내포 규칙을 적용할 수 있도록 확장한다. 권한 객체는 경로식으로 지정되는 XML 문서의 임의의 엘리먼트이다. 제공자 프라이버시 권한은 제공된 데이타 인스턴스의 사용목적을 나타내기 위한 권한이므로 기존의 XML 보안 모델에서의 인스턴스 레벨 권한[5,6]에 해당한다.

(s, o, p)와 (o, p)의 권한 타입으로 사용되는 사용목적 p는 DAG 형태의 계층 구조를 가진다. 노드(node)는 사용목적을 나타내고, 노드 p_i부터 p_j까지의 아크(arc)는 사용목적 p_i가 사용목적 p_j를 내포함을 나타낸다. 이때, p_i를 p_j의 **상위 사용목적**이라고 부르고 p_j를 p_i의 **하위 사용목적**이라 부른다. 그림 3은 DAG 형태의 사용목적 계층 구조의 예로서, *analysis* 사용목적은 *individual analysis, global analysis* 사용목적 및 그 아래의 사용목적을 내포함을 나타낸다.

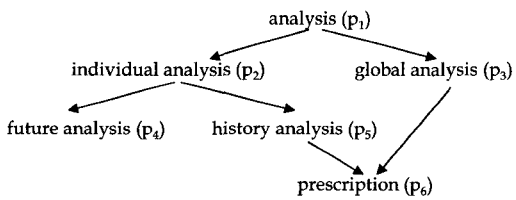


그림 3 사용목적 계층 구조의 예

관리자 프라이버시 권한과 제공자 프라이버시 권한은 명시적(explicit)/묵시적(implicit), 강성(strong)/약성(weak), 긍정적(positive)/부정적(negative) 프라이버시 권한으로 보다 세부적으로 구분된다. **명시적 프라이버시 권한**은 DTD의 특정 엘리먼트 혹은 XML 문서의 특정 엘리먼트에 대해 명시적으로 부여되는 권한이며, **묵시적 프라이버시 권한**은 가장 가까운 조상 엘리먼트에 부여된 명시적 권한에 의해 내포되는 후손 엘리먼트에 대한 권한이다. **강성 프라이버시 권한**은 묵시적 프라이버시 권한을 다른 명시적 프라이버시 권한이 오버라이드 하는 것을 허용하지 않는 권한이며, **약성 프라이버시 권한**은 묵시적 프라이버시 권한을 다른 명시적 프라이버시 권한이 오버라이드 하는 것을 허용하는 권한이다. 강성 프라이버시 권한은 (s, o, p)로 표시하고, 약성 프라이버시 권한은 $[s, o, p]$로 표시한다. **긍정적 프라이버시 권한**은 특정 사용목적을 위한 액세스를 허용하는 권한이고, **부정적 프라이버시 권한**은 특정 사용목적을 위한 액세스를 금지하는 권한이다. 긍정적 프라이버시 권한은 권한 타입을 p 혹은 $+p$로 표시하고, 부정적 프라이버시 권한은 권한 타입을 $\neg p$ 혹은 $-p$로 표시한다. 이러한 개념들은 계층 구조의 데이타를 가지는 객체지향 데이타베이스를

3) 본 논문에서는 경로식의 표기법으로 XPath[12] 표준을 사용한다. 경로식의 결과로 엘리먼트의 집합이 나올 수 있는데, 이러한 경우에는 각각의 엘리먼트마다 권한이 부여된다고 가정한다.

4) 만약 후손 엘리먼트에 대한 권한을 내포하는지 여부를 나타내기 위해서는, 권한의 정의에 옵션(imply option) 하나만을 추가함으로써 간단히 확장할 수 있다.

위한 보안 모델[13] 혹은 XML 데이타베이스를 위한 보안 모델[6]에서 일반적으로 지원하는 개념들이다.

데이타 사용자는 관리자 프라이버시 권한과 제공자 프라이버시 권한이 모두 부여된 엘리먼트만을 액세스할 수 있다.

예 2. 그림 4는 관리자 프라이버시 권한과 제공자 프라이버시 권한이 부여된 예를 나타낸다. 그림 4(a)의 *hospital.dtd*는 그림 4(b)의 *hospital.xml*의 DTD 이다. 그림 4 (a)에서 관리자 프라이버시 권한($user_A$, *document*("*hospital.dtd*")/*hospital*, *analysis*)은 *hospital.dtd*에 부여되므로, $user_A$는 이 DTD의 인스턴스인 *hospital.xml*의 *hospital* 엘리먼트 및 후손 엘리먼트들을 *analysis* 사용목적을 위해 액세스할 수 있다. 그림 4(b)에서 *hospital.xml*에 부여된 제공자 프라이버시 권한 (*document*("*hospital.xml*")/*hospital*/*patient*[0], *analysis*)은 *hospital.xml*의 첫번째 *patient* 엘리먼트 및 후손 엘리먼트들이 *analysis* 사용목적을 위해 데이타 제공자로부터 제공되었음을 나타낸다. 그러므로, 데이타 사용자 $user_A$는 관리자 프라이버시 권한과 제공자 프라이버시 권한이 모두 부여된 *hospital.xml*의 첫번째 *patient* 엘리먼트 및 후손 엘리먼트들을 *analysis* 사용목적을 위해 액세스할 수 있다. □

3.1.2 프라이버시 권한의 내포(implication) 규칙

관리자 프라이버시 권한과 제공자 프라이버시 권한은 히포크라테스 XML 데이타베이스 모델에 존재하는 두 가지 계층 구조인 데이타 계층 구조와 사용목적 계층 구조에서의 내포 규칙을 가진다. 관리자 프라이버시 권한과 제공자 프라이버시 권한은 동일한 내포 규칙을 가지기 때문에, 본 절에서는 관리자 프라이버시 권한으로 통칭하여 내포 규칙을 정의한다.

규칙 1: 데이타 계층 구조에서의 프라이버시 권한 내포 규칙

(1.a) 임의의 s, p에 대하여 o_i가 o_j의 조상 엘리먼트이면, (s, o_i, p)는 (s, o_j, p)를 내포한다.

(1.b) 임의의 s, p에 대하여 o_i가 o_j의 조상 엘리먼트이면, $(s, o_i, \neg p)$는 $(s, o_j, \neg p)$를 내포한다.

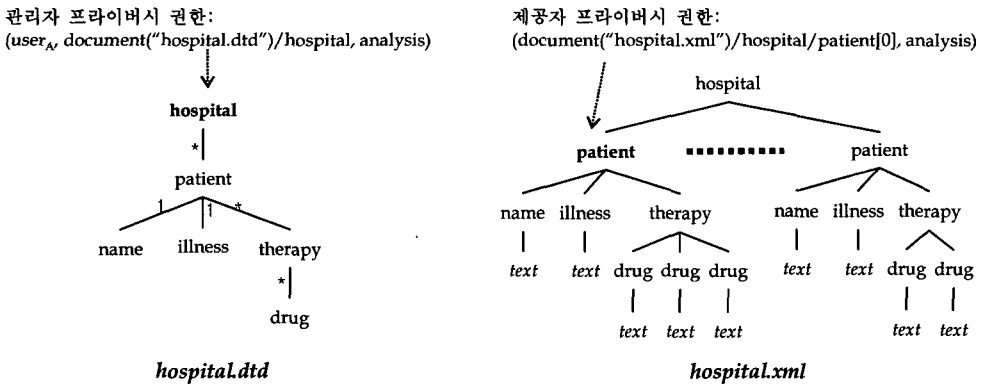
규칙 1에 의해, 데이타 계층 구조에서 특정 엘리먼트에 부여된 명시적 권한은 후손 엘리먼트에 대해 동일한 권한을 내포한다. 기존의 XML 보안 모델[5,6]도 이러한 내포 규칙과 동일한 권한 내포 규칙을 제공한다. 즉, 기존의 XML 보안 모델은 스키마 레벨의 권한과 인스턴스 레벨의 권한에 규칙 1과 동일한 권한 내포 규칙을 적용한다.

규칙 2: 사용목적 계층 구조에서의 프라이버시 권한 내포 규칙

(2.a) 임의의 s, o에 대하여 p_i가 p_j의 상위 사용목적이면, (s, o, p_i)는 (s, o, p_j)를 내포한다.

(2.b) 임의의 s, o에 대하여 p_i가 p_j의 상위 사용목적이면, $(s, o, \neg p_j)$는 $(s, o, \neg p_i)$를 내포한다.

규칙 2에 의해, 사용목적 계층 구조에서 긍정적 프라이버시 권한은 그 하위 사용목적을 위한 긍정적 프라이버시 권한을 내포하며, 부정적 프라이버시 권한은 그 상위 사용목적을 위한 부정적 프라이버시 권한을 내포한다. 규칙 (2.b)를 정형적으로 설명하면 다음과 같다. 사용목적 p_i가 p_j의 상위 사용목적이라고 가정하면, 규칙 (2.a)에 의해 권한 (s, o, p_i)는 사용목적 계층 구조에서 p_i의 모든 하위 사용목적을 위한 액세스를 허용한다. 이를 $(s, o, p_i) \equiv (s, o, p_j) \wedge (s, o, p_i')$로 표시할 수 있



(a) 관리자 프라이버시 권한의 부여 (b) 제공자 프라이버시 권한의 부여

그림 4 프라이버시 권한 부여의 예

으며, p_i'는 p_i에 의해 내포되는 사용목적 중에서 p_j를 제외한 사용목적들이다. 여기에서 (s, o, p_i)는 s가 o를 사용목적 p_i를 위해 액세스할 수 있음을 나타내는 식이다. $(s, o, \neg p_j) \equiv \neg(s, o, p_j) \Rightarrow \neg(s, o, p_i) \equiv (s, o, \neg p_i)$이므로, 규칙 (2.b)가 성립한다. 규칙 2는 객체지향 데이타베이스 보안 모델[13]의 권한 타입 계층 구조에서의 내포 규칙을 위해 제안되었으며, 본 논문에서는 이를 사용목적 계층 구조에 적용하였다.

3.1.3 프라이버시 권한의 충돌(conflict) 해결 정책

명시적 프라이버시 권한이 부여되어 있는 엘리먼트에 다른 명시적 프라이버시 권한을 부여하려 할 경우, 두 명시적 프라이버시 권한 간에 충돌이 발생할 수 있다. 충돌이 발생하면 추가적인 명시적 프라이버시 권한의 부여는 거절되고, 그렇지 않으면 추가적인 명시적 프라이버시 권한의 부여가 허용된다. 사용목적에 따른 명시적 프라이버시 권한 간의 충돌은 표 1과 같이 정의한다. 표 1에서 **T**는 충돌이 발생함을 의미하고 **F**는 충돌이 발생하지 않음을 의미한다. 표 1에서 사용목적 p_i는 사용목적 p_j의 상위 사용목적이라 가정한다.

표 1에서 두 명시적 프라이버시 권한의 관계가 **강화적** 혹은 **상존적** 일때만 충돌이 발생하지 않는다. **강화적**은 추가 프라이버시 권한의 사용목적이 기존 프라이버시 권한의 사용목적을 강화한다는 의미이다. **피포함적**은 강화적과 반대의 의미로 추가 프라이버시 권한의 사용목적이 기존 프라이버시 권한의 사용목적에 포함된다는 의미이다. **동일적**은 추가 프라이버시 권한의 사용목적이 기존 프라이버시 권한의 사용목적과 동일하다는 의미이다. **모순적**은 추가 프라이버시 권한의 사용목적이 기존 프라이버시 권한의 사용목적과 모순된다는 의미이다. **상존적**은 두 프라이버시 권한의 사용목적이 서로 강화적, 피포함적, 동일적, 모순적 의미 없이 상존할 수 있다는 의미이다.

한편, 묵시적 약성 프라이버시 권한이 부여되어 있는 엘리먼트에 다른 명시적 프라이버시 권한을 부여하면, 추가된 명시적 프라이버시 권한이 기존의 묵시적 프라이버시 권한을 오버라이드 한다. 즉, 하위 엘리먼트에 부여된 명시적 프라이버시 권한이 상위 엘리먼트에 부여된 명시적 프라이버시 권한을 오버라이드 하며, 이를 **최고 구체적 오버라이드**(*most specific overrides*)[5,6] 정책이라 부른다. 기존의 XML 보안 모델에서도 이와 동일한 정책을 채택하고 있다.

3.2 프라이버시 선호 및 프라이버시 정책 모델

3.2.1 프라이버시 선호 및 프라이버시 정책의 정의

히포크라테스 XML 데이타베이스 모델에서 프라이버시 정책은 데이타의 수집 및 사용을 위한 데이타 관리자의 정책을 나타내며, 프라이버시 선호는 데이타 제공자의 의도를 나타낸다. 본 모델의 프라이버시 선호 및 정책은 히포크라테스 관계형 데이타베이스 모델의 프라이버시 선호 및 정책을 DTD의 임의의 엘리먼트에 명시할 수 있도록 확장한 것이다. 프라이버시 선호와 프라이버시 정책을 명시하는데에는 공통의 DTD가 사용된다.

정의 3. 프라이버시 선호(*privacy preference*)와 **프라이버시 정책**(*privacy policy*)은 4-투플 (o, er, r, p)로 정의한다.

- o: DTD의 엘리먼트를 지정하는 경로식(path expression);
- er: 데이타를 받을 수 있는 외부 수령자들의 집합;
- r: 데이타가 저장되는 기간을 나타내는 보유 기간;
- p: 데이타의 사용목적.

프라이버시 선호 및 정책은 프라이버시 권한에서와 같이 명시적(explicit)/묵시적(implicit), 강성(strong)/약성(weak), 긍정적(positive)/부정적(negative) 프라이버시 선호 및 정책으로 보다 세부적으로 구분된다. 이러한 분류의 의미는 프라이버시 권한에서의 의미와 동일하다. 또한, 프라이버시 선호와 프라이버시 정책에도 프라이버시 권한의 내포 규칙과 동일한 내포 규칙이 적용된다.

3.2.2 프라이버시 선호와 프라이버시 정책의 매치

데이타 제공자로부터 데이타를 수집하기 위해서는 수집하려는 데이타에 명시된 프라이버시 선호 별로 매치하는 프라이버시 정책이 존재해야 한다. 프라이버시 선호와 프라이버시 정책의 **매치**는 정의 4에서 정의한다.

정의 4. 프라이버시 선호가 (o_i, er_i, r_i, p_i)이고 프라이버시 정책이 (o_j, er_j, r_j, p_j)일 때, 다음의 네 가지 조건을 모두 만족하면 프라이버시 선호와 프라이버시 정책이 매치한다고 정의한다: (1) o_i와 o_j가 동일하거나 혹

표 1 프라이버시 권한 충돌 행렬

기존 \ 추가	p_i	p_j	$\neg p_i$	$\neg p_j$
p_i	T (동일적)	T (피포함적)	T (모순적)	T (모순적)
p_j	F (강화적)	T (동일적)	F (상존적)	T (모순적)
$\neg p_i$	T (모순적)	F (상존적)	T (동일적)	F (강화적)
$\neg p_j$	T (모순적)	T (모순적)	T (피포함적)	T (동일적)

은 o_j가 o_i의 조상 엘리먼트이고, (2) er_i가 er_j를 포함하고 ($er_i \supseteq er_j$), (3) r_i가 r_j보다 긴 기간이고 ($r_i \geq r_j$), (4) p_i와 p_j가 동일하거나 p_i가 p_j의 상위 사용목적이다.

정의 4는 프라이버시 선호와 프라이버시 정책이 서로 매치하기 위해서는 o, er, r, p의 값이 각각 매치해야 한다는 것을 의미한다. 이때 o의 경우, 프라이버시 선호가 프라이버시 정책보다 더 제한적이어야 한다. 그렇지 않다면 프라이버시 정책이 명시되지 않은 데이타가 수집될 수 있다. 반면 er, r, p의 경우, 프라이버시 정책이 프라이버시 선호보다 더 제한적이어야 한다. 그렇지 않다면 데이타 제공자가 희망하지 않은 방식으로 데이타가 사용될 수 있기 때문이다.

4. 히포크라테스 XML 데이타베이스에서의 액세스 통제 방법

본 절에서는 히포크라테스 XML 데이타베이스 모델의 액세스 통제 방법을 설명한다. 제 4.1 절에서는 프라이버시 보호 방법의 개요를 설명한다. 제 4.2 절에서는 질의 처리 과정에서의 프라이버시 보호에 사용되는 권한 인덱스의 구조를 제안한다. 제 4.3 절에서는 권한 인덱스와 최근접 질의를 활용하는 효율적인 액세스 통제 방법을 제안한다. 이후부터는 관리자 프라이버시 권한과 제공자 프라이버시 권한을 구분해야 할 경우를 제외하고는 설명의 간결성을 위하여 간단히 **권한**이라고 통칭해서 부른다.

4.1 개요

그림 5는 프라이버시 보호를 위한 히포크라테스 XML 데이타베이스의 아키텍쳐를 나타낸다. 히포크라테스 XML 데이타베이스는 각각 프라이버시 정책 검사기(Privacy Policy Checker)와 권한 검사기(Authorization Checker)에 의해 데이타 수집과 질의 처리 시에 프라이버시 보호를 위한 검사를 수행한다. 본 모델에서 XML 질의는 XPath[12] 표준을 사용하여 명시한다.

프라이버시 정책 검사기는 데이타 수집 시에 데이타

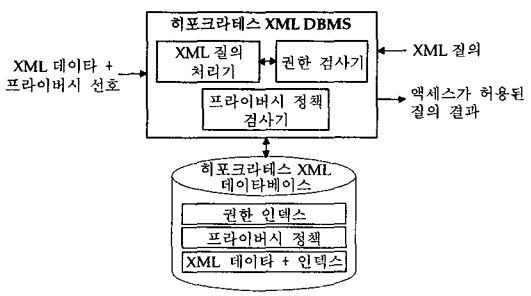


그림 5 프라이버시 보호를 위한 히포크라테스 XML 데이타베이스의 아키텍쳐

제공자의 프라이버시를 보호하는 모듈이다. 프라이버시 정책 검사기는 수집하려는 XML 데이타에 명시된 각각의 프라이버시 선호 별로 매치하는 프라이버시 정책이 존재하는지 검사한다. 모든 프라이버시 선호가 매치해야만 데이타를 수집하여 저장한다. 프라이버시 선호 및 정책은 DTD에 부여되므로, 스키마 레벨에서 검사가 수행된다. 따라서, 프라이버시 정책은 관계형 데이타베이스에서의 권한과 유사하게 카탈로그에 저장하고 검사한다.

권한 검사기는 질의 처리 시에 데이타 제공자의 프라이버시를 보호하기 위한 모듈이다. 권한 검사기는 XML 엘리먼트가 관리자 프라이버시 권한과 제공자 프라이버시 권한에 의해 모두 액세스가 허용되는지 검사한다. 관리자 프라이버시 권한은 스키마 레벨에서 부여되므로, 관계형 데이타베이스에서의 권한과 유사하게 카탈로그에 저장하고 질의 컴파일 시에 검사한다. 제공자 프라이버시 권한은 인스턴스 레벨에서 부여되므로, 질의 처리 도중에 검사한다. 인스턴스 레벨의 권한 검사는 스키마 레벨의 권한 검사에 비해 시간이 많이 소요되므로 질의 처리 성능에 큰 영향을 미칠 수 있다. 본 논문에서는 인스턴스 레벨의 권한 검사를 효과적으로 수행할 수 있도록, 인스턴스 레벨에 부여된 권한을 저장하는 **권한 인덱스**(*authorization index*)와 이를 사용하는 액세스 통제 방법을 제안한다.

4.2 권한 인덱스의 구조

권한이 부여된 엘리먼트는 많은 XML 질의 처리 방법에서 사용되는 번호화 방식(numbering scheme) [14-16]에 의해 2-차원 상의 점 (*start, end*)로 표현되므로, 이들을 저장하기 위한 권한 인덱스로서 2-차원 인덱스[17]를 사용한다. 번호화 방식에서 (*start, end*) 값은 엘리먼트 간의 조상-후손 관계를 나타내며, 다음의 두 조건을 만족한다[14]: (1) 엘리먼트 v가 엘리먼트 u의 부모일때 ($start_u$, end_u) 범위(interval)는 ($start_v$, end_v) 범위에 포함되며, (2) 두 형제(sibling) 엘리먼트 u, v 에서 엘리먼트 u가 엘리먼트 v 전에 나오면 $end_u < start_v$ 이다. 따라서, $start_a < start_d \wedge end_a > end_d$ 이면 엘리먼트 a는 d의 조상이다.

예 3. 그림 6과 같이 XML 문서에 제공자 프라이버시 권한이 부여되어 있다고 가정한다. 여기에서 권한이 부여된 엘리먼트는 굵은 글씨체로 표시되어 있다. 이때, 부여된 제공자 프라이버시 권한을 권한이 부여된 엘리먼트의 (*start, end*) 값으로 2-차원 인덱스에 표시하면 그림 7과 같다. □

이와 유사하게 Grust[18]와 Chien 등[16]은 XML 질의 처리를 위하여 엘리먼트의 (*start, end*) 값을 다차원 인덱스에 저장하는 방법을 제안하였다. Grust와 Chien 등의 방법은 XML 질의 조건을 만족하는 엘리먼트들을

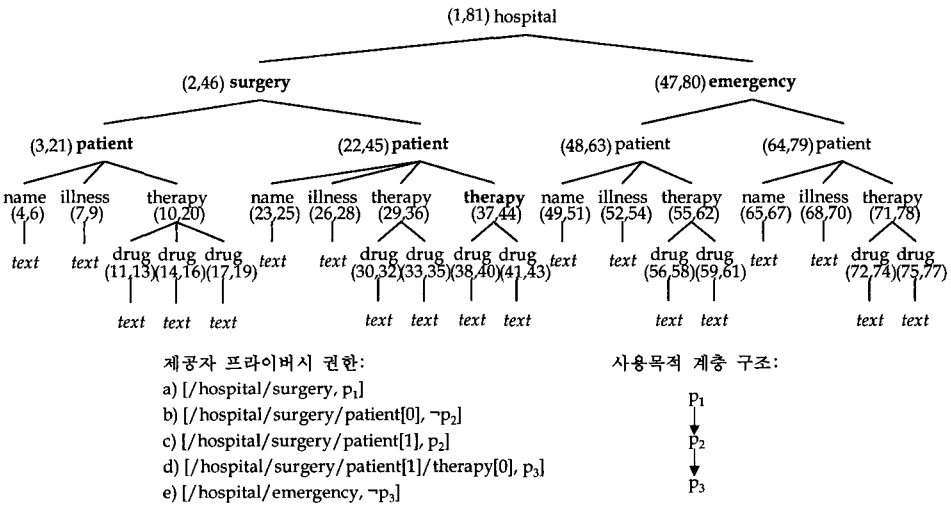


그림 6 XML 문서와 제공자 프라이버시 권한 부여의 예

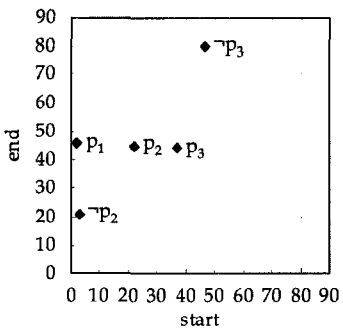


그림 7 부여된 권한을 권한이 부여된 엘리먼트의 (*start*, *end*) 값으로 2-차원 인덱스에 표시한 예

효율적으로 찾아내기 위하여 다차원 인덱스를 사용하며, 본 논문의 방법은 가장 가까운 조상 엘리먼트에 부여된 권한을 효율적으로 찾아내기 위하여 다차원 인덱스를 사용한다.

명시적 권한만을 저장하는 권한 인덱스와 달리, 각각의 엘리먼트마다 부여된 명시적 권한과 묵시적 권한을 모두 저장하는 방법도 고려할 수 있다. 하지만, 이 방법은 저장 공간의 낭비가 심해진다는 문제가 발생한다[8]. 묵시적 권한은 명시적 권한으로부터 유도되므로 기존의 모든 XML 보안 모델에서도 묵시적 권한까지 별도로 저장하지 않는다. 최근에 Yu 등[8]은 명시적 권한이 모든 엘리먼트에 부여되어 있을 때, 저장 공간을 줄이기 위하여 인접한 엘리먼트에 부여된 권한들을 압축하여 표현하는 방법을 제안하였다. 이러한 기존의 연구로 볼 때, XML 데이타베이스에서 모든 엘리먼트마다 부여된 권한을 저장하는 것은 일반적이지 않다.

4.3 권한 인덱스와 최근접 질의를 사용하는 액세스 통제 방법

XML 문서의 특정 엘리먼트에 부여된 권한은 후손 엘리먼트에 대한 묵시적 권한을 내포하며, 묵시적 권한은 후손 엘리먼트에 부여된 다른 명시적 권한에 의하여 오버라이드된다. 따라서, 검사하려는 엘리먼트에 대한 권한은 가장 가까운 조상 엘리먼트에 부여된 권한에 의해 내포된다. 이러한 조상 엘리먼트에 부여된 권한을 **최근접 조상 권한**(*nearest ancestor authorization*)이라 부르고 정의 5에서 이를 정형적으로 정의한다. 그리고, 보조정리 1에서 이를 구하는 방법을 제시한다.

정의 5. 엘리먼트 e의 **최근접 조상 권한**인 $auth_{naa}$는 다음의 두 조건을 만족하는 권한이다: (1) $auth_{naa}$는 엘리먼트 e 혹은 e의 조상 엘리먼트에 부여된 명시적 권한이며, (2) 엘리먼트 e와 $auth_{naa}$가 부여된 엘리먼트 사이의 패스 상의 엘리먼트에 다른 명시적 권한이 존재하지 않는다.

보조정리 1. 번호화 방식을 사용할 때, 엘리먼트 e의 **최근접 조상 권한**인 $auth_{naa}$는 $start(auth) \le start(e) \wedge end(auth) \ge end(e)$를 만족하는 권한 $auth$ 중에서 $|start(e) - start(auth)|$의 값을 최소로 하는 권한이다.

즉, $auth_{naa}$는 엘리먼트 e의 upper-left 영역에 위치한 권한 중에서 엘리먼트 e와의 *start* 값의 차이가 가장 작은 권한이다. 여기에서 $start(e)$, $end(e)$는 엘리먼트 e의 *start*, *end* 값을 나타내며, $start(auth)$, $end(auth)$는 권한 $auth$가 부여된 엘리먼트의 *start*, *end* 값을 나타낸다.

증명. 보조정리 1에 따라 구해진 권한은 정의 5의 두 조건을 모두 만족함을 보인다. 첫째, 번호화 방식의 정의에 따라 권한 $auth_{naa}$는 엘리먼트 e 혹은 e의 조상 엘리먼트에 부여된 권한이므로 조건 (1)을 만족시킨다. (권한이 부여된 e의 조상 엘리먼트를 e_{naa}라고 부른다.) 둘째, $|start(e) - start(auth)|$의 값을 최소로 하는 권한이 $auth_{naa}$이고, 엘리먼트 e와 엘리먼트 e_{naa}의 패스 상의 엘리먼트에 다른 명시적 권한 $auth_{exp}$가 존재한다고 가정한다. 그러면, 번호화 방식의 정의에 따라 $start(e_{naa}) < start(auth_{exp}) < start(e)$ 이다. (여기서 $start(auth_{naa}) = start(e_{naa})$ 이다.) 이는 $start$ 값의 차이가 최소인 권한이 $auth_{naa}$라는 가정에 위배된다. 따라서, 엘리먼트 e와 엘리먼트 e_{naa}의 패스 상의 엘리먼트에는 다른 권한이 존재하지 않으므로 조건 (2)를 만족시킨다. □

권한 인덱스와 최근접 질의를 사용하는 액세스 통제 알고리즘을 제안하며, 이를 *Nearest Ancestor Filtering*이라 부른다. 본 알고리즘은 질의를 수행한 다음 매 질의 결과마다 보조정리 1에 따라 최근접 조상 권한을 구하여 권한 검사를 수행한다. 이때, 최근접 조상 권한은 최근접 질의[10,11] 기법을 활용하여 구할 수 있다. 마지막으로, 질의 결과에 대한 액세스가 허용되는지 검사하기 위해 최근접 조상 권한의 사용목적이 질의 사용목적을 내포하는지 검사한다. 그림 8은 알고리즘 Nearest Ancestor Filtering을 나타낸다.

5 성능 평가

본 절에서는 제안하는 액세스 통제 방법의 성능을 기존의 XML 액세스 통제 방법들과 비교하여 성능 평가 결과를 제시한다. 제 5.1 절에서는 성능 평가를 수행한 실험 데이타와 실험 환경을 소개하고, 제 5.2 절에서는 실험 결과를 설명한다.

5.1 실험 데이타 및 실험 환경

본 실험에서는 제 2.3 절에서 설명한 하향식(top-down), 상향식(bottom-up) 액세스 통제 방법과 제 4.3 절에서 제안한 Nearest Ancestor Filtering을 비교한다. 비교를 위한 척도(measure)는 각 방법을 사용하여 권한 검사를 수행할 때의 질의 처리 시간(wall clock time)이다.

데이타셋의 종류에 따른 성능 변화를 측정하기 위해, 합성 데이타인 XMark[19] 벤치마크 데이타와 실제 데이타인 TreeBank[20] 데이타를 사용한다. XMark 벤치마크 데이타는 하나의 큰 XML 문서로 구성되며, 본 실험에는 **데이타베이스 크기** 변화에 따른 성능 변화를 측정하기 위해 10MB, 100MB, 1GB의 XML 문서를 사용한다. 이 데이타는 유사한 형태의 서브트리가 동일한 레벨에서 여러번 반복되어 나타나는 특성을 가진다. TreeBank 데이타는 약 86MB의 XML 문서로 구성되며, XML 엘리먼트가 여러 레벨로 반복적으로 중첩되어 있는 특성을 가진다.

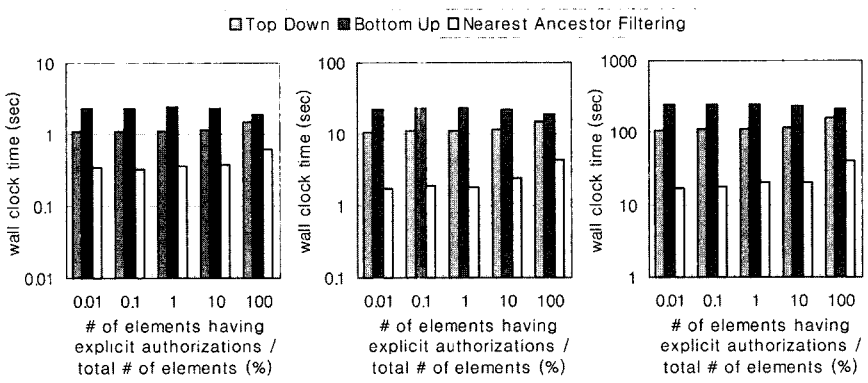
명시적 권한의 개수에 따른 성능 변화를 측정하기 위해, 데이타셋의 전체 엘리먼트의 개수에 대한 명시적 제공자 프라이버시 권한이 부여된 엘리먼트의 개수를 0.01%에서 100%까지 변화시킨다. 권한 (o, p)에서 o를 결정하기 위해, 데이타셋에서 0.01%~100%의 엘리먼트를 랜덤하게 선택한다. 이때, 루트 엘리먼트에 명시적

```
Algorithm Nearest Ancestor Filtering
Input: (1) a set of XML query results R and a usage purpose p of a query
       (2) the authorization index storing provider privacy authorizations
Output: authorized query results
Algorithm:
01:  for each query result r ∈ R do
         /* search the authorization index */
02:      Find the nearest ancestor authorization auth_naa of r according to Lemma 1;
03:      if the usage purpose of auth_naa implies the usage purpose p then
04:          output r;
```

그림 8 액세스 통제 알고리즘 Nearest Ancestor Filtering

dataset	simple queries	complex queries
XMark	(1) *//person//interest* (2) *//person//education*	*//site//open_auctions//open_auction//bidder//increase*
TreeBank	*//NP//NN*	*//ADJP//SBAR//VP//NP//PP*

그림 9 실험에 사용한 XML 질의



(a) 10MB database. (b) 100MB database. (c) 1GB database.

그림 10 XMark 벤치마크 데이타에서 *//person//interest* 질의의 처리 시간

권한이 부여되도록 보장하기 위해 루트 엘리먼트를 먼저 선택한다. 권한 충돌이 발생하지 않는다면 여러 개의 권한이 동일한 엘리먼트에 부여될 수 있다. 권한 (o, p)에서 p를 결정하기 위해, 5개의 사용목적 계층 구조에서 균등하게 사용목적을 선택한다. 긍정적 프라이버시 권한과 부정적 프라이버시 권한은 9:1의 비율로 부여한다.

질의 형태에 따른 성능 변화를 측정하기 위해, $//e_A//e_B$ 형태의 단순한 질의와 $//e_A//e_B//e_C//e_D//e_E$ 형태의 복잡한 질의를 수행한다. 같은 형태의 질의들은 유사한 경향을 가지므로, 각각의 데이타셋 별로 대표적인 경향을 나타내는 그림 9의 질의에 대해서 실험 결과를 제시한다. $//e_A//e_B$ 형태의 질의에서 e_A와 e_B의 선택율(selectivity)에 따른 성능 비교를 수행하기 위해, XMark 벤치마크 데이타에 대해서는 두 가지 종류의 단순한 질의를 사용한다. 이때, *person* 엘리먼트의 선택율은 0.015이고, *interest* 엘리먼트의 선택율은 0.023이며, *education* 엘리먼트의 선택율은 0.004이다.

실험은 450MHz CPU와 512MB 메모리를 가진 SUN Ultra 60 워크스테이션에서 수행한다. 권한 인덱스를 위한 다차원 색인으로는 MLGF[21,22]를 사용하며,[5] 데이타 및 색인 페이지의 크기는 4096 바이트로 한다.

5.2 실험 결과

데이타베이스 크기와 명시적 권한의 개수에 따른 효과

그림 10은 데이타베이스 크기 변화와 명시적 권한의 개수 변화에 따른 *//person//interest* 질의의 처리 시간을 나타낸다. Nearest Ancestor Filtering은 기존의 방법들에 비해 크게 향상된 성능을 보인다. Nearest Ancestor Filtering은 기존의 XML 액세스 통제 방법에 비해 그림 10(a)에서는 2.5~6.6배의 성능 향상을 보였고, 그림 10(b)에서는 3.5~12.7배의 성능 향상을 보였으며, 그림 10(c)에서는 3.9~13.9배의 성능 향상을 보였다. 데이타베이스 크기가 커지더라도 Nearest Ancestor Filtering은 기존의 방법에 비해 항상 좋은 성능을 보인다. 이는 권한 인덱스를 검색할 때 최근접 질의 기법이 권한 인덱스의 검색 공간(search space)을 효과적으로 줄여주기 때문이다[10,11].

명시적 권한이 0.01%~10%의 엘리먼트에 부여되어 있을 때, Nearest Ancestor Filtering의 성능은 명시적 권한의 개수에 의해 거의 영향을 받지 않는다. 이는 권한 인덱스를 검색하는 횟수는 질의 결과의 개수에 의해 결정되므로 항상 일정하며, 권한 인덱스를 검색하는 비용도 거의 증가하지 않기 때문이다. 그러나, 명시적 권한이 모든 엘리먼트(100%)에 부여되어 있을 때, Nearest Ancestor Filtering의 질의 처리 시간이 다소 증가하였는데, 이는 권한 인덱스의 깊이가 1 증가하였기 때문이다. 이러한 결과로 볼 때, Nearest Ancestor Filtering의 질의 처리 시간은 명시적 권한의 개수가 증가함에 따라 로그 비율(log scale)로 증가함을 알 수 있다.

질의 형태에 따른 효과

그림 11은 데이타베이스 크기 변화와 명시적 권한의 개수 변화에 따른 *//person//education* 질의의 처리 시간을 나타낸다. Nearest Ancestor Filtering은 기존의 방법들보다 모두 좋은 성능을 보인다. Nearest Ancestor Filtering은 기존의 XML 액세스 통제 방법에 비해 그림 11(a)에서는 1.3~3.7배의 성능 향상을 보였고, 그림 11(b)에서는 1.8~8.6배의 성능 향상을 보였으며, 그림 11(c)에서는 2.0~9.6배의 성능 향상을 보였다. 여기에서는 그림 10과 반대로 상향식 방법이 하향식 방법

5) 본 실험에서는 MLGF를 사용하나, 다차원 점을 다룰 수 있는 다른 색인도 마찬가지로 사용할 수 있다. 그러한 인덱스의 예로는 R-tree[23], buddy tree[24], quad tree[25] 등이 있다.

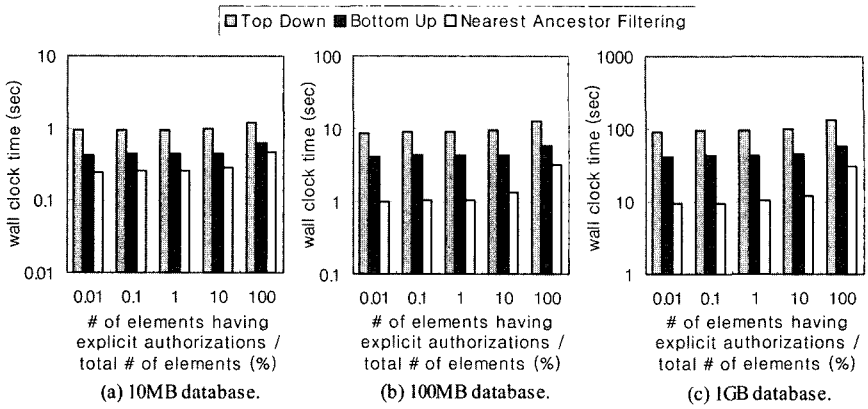


그림 11 XMark 벤치마크 데이타에서 *//person//educaion* 질의의 처리 시간

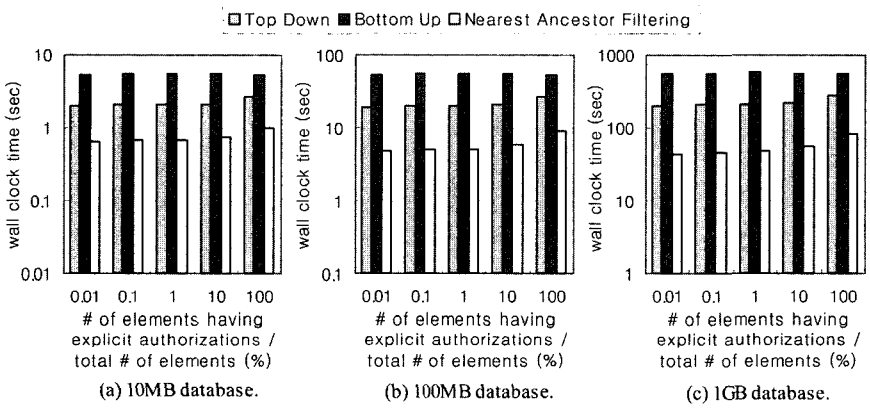


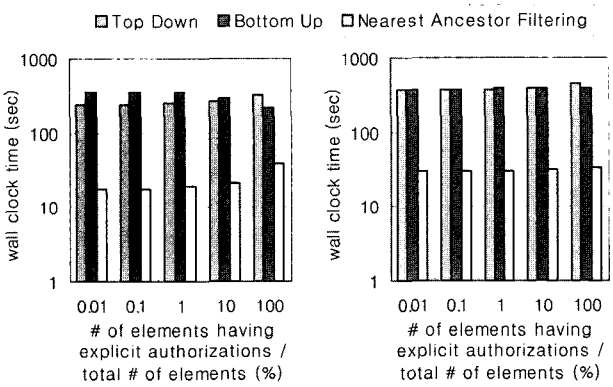
그림 12 XMark 벤치마크 데이타에서 *//site//open_auctions//open_auction//bidder//increase* 질의의 처리 시간

보다 항상 더 좋은 성능을 보인다. 이는 *education* 엘리먼트의 선택율이 *person* 엘리먼트의 선택율보다 낮으므로 상향식으로 패스를 탐색하며 질의를 처리하는 방법이 더 유리하기 때문이다. 그럼에도 불구하고, 그림 10의 결과와 더불어 Nearest Ancestor Filtering은 질의에 사용된 엘리먼트들의 선택율과 관계 없이 기존의 방법들보다 좋은 성능을 보인다는 것을 알 수 있다.

그림 12는 데이타베이스 크기 변화와 명시적 권한의 개수 변화에 따른 길이가 긴 경로식을 가지는 보다 복잡한 형태의 질의인 *//site//open_auctions//open_auction//bidder//increase* 질의의 처리 시간을 나타낸다. Nearest Ancestor Filtering은 복잡한 질의에서도 기존의 방법들에 비해 좋은 성능을 보인다. 이는 복잡한 질의에서 질의 결과를 얻기 위해 구조적 조인[14-16]을 수행하는 비용은 증가하지만, 질의 결과를 얻은 후 권한 인덱스를 검색하여 권한 검사를 수행하는 비용은 질의 형태에 의해 영향을 받지 않기 때문이다. Nearest Ancestor Filtering은 기존의 XML 액세스 통제 방법에 비해 그림 12(a)에서는 2.6~8.4배의 성능 향상을 보였고, 그림 12(b)에서는 3.0~11.2배의 성능 향상을 보였으며, 그림 12(c)에서는 3.4~12.7배의 성능 향상을 보였다.

XML 데이타의 깊이에 따른 효과

그림 13은 TreeBank 데이타(86MB)에 대한 명시적 권한의 개수 변화에 따른 *//NP//NN* 질의와 *//ADJP//SBAR//VP//NP//PP* 질의의 처리 시간을 나타낸다. Nearest Ancestor Filtering은 TreeBank 데이타와 같이 XML 엘리먼트가 여러 레벨로 반복적으로 중첩되어 있는 데이타셋에서도 기존의 방법들에 비해 좋은 성능을 보인다. 이는 권한 인덱스를 검색하는 비용이 저장된 명시적 권한의 개수에 의해서만 영향을 받으며, XML 데이타의 깊이에 의해서는 영향을 받지 않기 때문이다. 그림 13(a)에서 Nearest Ancestor Filtering은 하향식 방법에 비해 8.2~13.6배의 성능 향상을 보였고, 상향식 방법에 비해 5.5~20.3배의 성능 향상을 보였다.



(a) *//NP//NN* query. (b) *//ADJP//SBAR//VP//NP//PP* query.

그림 13 TreeBank 데이타(86MB)에서 단순한 질의와 복잡한 질의의 처리 시간

6. 결 론

본 논문에서는 기존의 히포크라테스 관계형 데이타베이스 모델을 XML 데이타베이스에 적용할 수 있도록 확장한 **히포크라테스 XML 데이타베이스** 모델을 제안하였다. 본 모델에서는 히포크라테스 관계형 데이타베이스 모델에서 제안된 개념들인 프라이버시 권한, 프라이버시 선호 및 프라이버시 정책, 데이타 레코드의 사용목적을 XML 데이타의 트리 형태의 계층 구조에 맞게 확장하였으며, 확장된 개념들을 정형적으로 정의하였다.

히포크라테스 XML 데이타베이스에서의 프라이버시 보호를 위해, **권한 인덱스**와 **최근접 질의 기법**을 활용하는 효율적인 액세스 통제 방법을 제안하였다. 권한 인덱스는 최근접 질의 기법을 활용함으로써 가장 가까운 조상 엘리먼트에 부여된 권한을 효율적으로 찾을 수 있게 해준다. 최근접 질의 기법을 도입함의 정당성을 보조정리 1에서 증명하였다. 그리고, 이러한 액세스 통제 방법을 알고리즘 Nearest Ancestor Filtering으로 구현하였다.

제안한 액세스 통제 방법의 효율성을 입증하기 위해, 합성 데이타셋과 실제 데이타셋을 사용하여 많은 실험을 수행하였다. 실험 결과, Nearest Ancestor Filtering은 하향식 방법에 비해 최대 13.6배, 상향식 방법에 비해 최대 20.3배까지 성능을 향상시킴을 보였다. Nearest Ancestor Filtering은 데이타베이스 크기, 명시적 권한의 개수, 질의 형태, XML 데이타의 깊이에 관계 없이 항상 기존의 XML 액세스 통제 방법보다 좋은 성능을 보였다. 이러한 결과들은 권한 인덱스와 최근접 질의 기법을 사용한 액세스 통제 방법의 효율성을 입증한 것이다.

본 논문의 주요 공헌은 1) 히포크라테스 관계형 데이타베이스 모델을 히포크라테스 XML 데이타베이스 모델로 확장하고 2) 제안한 모델 상에서 권한 인덱스와 최근접 질의 기법을 사용하는 효과적인 액세스 통제 방법을 제안한 것이다. 본 연구의 결과로서 추후 연구를 위한 정형적인 기반과 상용 XML DBMS에 구현될 수 있는 실용적인 방법을 제안한 것으로 사료된다.

참 고 문 헌

[1] Information and Privacy Commissioner of Ontario, "Intelligent Software Agents: Turning a Privacy Threat into a Privacy Protector," Apr. 1999.

[2] Information and Privacy Commissioner of Ontario, "An Internet Privacy Primer: Assume Nothing," Aug. 2001.

[3] Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y., "Hippocratic Databases," In *Proc. 28th Int'l Conf. on Very Large Data Bases*, Hong Kong, China, Aug. 2002.

[4] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J., The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation, Apr. 2002.

[5] Bertino, E., Castano, S., Ferrari, E., and Mesiti, M., "Specifying and Enforcing Access Control Policies for XML Document Sources," *World Wide Web Journal*, Vol. 3, No. 3, pp. 139~151, 2000.

[6] Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., and Samarati, P., "A Fine-Grained Access Control System for XML Documents," *ACM Trans. On Information and System Security*, Vol. 5, No. 2, pp. 169~202, May 2002.

[7] Gabillon, A. and Bruno, E., "Regulating Access to XML Documents," In *Proc. 15th Annual Working Conference on Database and Application Security*, Niagara on the Lake, Ontario, Canada, pp. 299~314, July 2001.

[8] Yu, T., Srivastava, D., Lakshmanan, V. S., and

Jagadish, H. V., "Compressed Accessibility Map: Efficient Access Control for XML," In *Proc. 28th Int'l Conf. on Very Large Data Bases*, Hong Kong, China, Aug. 2002.

[9] Cho, S., Amer-Yahia, S., Lakshmanan, V. S., and Srivastava, D., "Optimizing the Secure Evaluation of Twig Queries," In *Proc. 28th Int'l Conf. on Very Large Data Bases*, Hong Kong, China, Aug. 2002.

[10] Hjaltason, G. R. and Samet, H., "Distance Browsing in Spatial Databases," *ACM Trans. on Database Systems*, Vol. 24, No. 2, pp. 265~318, June 1999.

[11] Roussopoulos, N., Kelley, S., and Vincent, F., "Nearest Neighbor Queries," In *Proc. 1995 ACM SIGMOD Int'l Conf. on Management of Data*, ACM SIGMOD, San Jose, California, pp. 71~79, June 1995.

[12] Berglund, A., Boag, S., Chamberlin, D., Fernández, M. F., Kay, M., Robie, J., and Siméon, J., XML Path Language (XPath) Version 2.0, W3C Working Draft, Nov. 2003.

[13] Rabitti, F., Bertino, E., Kim, W., and Woelk, D., "A Model of Authorization for Next-Generation Database Systems," *ACM Trans. on Database Systems*, Vol. 16, No. 1, pp. 88~131, Mar. 1991.

[14] Li, Q. and Moon, B., "Indexing and Querying XML Data for Regular Path Expressions," In *Proc. 27th Int'l Conf. on Very Large Data Bases*, Italy, pp. 361-370, Sept. 2001.

[15] Al-Khalifa, S., Jagadish, H. V., Koudas, N., Patel, J. M., Srivastava, D., and Wu, Y., "Structural Joins: A Primitive for Efficient XML Query Pattern Matching," In *Proc. 18th Int'l Conf. on Data Engineering*, San Jose, California, Feb. 2002.

[16] Chien, S.-Y., Vagena, Z., Zhang, D., Tsotras, V. J., and Zaniolo, C., "Efficient Structural Joins on Indexed XML Documents," In *Proc. 28th Int'l Conf. on Very Large Data Bases*, Hong Kong, China, Aug. 2002.

[17] Gaede, V. and Gunther, O., "Multidimensional Access Methods," *ACM Computing Surveys*, Vol. 30, No. 2, pp. 170~231, June 1998.

[18] Grust, T., "Accelerating XPath Location Steps," In *Proc. 2002 ACM SIGMOD Int'l Conf. on Management of Data*, ACM SIGMOD, Madison, Wisconsin, June 2002.

[19] Schmidt, A. R., Waas, F., Kersten, M. L., Carey, M. J., Manolescu, I., and Busse, R., "XMark: A Benchmark for XML Data Management," In *Proc. 28th Int'l Conf. on Very Large Data Bases*, Hong Kong, China, pp. 974~985, Aug. 2002.

[20] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B., "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics*, Vol. 19, No. 2, June 1993.

[21] Whang, K.-Y. and Krishnamurthy, R., Multilevel Grid Files, IBM Research Report RC11516, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, Nov. 1985.

[22] Whang, K.-Y. and Krishnamurthy, R., "The Multi-level Grid File - A Dynamic Hierarchical Multidimensional File Structure," In Proc. *Int'l Conf. on Database Systems for Advanced Applications*, pp. 449~459, Tokyo, Apr. 1991.

[23] Guttman, A., "R-Trees: A Dynamic Index Structure for Spatial Searching," In *Proc. 1984 ACM SIGMOD Int'l Conf. on Management of Data*, ACM SIGMOD, Boston, Massachusetts, pp. 47~57, June 1984.

[24] Seeger, B. and Kriegel, H.-P., "The Buddy-Tree: An Efficient and Robust Access Method for Spatial Data Base Systems," In *Proc. 16th Int'l Conf. on Very Large Data Bases*, Queensland, Australia, pp. 590~601, Aug. 1990.

[25] Samet, H., "The Quadtree and Related Hierarchical Data Structures," *ACM Computing Surveys*, Vol. 16, No. 2, pp. 187~260, June 1984.

이 재 길

1999년 2월~현재 한국과학기술원 전자전산학과 전산학전공 박사과정. 1997년 3월~1999년 2월 한국과학기술원 전자전산학과 전산학전공 석사. 1993년 3월~1997년 2월 한국과학기술원 전자전산학과 전산학전공 학사. 2001년 7월~2001년 8월 Visiting Scholar, 미국 HP Labs. 관심분야는 객체관계형 데이타베이스 시스템, 정보 검색, 질의 최적화, XML 데이타베이스, 데이타베이스 보안

한 욱 신

정보과학회논문지 : 데이타베이스 제 31 권 제 4 호 참조

황 규 영

정보과학회논문지 : 데이타베이스 제 31 권 제 4 호 참조