http://dx.doi.org/10.5392/JKCA.2014.14.01.014

벡터 표현을 기반으로 한 XML 동적 레이블링 기법

XML Dynamic Labeling Scheme Based On Vector Representation

홍석희
경성대학교 컴퓨터공학부

Seok Hee Hong(shong@ks.ac.kr)

요약

인터넷 상에서 광범위한 데이터 교환 및 저장의 수단으로 XML에 대한 많은 연구가 진행되어왔다. 특히, XML 문서에 대한 구조 정보를 검색하기 위해서 XML 트리의 각 노드에 레이블을 부여하는 레이블링 기법에 대한 연구가 요구되었다. 레이블링 기법은 각 노드에 레이블을 할당하여 XML 트리 상에서 조상-후손 또는 부모-자식 등의 구조 정보를 검색 할 수 있게 한다. 또한, 레이블링 기법은 기존의 레이블들에 영향을 주지 않도록 동적인 XML 문서 환경을 효율적으로 지원해야 하는 요구 사항을 가진다. 본 논문에서 제안하는 레이블링 기법은 벡터 표현 방식을 기반으로 동적인 XML 문서의 변경을 효율적으로 지원하고 레이블의 길이를 줄임으로서 XML 문서의 레이블 크기를 작게 하여 저장 공간을 적게 요구할 뿐 아니라 검색 시간을 향상시킨다. 성능 실험을 통하여 기존의 레이블링 기법보다 레이블 크기와 검색 시간 등에서 우수함을 보인다.

■ 중심어 : | XML | 구조 정보 | 레이블 | 동적인 변경 |

Abstract

There have been many researches for XML as the international standard to store and exchange data on the internet. Among these research fields, we focus on the techniques labeling the nodes of the XML tree that is required for querying the structural information. A labeling scheme assigns the unique label to the nodes and supports the queries for the structural information such as Ancestor-Descendant and Parent-Child relationships. In this paper, we propose a labeling scheme using vector representation where the assigned labels are not altered although XML documents are changed dynamically. Our labeling scheme reduces the storage requirement for the labels of the XML tree and provides the efficient query by using the fixed-length labels with a short size. Result of performance evaluation shows that our labeling scheme is superior to the previous approaches.

■ keyword : | XML | Structural Information | Labeling | Dynamic Updates |

I. 서 론

XML은 인터넷 상에서 문서 데이터를 보급하고 표현하기 위한 국제표준으로 많은 분야에서 연구되고 있다[1-3]. XML 문서는 트리 구조로 표현될 수 있으며 이를 기반으로 엔리먼트들 사이의 관계를 이용하여 다양

* 이 연구는 2007학년도 경성대학교 연구년 지원에 의하여 수행되었음.
접수일자 : 2013년 11월 02일
수정일자 : 2013년 11월 28일
심사완료일 : 2013년 11월 28일
교신저자 : 홍석희, e-mail : shong@ks.ac.kr

한 질의를 수행한다. XML 문서의 엘리먼트들은 트리의 노드로 표현되어 엘리먼트들 사이의 계층적인 관계를 추출할 수 있게 한다. XML 트리의 각 노드에 유일한 식별자를 연관시키는 레이블링 기법(labeling scheme)으로 XML 문서를 접근하지 않고도 엘리먼트들 사이의 부모-자식 또는 조상-후손 등의 관계를 확인할 수 있다.

레이블링 기법으로 엘리먼트에 해당하는 각 노드에 부여된 레이블은 XPath와 XQuery등을 통해 신속하고 효율적으로 XML 문서에 대한 질의를 처리하게 한다. 다양한 레이블링 기법들은 다음과 같은 조건을 고려하여 노드에 레이블을 연관 짓는다[3-5].

- 트리의 계층 구조 정보의 표현 : 부모와 자식 노드 사이의 관계(parent-child: PC), 조상과 후손 사이의 관계(ancestor-descendant: AD), 형제 노드들 사이의 관계(siblings: SB) 등을 표현한다.
- 문서 순서(document order)의 표현 : XML 문서 내에서 나타난 엘리먼트들 사이의 순서를 표현한다. 이 순서는 트리의 전위 탐색(pre-order traverse)으로 표현된다.
- 동적인 XML 문서의 지원 : 새로운 엘리먼트를 삽입하는 경우 기존의 레이블들에 영향을 주지 않아야 한다. 만일, 엘리먼트를 추가한 후 XML 문서의 전체 또는 일부 노드의 레이블이 변경되어야 한다면 문서의 변경 성능을 악화시키고 결과적으로 XML 문서에 대한 질의 시간을 연장시키게 된다.

모든 레이블링 기법들이 위 조건들을 다 만족하지는 않는다. LSDX 기법[6]은 동적인 XML 문서를 제한적으로 지원한다. 이 기법은 특정한 상황에서 새로운 엘리먼트의 레이블이 중복되는 상황을 발생시킨다. [4]의 레이블링 기법은 벡터 표현방법을 Dewey 기법[2]에 적용하여 위 조건들을 만족시켰다. 특히, 벡터 표현방법으로 특정 엘리먼트의 하위 엘리먼트들을 무한하게 추가하더라도 이론적으로는 기존 노드들에 대한 레이블을 변경할 필요가 없도록 했다. 그러나 [4]의 기법은 트리의 깊이가 깊어질수록 레이블의 길이가 길어지는 문제가 있으며 결과적으로 XML 문서의 레이블 크기가 증가하여 구조정보에 대한 검색과 XML 문서의 동적인 변경 성능을 악화시킨다. 본 논문에서는 단말 노드에 가까울수록 노드의 레이블이 길어지고 형제 노드들 사이의 레이블의 중복을 해결하기 위한 벡터표현 방식의 레이블링 기법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 XML 레이블링 기법에 대한 관련연구를 기술하며, 3장에서는 벡터 표현 방식의 DDE 레이블링 기법에 대해 소개하고 문제점을 분석한다. 4장에서는 본 논문에서 제안하는 벡터표현을 기반으로 하는 레이블링 기법의 구조정보 표현 방식과 동적인 XML 문서 변경 기법을 제시한다. 5장에서는 제안하는 레이블링 기법의 성능평가를 하며 6장에서 결론으로 끝을 맺는다.

II. 관련 연구

레이블링 기법은 XML 문서를 트리로 표현하여 부모-자식, 조상-후손 등의 구조 정보(structural information)를 각 노드에 레이블로 연관시킨다. 또한, XML 문서에 대한 임의의 변경에도 기존의 레이블을 고정적으로 유지시키도록 한다. 본 절에서는 기존에 연구된 다양한 레이블링 기법들을 소개한다.

X. Wu의 연구에서는 트리의 각 노드에 유일한 소수(prime number)를 할당하여 트리의 구조 정보를 표현하였다[2]. 각 노드에 부모 레이블(parent-label)과 자체 레이블(self-label)의 곱으로 레이블(label)을 부여한다. label(A)가 label(B)/self-label(B)와 동일한 경우 노드 A는 노드 B의 부모 노드가 된다. 또한, label(A) mod label(B)가 0인 경우 노드 B는 노드 A의 조상 노드가 된다. 예를 들어, 루트 노드 A의 레이블이 1인 경우 self-label이 2인 자식 노드 B의 레이블은 label(A) × self-label(B)인 2가 된다. 또한, self-label이 소수 5인 B의 자식 노드 C의 레이블은 label(B) × self-label(C)인 10이 된다. label(C) mod label(A)가 0이므로 노드 A는 노드 C의 조상 노드인걸 알 수 있다. 이 기법은 조상-후손과 부모-자식 관계의 표현은 가능하지만 형제 노드들 사이의 관계나 문서 순서의 표현은 쉽지 않다.

각 노드의 레벨 값과 사전식 또는 수리적 순서를 표현하는 식별자의 조합 등을 이용하여 레이블을 할당하

는 전치 표기법(prefix) 기반의 레이블링 기법들에 대한 연구가 있었다. I. Tatarinov의 Dewey 인코딩 기법은 루트 노드에서 현재 노드까지의 경로를 표현하는 벡터로 각 노드의 레이블을 표현한다[2].

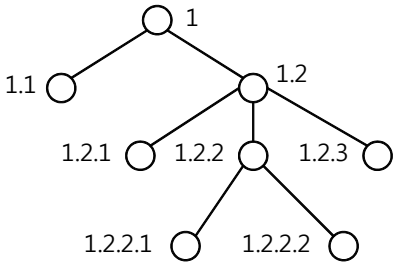


그림 1. Dewey 레이블링 기법

[그림 1]에서 Dewey 레이블링 기법으로 표현한 각 노드는 루트 노드까지의 경로에 해당하는 숫자들이 점으로 연결된 벡터를 레이블로 가짐을 알 수 있다. 특히, 레이블의 가장 우측 숫자는 형제 노드들 사이의 순서를 의미한다. 이 기법은 노드에 연관된 레이블이 정적인 정보를 표현하기 때문에 노드가 삽입되는 경우 기존의 레이블들이 변경되어야 한다. 따라서 동적인 XML 문서 환경에는 적합하지 않다. 이 기법은 XML 문서에 임의의 변경이 가능하도록 연구된 다른 레이블링 기법들에 많은 영향을 주었다. P. O'Neil의 ORDPATH 기법은 Dewey 기법을 기반으로 동적인 XML 문서 환경을 지원하도록 변형하였다[7]. 이 기법은 현재 노드의 자식 노드들에 최하위 숫자에 2씩 더한 홀수로 레이블을 할당한다. 예를 들어, 노드 1.3의 자식 노드들의 레이블은 각각 1.3.1, 1.3.3, 1.3.5가 된다. 노드 1.3.1과 1.3.3 사이에 새로운 노드를 삽입하는 경우 짝수를 추가하여 기존 노드의 레이블과 구별한다. 예를 들어, 노드 1.3.1과 1.3.3 사이에 삽입되는 형제 노드들의 레이블은 1.3.2.1, 1.3.2.3, 1.3.2.5 등이 된다. 이 경우 레이블의 짝수는 레이블의 경로상의 노드 수에 포함시키지 않는다. ORDPATH 기법은 단말 노드의 레벨 값이 큰 깊은 트리의 경우 짝수를 포함하는 레이블의 경우 조상-후손, 부모-자식, 형제 관계 등의 구조 정보의 표현이 제한적이게 되는 문제점이 있다. M. Doung은 Dewey와 ORDPATH 기법과 달리 레이블에 숫자와 문자의 조합을 사용한 LSDX 기법을 제안하였다[6]. 루트에서 현재 노드 사이의 경로 정보와 레벨 값을 이용하여 LP.C의 형식으로 레이블을 할당한다. L은 현재 노드의 레벨 값, P는 부모 노드의 자체 레이블(self-label), C는 현재 노드의 자체 레이블을 의미한다. P와 C는 영문 소문자들로 구성되어 레이블은 사전식 순서로 구조 정보를 표현할 수 있다. 특히, 형제노드들 중 첫 번째 노드의 자체 레이블은 b로 시작한다.

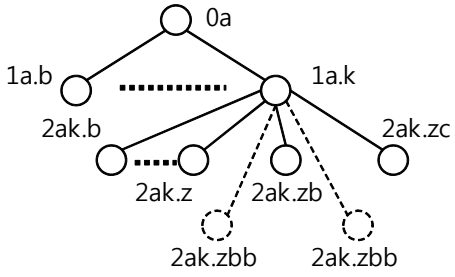


그림 2. LSDX 레이블링 기법

[그림 2]는 LSDX 레이블링 기법의 예를 보여준다. 두 번째 레벨의 노드인 2ak.b의 24번째 형제노드의 레이블은 2ak.z이다. 이 레이블은 부모 노드의 자체 레이블이 ak이고 이 노드의 자체 레이블이 z임을 나타낸다. 이 기법은 동적인 노드 삽입에도 기존 노드들의 레이블에 영향을 주지 않지만 특정 상황에서 삽입된 노드들의 레이블이 중복되는 문제가 발생한다. LSDX 레이블링 과정에 따르면 노드 2ak.z 다음에 삽입되는 노드의 레이블은 2ak.zbb가 되며 노드 2ak.zb 후에 삽입되는 노드의 레이블 역시 2ak.zbb로 할당된다. 이 문제를 해결하기 위해 A. A. Khaing의 연구에서는 LSDX의 레이블에 숫자를 추가하였으나 두 노드 사이에 새로운 노드를 삽입하는 특정 상황에서 중복된 레이블을 부여할 수 있는 유사한 문제가 있다[8]. L. Wu는 2차원 벡터(vector)를 기반으로 노드에 레이블을 연관시키는 연구를 하였다[4][9]. 다음 절에서 소개할 이 기법은 2차원 좌표 값을 표현하는 벡터 표기법을 Dewey 기법에 적용하여 동적인 XML 문서 환경을 효율적으로 지원하고자 하였다.

III. 벡터 기반 DDE 기법

1. 벡터 코드

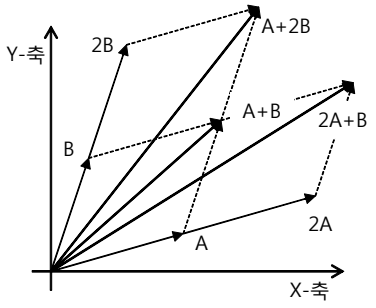


그림 3. 2차원 벡터의 표현

L. Wu의 연구는 XML 트리의 노드에 레이블을 할당하기 위해 [그림 3]에서 볼 수 있는 2차원 벡터 개념을 적용하였다[4][9]. 2차원 벡터 A는 (a, b)의 코드로 표현하며 a와 b는 각각 X-축과 Y-축의 좌표 값이며 X-축으로부터 Y-축으로 b만큼의 각도인 Θ로 변환된다. 벡터 A와 B의 각도를 각각 Θ_a, Θ_b라 했을 때 $\tan(\Theta_a)<\tan(\Theta_b)$의 순서를 가진다. L. Wu의 연구에 의해서 A=(a, b), B=(c, d)인 경우 다음 정의는 동일한 의미를 전달한다.

정의 3.1 벡터 순서: b/a ≦ d/c가 성립하면 벡터 순서에서 A는 B의 앞에 온다. 이는 A ≦v B로 표현된다.

정의 3.2 벡터의 덧셈 : A + B = (a+c, b+d)

정의 3.1과 3.2에 의해서 L. Wu의 연구는 다음의 정리를 제시하였다.

정리 3.1 A≦vB인 경우 A≦v(A+B)≦vB가 성립한다.

증명 : A≦vB이기 때문에 정의 3.1에 의해서 b×c≦a×d가 성립한다. 양변에 a×b를 더하면 b×(a+c)≦a×(b+d)가 됨을 알 수 있고 정의 3.1에 의해서 A≦v (A+B)가 성립함을 증명할 수 있다. 나머지 (A+B)≦vB의 증명도 같은 과정을 적용한다.

L. Wu의 연구는 정리 3.1을 이용하여 $A<_v(A+B)<_vB$와 $A=_v(A+B)=_vB$가 성립함을 증명하였다.

2. 벡터 기반의 DDE 기법

L. Wu은 정리 1에서 증명한 벡터 코드들의 대소 비교를 기반으로 Dewey 기법을 동적 XML 환경에 적용한 DDE(Dynamic DEwey) 레이블링 기법을 제안하였다[9]. 본 절에서는 벡터 기반 DDE 기법을 소개하고 문제점을 분석하고자 한다. DDE 기법은 Dewey 기법의 레이블링 규칙을 사용하지만 레이블의 형태는 벡터 코드 방식을 적용한다. 다음 정의는 벡터 코드가 DDE 레이블로 변환되는 과정을 보여준다.

정의 3.3 DDE 레이블 $x.y_1.y_2....y_m$은 $v_1=(x,y_1)$, $v_2=(x,y_2)$,..., $v_m=(x,y_m)$ 등의 벡터인 경우 벡터 코드 $v_1.v_2....v_m$으로 표현된다.

벡터 순서에 대한 정의 3.1과 유사하게 DDE 레이블들 사이의 순서를 다음과 정의할 수 있다.

정의 3.4 두 DDE 레이블 $A=(v_1,v_2,...,v_m)$과 $B=(w_1,w_2,...,w_n)$에 대해서 다음 두 조건중 하나가 만족되면 $A<_{dde}B$가 성립한다.

- m < n이고 $v_1 =_v w_1$, $v_2 =_v w_2$,...,$v_m =_v w_m$
- $v_1 =_v w_1$, $v_2 =_v w_2$,...,$v_{k-1} =_v w_{k-1}$과 $v_k<_v w_k$를 만족하기 위한 k=min(m, n)가 존재한다.

두 DDE 레이블은 정의 3.2와 같이 다음과 같은 덧셈이 가능하다.

정의 3.5 두 DDE 레이블 $A=(v_1,v_2, ..., v_m)$과 $B=(w_1,w_2, ..., w_n)$에 대해서 A+B는 $(v_1+w_1).(v_2+w_2)$ .... (v_m+w_m)으로 정의된다.

정리 3.2 $A<_{dde}(A+B)<_{dde}B$와 $A=_{dde}(A+B)=_{dde}B$가 성립한다.

증명 : L. Wu[9]의 연구 참고.

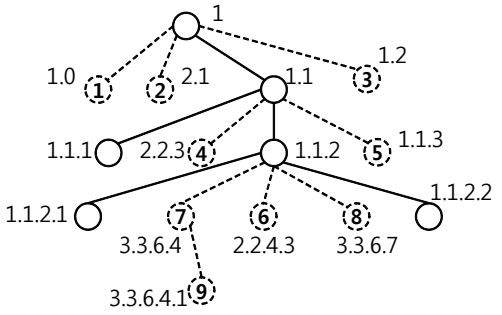


그림 4. 벡터 기반 DDE 레이블링

벡터 기반 DDE 기법은 [그림 4]와 같이 Dewey 기법의 레이블링 규칙과 유사하게 각 노드에 레이블을 할당

하지만 노드가 삽입될 때 벡터 기반의 규칙이 적용된다. [그림 4]는 벡터 기반의 DDE 레이블링 기법을 적용한 XML 트리이다. 실선으로 표시된 노드들은 Dewey 기법을 적용하여 초기에 생성한 것이다. 점선으로 표시된 노드들은 이후에 추가로 삽입된 노드들이며 노드 내의 번호는 추가된 순서를 나타낸다.

XML 트리가 구축된 후에 노드의 삽입 과정을 요약하면 다음과 같다.

1. 좌측(또는 우측)에 형제 노드가 없는 노드 N 이전(또는 이후)에 노드 M을 삽입 : N의 레이블을 $v_1.v_2. \ldots .v_n$라 하면 M은 $v_1.v_2. \ldots .(v_n-1)$(또는 $v_1.v_2. \ldots .(v_n+1)$)의 레이블을 할당한다.
2. 노드 X와 Y 사이에 노드 M을 삽입 : X와 Y의 레이블을 각각 A와 B라 했을 때 M의 레이블은 A+B가 된다.

예를 들어 노드 1과 3은 레이블이 1.1인 노드의 좌측과 우측에 각각 삽입되므로 1.0과 1.2가 된다. 또한, 노드 6은 레이블이 각각 1.1.2.1과 1.1.2.2인 노드들 사이에 삽입되므로 두 레이블의 합인 2.2.4.3이 된다. 정리 3.2에 의해서 $1.1.2.1 <_{dde} (1.1.2.1)+(1.1.2.2)=2.2.4.3 <_{dde} 1.1.2.2$가 성립하므로 이 노드들은 XML 문서 순서를 유지함을 알 수 있다.

벡터 기반 DDE 기법은 트리의 구조 정보를 레이블로 표현할 수 있으며 동적인 XML 문서의 변경을 효율적으로 지원한다. 그러나 이 기법은 다음과 같은 문제점이 있다. 첫째, 트리가 깊어질수록 레이블의 길이가 늘어난다. 예를 들어 노드 7의 6번째 후손 노드의 레이블은 10개 항목으로 구성된다. 둘째, 형제 노드들 사이에 가장 하위 항목을 제외한 레이블의 항목들은 빈번하게 중복되는 상황이 발생한다. 노드 7의 6번째 후손 노드들의 형제 노드들은 9개 항목이 같다. 마지막으로 이와 같은 길어진 레이블로 인해 형제 노드들 사이의 관계 검증에 많은 처리 시간이 필요하게 된다. 이에 본 연구에서는 벡터 기반 DDE 기법의 이와 같은 문제점을 해결하고자 한다.

IV. 개선된 벡터 기반 DDE 기법

1. 단순화된 레이블의 사용

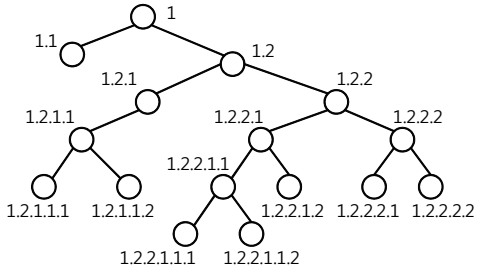


그림 5. 벡터 기반 DDE 기법의 레이블 중복

기존의 벡터 기반 DDE 기법은 형제 노드들 사이에 중복된 레이블 항목들로 인해 트리의 구조 정보 표현에 비효율성이 발생한다. [그림 5]에서 노드 1.2.2.1.1의 두 자식 노드들의 레이블은 공통적으로 1.2.2.1.1의 항목들을 포함한다. 제안하는 레이블링 기법은 이와 같은 레이블의 중복되는 항목을 단순화된 형태로 표현하여 레이블의 길이를 줄이고자 한다. [그림 6]에서 제안하는 레이블링 기법으로 [그림 5]의 레이블들을 단순화하여 할당하였다. 1.2.2.1.1.1의 노드는 제안하는 레이블링 기법으로 (6,1).1의 레이블로 단순화됨을 알 수 있다.

2. 초기 레이블링 과정

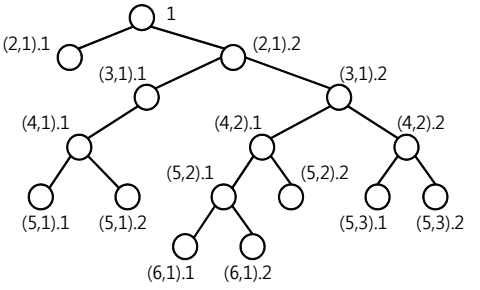


그림 6. 초기 레이블링

본 절에서는 XML 문서를 트리로 변환하여 각 노드에 레이블을 할당하는 초기 과정을 설명한다. 제안하는 레이블링 기법의 단순화된 레이블(이하 SDDE 레이블로 지칭) 형식은 다음과 같다.

정의 4.1 SDDE 레이블은 (x, y).z의 형식을 가진다. x는 루트 노드를 레벨 1로 했을 때 상대적인 레벨 값이다. y는 형제 노드 그룹에 대한 상대위치로 초기에 1로 시작한다. 형제 노드들의 초기 (x,y) 값은 항상 동일하다. z는 형제 노드들 사이의 상대위치로 초기에 1로 시작한다.

[그림 6]에서 (2,1).2의 레이블을 가진 노드의 두 자식 노드들은 각각 (3,1).1과 (3,1).2의 레이블이 할당된다. 3과 1은 각각 레벨과 형제 노드 그룹의 상대위치를 나타낸다. 노드 (5,2).1은 두 번째 형제 노드 그룹이므로 (5,2)의 레벨과 상대위치를 가진다. 제안하는 레이블링 기법은 [그림 5]의 벡터 기반 DDE 기법을 기반을 두기 때문에 (x,y)를 원래의 레이블(이하 DDE 레이블로 지칭한다.)과 연관시켜야 한다. 노드 A의 DDE 레이블이 $(v_1,v_2,...,v_m)$인 경우 다음 정의가 필요하다.

정의 4.2 노드 A의 단순화된 레이블의 (x,y)는 변환함수 S로 다음과 같이 정의된다.

$$S(v_1,v_2,...,v_{m-1}) \rightarrow (m, y)$$

y는 가장 좌측의 형제 노드 그룹을 1로 하는 상대적인 위치를 나타낸다.

정의 4.3 노드 A의 단순화된 레이블 $(x,y).v_m$는 $S(v_1,v_2,...,v_{m-1}).v_m$으로 정의된다.

정의 4.4 노드 A의 SDDE 레이블 (x,y)는 역변환 함수인 S^{-1}로 DDE 레이블로 변환된다.

$$S^{-1}(x,y) \rightarrow (v_1,v_2,...,v_{m-1})$$

정의 4.5 노드 A의 SDDE 레이블 $(x,y).v_m$는 DDE 레이블을 $S^{-1}(x,y).v_m$으로 정의한다.

3. SDDE 레이블의 구조정보 표현

제안하는 레이블링 기법은 XML 문서를 [그림 5]의 DDE 레이블 구조로 역변환될 수 있도록 [그림 6]의 SDDE 레이블로 표현하여 트리의 구조정보를 제공한다. 다음은 노드 A와 B의 SDDE 레이블이 각각 (j, k).v_m와 (x, y).w_n이고 $S^{-1}(j,k)$와 $S^{-1}(x,y)$를 각각 $(v_1,v_2,...,v_{m-1})$와 $(w_1,w_2,...,w_{n-1})$라 할 때 다음은 구조정보를 추출하는 규칙이다.

① 조상-후손 관계 : $m < n$이고 $v_1 =_v w_1$, $v_2 =_v w_3$, ..., $v_m =_v w_m$을 만족하면 A는 B의 조상노드이다.

② 부모-자식 관계 : $m = n-1$이고 $v_1 =_v w_1$, $v_2 =_v w_3$, ..., $v_m =_v w_m$을 만족하면 A는 B의 부모노드이다.

③ 형제 관계 : $j = x$, $k = y$, $v_m < w_n$를 모두 만족하면 B는 A 다음에 오는 형제 노드이다.

④ 문서 순서 : $(v_1,v_2,...,v_m) <_{dde} (w_1,w_2,...,w_n)$인 경우 XML 문서상에서 A가 B를 선행한다.

위 규칙에 의해서 (5,1).1과 (5,1).2는 서로 형제 노드이며 (5,1).1이 (5,1).2 보다 선행함을 알 수 있다. 또한, SDDE 레이블이 (5,1).1인 노드와 (2,1).2인 노드는 $S^{-1}(5,1)$과 $S^{-1}(2,1)$이 각각 (1.2.1.1)과 (1)이므로 (1.2.1.1.1)과 (1.2)의 DDE 레이블을 가진다. 따라서 (5,1).1인 노드는 (2,1).2의 후손 노드임을 알 수 있다.

4. 동적인 XML 문서의 변경

XML 문서를 트리 구조로 변환하여 초기 레이블링을 수행한 후에 동적으로 노드가 추가되는 경우 기존의 레이블은 변경되지 말아야 한다. 제안하는 레이블링 기법은 초기에 XML 문서의 레이블링이 완료된 이후에 추가되는 노드들에 대한 레이블만 할당하면 되기 때문에 동적인 XML 문서 환경을 지원한다. 노드의 동적인 추가를 위해 다음 정리가 필요하다.

정리 4.1 A와 B의 SDDE 레이블을 (h,j).a와 (k,l).b라 할 때 $(h,j).a <_{sdde} (h+k,j+l).(a+b) <_{sdde} (k,l).b$가 성립한다.

증명 : SDDE 레이블 (a,b).c를 DDE 레이블 (a.b.c)라 치환한다면 정의 3.2에 의해서 $(h.j.a) <_{dde} (h+k).(j+l).(a+b) <_{dde} (k.l.b)$가 성립한다. 따라서 $(h,j).a <_{sdde} (h+k,j+l).(a+b) <_{sdde} (k,l).b$도 성립함을 알 수 있다.

위 정리에 의해서 $(4,2).1 <_{sdde} (8,4).3 <_{sdde} (4,2).2$가 성립함을 알 수 있다.

제안하는 레이블링 기법은 노드 B를 XML 트리에 추가하는 경우 다음의 규칙에 따라서 SDDE 레이블이 할당된다. 노드 A와 C의 SDDE 레이블을 각각 $(v_0,w_0).x_0$와 $(v_2,w_2).x_2$라 하자. 또한, level(A)를 노드 A의 트리상의 레벨이라 하자.

① 형제노드인 A와 C 사이에 추가 : $(v_0+v_2,w_0+w_2).(x_0+x_2)$

② 가장 우측 형제 노드 A 다음에 추가되는 경우 : $(v_0,w_0).x_0+1$

③ 가장 좌측 형제 노드 A 이전에 추가되는 경우 : $(v_0,w_0).x_0-1$

④ 노드 A의 유일한 자식 노드로 추가될때 다음 조건에 따라 SDDE 레이블 할당 :

(a) 노드 B가 level(A)+1인 레벨의 유일한 노드인 경우 (level(A)+1,1).1

(b) level(A)+1인 가장 가까운 좌측 노드 A와 가장 가까운 우측 노드 C가 있는 경우 $(v_0+v_2,w_0+w_2).1$

(c) level(A)+1인 가장 가까운 좌측 노드가 없고 가장 가까운 우측 노드 C만 있는 경우 $(v_2,w_2-1).1$

(d) level(A)+1인 가장 가까운 우측 노드가 없고 가장 가까운 좌측 노드 A만 있는 경우 $(v_0,w_0+1).1$

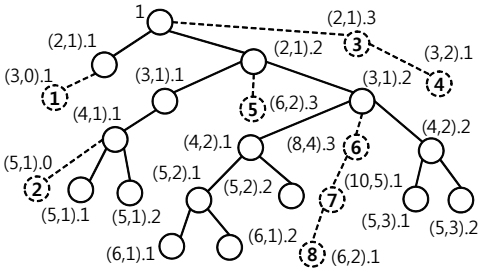


그림 7. 추가된 노드의 SDDE 레이블

위 규칙 ①은 정리 4.1에 의해서 좌측 형제노드와 우측 형제노드 사이에 추가되는 노드 B에 $(v_0+v_2,w_0+w_2).(x_0+x_2)$의 레이블을 할당하여 세 형제노드들 사이에 레이블의 순서가 유지되게 한다. [그림 7]은 [그림 6]의 XML 트리에 8개의 노드를 추가하여 SDDE 레이블링 규칙에 따라 레이블을 할당한 상황을 보여준다. 점선으로 표시된 노드 내의 번호 순으로 추가 된 후의 XML 트리의 상태이다. 1번 노드의 경우 (2,1).1의 레이블인 노드의 유일한 자식 노드이고 같은 레벨의 (3,1).1인 노드가 있으므로 규칙 ④-(c)에 의해 (3,0).1의 레이블이 할당된다. 5번 노드의 경우 좌. 우측에 (3,1).1과 (3,1).2의 형제 노드 사이에 추가되므로 규칙 ①에 의해 (3+3, 1+1).(1+2)인 (6,2).3의 레이블이 할당된다. 또한, 2번 노드의 경우 가장 좌측 형제 노드인 (5,1).1의 좌측에 추가되어 규칙 ③에 의해서 (5,1).0의 레이블이 할당된다. 위 규칙으로 새로 추가되는 노드에 대한 레이블만 할당하므로 기존 노드의 레이블은 변경되지 않는다. 따라서 제안하는 기법은 동적인 XML 문서의 변경을 효율적으로 지원함을 알 수 있다.

5. DDE 레이블의 추출

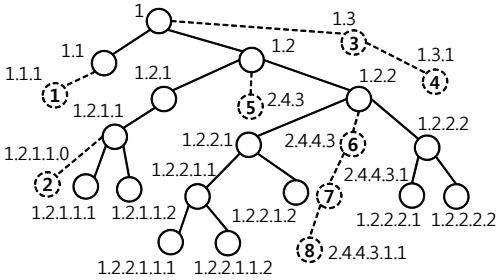


그림 8. DDE 레이블로 역변환된 XML 트리

4절에서 XML 트리에 추가되는 노드들에 SDDE 레이블을 할당하는 규칙에 대해서 알아보았다. SDDE 레이블로 3절의 구조정보 추출 규칙에 따라 XML 트리 구조를 유추하기 위해서는 DDE 레이블들이 반드시 필요하다. 본 절에서는 4절의 규칙으로 SDDE 레이블을 할당하면서 DDE 레이블로 변환하기 위한 규칙을 제시하고자 한다. 정의 4.4의 S^{-1}을 통해 SDDE 레이블이 DDE 레이블로 변환되기 위해 다음과 같은 규칙이 필요하다. 4절의 각 규칙 번호에 대응하는 상황에 근거하여 $S^{-1}(x,y)$는 다음과 같이 DDE 레이블로 변환된다.

① $S^{-1}(v_0,w_0).x_0 + S^{-1}(v_2,w_2).x_2$

② $S^{-1}(v_2,w_2).(x_2+1)$

③ $S^{-1}(v_2,w_2).(x_2-1)$

④ 노드 A의 유일한 자식노드로 추가되고 (a)~(d)의 모든 조건에 대해 $S^{-1}(v_0,w_0).x_0.1$로 변환된다.

위 규칙에 따라서 [그림 7]의 SDDE 레이블로부터 [그림 8]의 DDE 레이블로 변환할 수 있다. 3번 노드의 경우 규칙 ②에 의해 좌측 형제노드인 (2,1).2의 DDE 레이블로부터 $S^{-1}(2,1).(2+1)$로 변환되어 $S^{-1}(2,1)$이 1이므로 1.3의 DDE 레이블이 된다. 4번 노드는 3번 노드의 유일한 자식노드로 추가되므로 규칙 ④에 의해

S-1(2,1).3.1이므로 1.3.1의 DDE 레이블로 변환된다. 또한, 노드 5의 경우 형제노드들 사이에 추가되므로 규칙 ①에 의해 S^{-1}(3,1).1 + S^{-1}(3,1).2로 변환된다. 5번 노드의 SDDE 레이블은 S^{-1}(3,1)이 1.2이므로 1.2.1 + 1.2.2인 2.4.3의 DDE 레이블로 변환된다. 이와 같은 변환 규칙에 따라 [그림 7]의 SDDE 레이블로 부여된 XML 트리는 그림 8의 DDE 레이블로 표현된 XML 트리로 변환될 수 있다. 따라서 3절의 구조 정보 추출 규칙으로 노드들 사이의 트리 구조를 추출할 수 있게 된다.

V. 성능 평가

표 1. XML 문서 데이터

XML 문서	크기(MB)	총 노드 수	최대/평균 Fan-Out	최대/평균 깊이
XMark	113	1666315	25500/3242	12/6
Nasa	23.8	476646	2435/225	10/7
Treebank	85.4	2437666	56384/1623	36/8
DBLP	127	3332130	328858/65930	6/3

본 논문에서 제안한 SDDE 레이블링 기법의 성능을 평가하기 위해 DDE 레이블링 기법을 비교 대상으로 선택하였다. L. Wu[1,3]의 연구에서 다른 레이블링 기법의 성능 평가를 통해서 DDE 레이블링 기법의 성능을 분석한바 있다. 따라서 본 논문에서는 제안하는 SDDE 레이블링 기법과 DDE 레이블링 기법의 성능 평가 결과만을 제시한다. 성능 평가는 Java 1.6.0과 XML 문서의 파싱을 위해 Xerces2 라이브러리를 이용하여 레이블링 기법을 구현하여 수행되었다. 성능 평가에 사용되는 XML 문서 데이터는 워싱턴 대학교의 XML data repository[10]에서 선택하였다. [표 1]은 성능 평가에 사용할 4가지 XML 문서 데이터를 요약한다. 먼저, 표 1의 각 문서를 대상으로 초기 레이블링 작업을 수행하여 레이블의 크기를 비교하였다.

[그림 9]는 DDE 레이블링 기법과 제안하는 SDDE 레이블링 기법으로 4개의 XML 문서에 대한 레이블 크기를 나타낸다. SDDE 레이블의 크기가 DDE 레이블의 크기보다 전반적으로 작은 것으로 나타났다. 특히, Treebank와 DBLP의 경우 그 차이가 더 많은 이유는 이들 문서가 다른 XML 문서에 비해서 평균 자식 노드의 수나 평균 깊이가 크기 때문이다. SDDE 레이블링 기법은 깊이에 관계없이 일정한 길이의 레이블을 유지하지만 DDE 레이블링 기법은 노드의 레벨 값이 커질수록 레이블의 크기도 비례하여 커지기 때문이다.

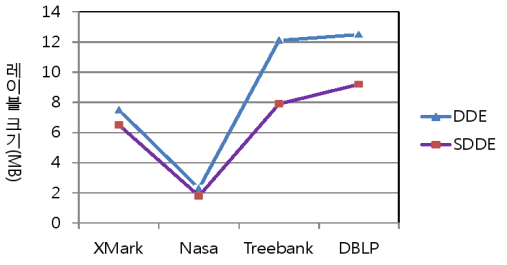


그림 9. XML 문서의 레이블 크기

[그림 10]은 10,000개의 노드를 랜덤하게 삽입한 후 트리 정보에 대한 검색 시간을 평가한 결과이다. [그림 10]에서 XML 문서 내의 순서를 DO, 조상-후손 관계를 AD, 부모-자식 관계를 PC, 형제 관계를 SB로 나타냈다. 검색 시간은 10,000개의 노드들 사이의 트리 정보를 검증하는데 걸리는 시간을 의미한다. DO와 SB의 경우 SDDE 레이블링 기법이 DDE 레이블링 기법보다 우수한 것으로 나타났다. 이는 SB 관계의 경우 DDE 레이블보다 짧아진 SDDE 레이블로부터 직접적으로 유추될 수 있기 때문이다. 그러나 AD와 PC의 경우 근소한 차이로 SDDE 레이블링 기법의 검색 성능이 나쁘게 나왔다. 이는 조상-후손 또는 부모-자식 관계를 검증하기 위해 DDE 레이블로 역변환하는 시간 때문에 DDE 레이블링 기법 보다 검색 성능이 나빠지게 된다.

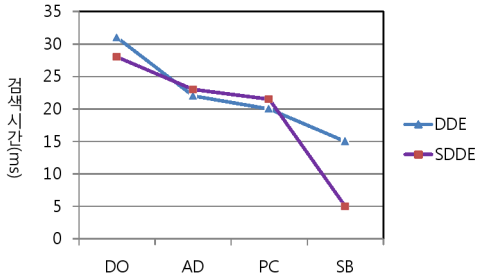


그림 10. XML 문서에 대한 검색시간

VI. 결 론

본 논문에서는 기존의 DDE 레이블링 기법의 문제점을 해결하여 동적인 XML 환경을 위한 레이블링 기법을 제안하였다. 제안하는 SDDE 레이블링 기법은 DDE 레이블링 기법에서 사용한 벡터 표현 방식을 사용하여 XML 문서의 동적인 변경을 효율적으로 지원한다. 그러나 기존의 DDE 레이블링 기법은 XML 트리의 단말 노드에 근접할수록 긴 길이의 레이블과 형제 노드들 사이의 레이블의 중복 문제가 발생한다. 이를 해결하기 위하여 SDDE 레이블링 기법은 노드의 깊이에 관계없이 동일한 길이를 갖는 레이블 형식을 사용하였다. 또한, 형제 노드들 사이에 발생하는 레이블의 중복 요소들을 두 항목으로 표현하여 XML 문서의 전체적인 레이블 크기를 줄이고자 하였다. 결과적으로, 짧고 일정한 길이의 레이블은 XML 트리의 구조 정보를 검색하는 시간을 단축할 뿐 아니라 XML 문서를 위한 레이블의 저장 공간을 효율적으로 활용할 수 있게 하였다. 축약된 SDDE 레이블을 사용하더라도 동적으로 노드들이 추가되어도 여전히 기존의 레이블에 영향을 주지 않는다. 공개된 성능평가용 XML 문서 데이터를 대상으로 기존의 DDE 레이블링 기법과 성능 평가를 수행하여 제안하는 레이블링 기법이 레이블 크기 및 검색 시간에서 우수한 성능으로 평가되었다. 향후 연구 과제로는 DDE 레이블링 기법에 비해서 상대적으로 느린 조상-후손 관계와 부모-자식 관계의 검색 속도를 향상 시킬 수 있는 방안을 연구하고자 한다.

참 고 문 헌

[1] H. Kang, J. S. Yoo, and B. Y. Lee, "XML Repository System Using DBMS and IRS," Int'l J. of Contents, Vol.3, No.3, pp.6-14, 2007.

[2] I. Tatarinov, S. Viglas, K. S. Beyer, J. Shanmugasundaram, E. J. Shekita, and C. Zhang, "Storing and Querying Ordered XML Using a Relational Database System," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp.204-215, 2002.

[3] S. Subramaniam, S. Haw, and P. K. Hoong, "s-XML : An Efficient Mapping Scheme for Storing XML Data in a Relational Database," Proc. 3rd Int'l Conf. on Advanced Computer Theory and Engineering(ICACTE), pp.149-153, 2010.

[4] L. Xu, T. W. Ling, and H. Wu, "Labeling Dynamic XML Documents: An Order-Centric Approach," IEEE Trasns. on Knowledge and Data Engineering, Vol.24, No.1, pp.100-113, 2012.

[5] X. Wu, M. L. Lee, and W. Hsu, "A Prime Number Labeling Scheme for Dynamic Ordered XML Trees," Proc. 20th Int'l Conf. Data Eng.(ICDE), pp.66-78, 2004.

[6] M. Duong and Y. Zhang, "LSDX: A New Labeling Scheme for Dynamically Updating XML Data," Proc. 16th Australasian Database Conf., Vol.39, pp.185-193, 2005.

[7] P. O'Neil and E. O'Neil, "ORDPATHs : Insert-Friendly XML Node Labels," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp.903-908, 2004.

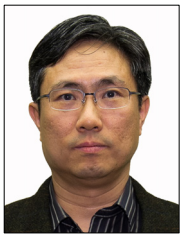
[8] A. A. Khaing and N. L. Thein, "A Persistent Labeling Scheme for Dynamic Ordered XML Trees," Proc. Int't Conf on Web Intelligence, pp.498-501, 2006.

[9] L. Xu, Z. Bao, and T. W. Ling, "A Dynamic Labeling Scheme Using Vectors," Proc. 18th Int'l Conf. Database and Expert Systems Applications(DEXA), pp.130-140, 2007.

[10] http://www.cs.wash ington.edu/research/xmlda tasets.

저 자 소 개

홍 석 희(Seok Hee Hong) 정회원



- 1989년 2월 : 홍익대학교 컴퓨터공학과(공학사)
- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1997년 2월 : 한국과학기술원 전산학과(공학박사)
- 1997년 3월 ~ 8월 : 한국전자통신연구원 박사후연구원
- 1997년 9월 ~ 현재 : 경성대학교 컴퓨터공학부 교수

<관심분야> : XML, 저장구조, 실시간 데이터베이스, 트랜잭션 관리, 소프트웨어 시험