

통신 환경에서 비정형적 구조를 갖는 데이터세트의 효과적인 제어 방법

배 명 남[†] · 최 원^{††} · 이 동 춘^{†††}

요 약

교환기 시스템(Switching System)과 같은 통신 시스템에서는 제안된 이벤트들이 반드시 명시된 시간 제약 내에 완료되어야 한다. 따라서, 시스템에 유지되는 응용 데이터들은 빠른 접근이 가능해야 하며, 동시에 제한된 시간 내에 이벤트의 완료가 보장되어야 한다. 현재, 많은 데이터 시스템들이 사용되고 있지만, 이들은 정형화된 구조와 이에 대한 기본적인 연산들만을 제공하고 있다. 최근 통신 응용에서 데이터의 복잡성이 증가함으로써, 기존의 방식과는 달리, 비정형화된 구조의 표현이 가능하며, 이들에 대해 쉽게 접근 가능한 체계가 요구되고 있다. 이를 위해, 본 논문에서는 비정형화된 다중 응용 환경의 모델링에 적합한 데이터 모델을 소개한다. 모델은 데이터세트에 대한 빠른 접근과 필요한 데이터를 쉽게 추출할 수 있는 체계를 제공한다. 추가로, 모델의 특징을 명확히 하기 위해 몇몇 세부 알고리즘을 함께 설명한다.

An Effective Control Scheme for Unstructured Dataset in the Communication Environments

Myung Nam Bae[†] · Wan Choi^{††} · Dong Chun Lee^{†††}

ABSTRACT

Communication systems, such as Switching System, are operated in the restricted conditions that the suggested events must finish in the time-constraints. Therefore, the data in the systems requires not only rapid access time, but also completion in the restricted time. Many existing data systems have been developed and used in the communication environments. But, the system construct a structural scheme and provide users with basic data services only. In recent, as the complexity of data in the communication area is rapidly increasing, it requires the data system which can represent the unstructured dataset and complete the data access in this dataset on the restricted condition. In this paper, we propose the data model which is suitable to the unstructured multi-dataset environment. The data model supports the rapid data access for unstructured dataset and enables users to easily retrieve data needed at the execution. In addition to, we define the several algorithms to clarify the structure of our model.

키워드 : 비정형적 구조의 데이터세트(the unstructured dataset), 식별자기반 관계성(OID-based Relationship), 항해기반 커서(the navigation-based Cursor)

1. 서 론

현재의 통신 환경은 시스템 내에 유지되는 통신데이터의 복잡성이 크지 않아 대부분 정형화된 형태로 표현되고 데이터 룩업(Lookup)과 같은 단순 검색 기능으로 충분하다. 통신 응용에서 이러한 특성은 데이터의 시간 제약적인 처리를 보장하는데 큰 어려움이 없었다[1]. 최근 통신 시스템 환경은 매우 다양한 변화를 맞고 있다. 큰 변화로서는 향후 인터넷망과의 연계를 통해 다양한 인터넷 서비스와의 연계를 수용하여야 하며, 이로 인해 기존과 같은 정형화된 데이

터의 저장/관리만으로는 처리 시간 및 확장에 많은 한계를 갖게 된다. 또한, 개방형 구조와 범용성을 갖는 통신 시스템 소프트웨어 환경에 대한 통신 사업자의 요구가 증대되면서, 통신 환경 전반적으로 기존의 모델에 비해 보다 복잡하고 포괄적인 시맨틱(semantic)을 갖는 통신 데이터의 정의 및 관리 방법을 필요로 하게 되었다. 다른 측면으로, 최근 초고속 정보 통신망을 기반으로 한 인터넷 서비스가 활성화됨에 따라 높아지는 데이터 전송속도에 비례하여 서비스 제공 시스템의 데이터 처리량이 많아지고 있다. 그 결과 다량의 복잡한 구조의 데이터에 대해, 보다 빠른 연산 처리가 필요하게 되었다[2, 3]. 보편적으로, 다량의 데이터를 관리하는 방법으로 데이터베이스 시스템을 사용하고 있지만, 디스크에 기반한 데이터 처리 시스템들은 다양한 기능에

† 정 회 원 : 한국전자통신연구원 네트워크 S/W 플랫폼팀 선임연구원
 †† 정 회 원 : 한국전자통신연구원 네트워크 S/W 플랫폼팀장
 ††† 정 회 원 : 호원대학교 컴퓨터학부 교수
 논문접수 : 2001년 8월 8일, 심사완료 : 2001년 10월 15일

비해, 규모가 지나치게 크고 데이터 처리 성능이 떨어져 통신 환경에서 요구되는 데이터의 처리 속도 및 시스템 적용 측면에서 적합하지 않다. 따라서, 교환기와 같이 매우 시간 제한적인 시스템 환경에서는 매우 빠르고 균일한 접근 시간을 제공하도록 하기 위해, 운용될 모든 데이터셋을 주 메모리에 상주시키고 이에 대한 일관성과 고성능 접근을 제공하는 방법이 사용되었다[1].

보편적으로 통신데이터를 다루기 위한 데이터 모델과 시스템들은 공통적으로 데이터에 대한 접근 속도의 향상을 가장 주요한 목적으로 하고 있어, 고수준의 데이터 모델을 제공하지 않았지만, 최근, 하드웨어의 가격하락과 비약적인 성능 향상, 그리고 통신용 객체지향 언어의 지원[4], 교환기 소프트웨어의 복잡성 증가, 통신 소프트웨어의 컴포넌트화 요구 등으로[3], 기존의 데이터 처리 방법들[1,5,6]과는 달리 비정형화되고 복잡한 데이터의 표현을 보장하면서도 빠르고 균일한 데이터 처리를 제공하는 방법이 전 세계적으로 주목을 받고 있다.

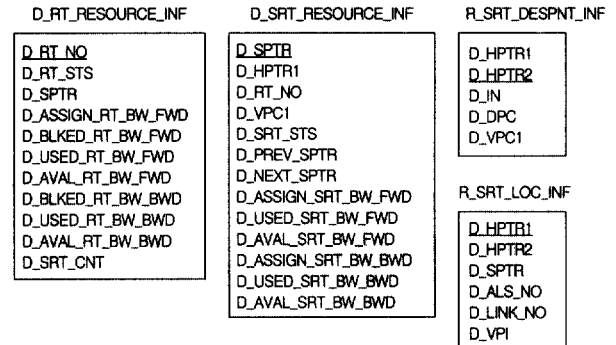
본 논문에서는 통신 운용 환경에서 요구되는 모든 데이터들을 주메모리에 상주시키고 이들 데이터에 대한 고수준의 데이터 표현 방법과 빠르고 균일한 접근을 제공하는 방법을 기술한다. 구체적인 방법으로는 1) 데이터에 대한 스키마(Schema)를 정의할 때, REF/SET, inverse를 사용해 데이터셋들간의 관계성을 쉽게 표현하도록 하고, 2) 다중 데이터셋에 대한 접근 연산시 이를 사용한 항해를 통해, 부가적인 메모리의 사용과 별도의 알고리즘을 사용하는 기존의 방식과는 달리 보다 빠르고 균일한 접근 시간을 제공하는 방법에 대해 설명한다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 통신 시스템 중 기존의 ATM 교환기에서 사용되는 데이터 처리 방법과 이때 발생할 수 있는 문제점에 대해 간략히 소개한다. 3장에서는 비정형화된 데이터를 표현하기 위해 고안한 데이터 모델에 대해, 4장에서는 데이터셋에 대해 균일하고 빠른 접근을 제공하도록 개발한 항해기반 커서에 대해 기술한다. 마지막으로, 5장에서는 결론 및 향후 연구과제에 대해 기술한다.

2. 관련 연구

DREAM-S는 지난 10여 년간 전전자 교환기에서 호 처리 및 가입자 데이터 관리를 위해 사용되었으며, 실제계의 정보를 테이블 형태로 구조화하여 관리한다[1]. (그림 1)은 DREAM-S사용의 한 예로서, 출 증계호 라우팅 관련 데이터를 보이고 있다. 데이터셋 D_RT_RESOURCE_INF는 루트 자원에 관한 정보를 저장관리하며, D_SRT_RESOURCE_INF는 서브 루트가 속해있는 시스템내 물리적 위치 정보를 유지한다. 이때, 각 데이터셋들은 서로 독립되어 관리되

고 있으며, 이들간의 설계 시멘틱에 따라 관련성과 제약을 명시할 데이터 모델을 갖고 있지 않다.



(그림 1) 출 증계호의 라우팅 처리 관련 데이터

이러한 구성을 갖는 데이터셋은 그 구조와 처리를 단순화하는 반면, 모델의 부재로 데이터셋들간의 시멘틱을 인식할 수 없기 때문에 실제 실제계의 문제 해결에 많은 제약점을 가진다. 예를 들어, D_RT_RESOURCE_INF와 같이 하나의 루트 자원이 다수개의 서브 루트를 갖는 구조의 경우에, D_RT_RESOURCE_INF에서 유일한 값으로 정의된 D_RT_NO값을 키로 하여, D_SRT_RESOURCE_INF에서 동일한 값의 D_RT_NO를 가진 데이터들을 추출할 수 있다. 이를 위해서는 D_SRT_RESOURCE_INF의 D_SPTR 값은 항상 D_SRT_RESOURCE_INF.D_SPTR에 존재하는 값을 가져야 하는 참조 무결성(Referencial Integrity)을 개발자(혹은 응용 시스템)가 항상 보장하도록 유지하여야 한다. 이러한 제약을 보장하기 위해서는 응용 시스템의 복잡성이 증가할 뿐만 아니라, 데이터 처리 시스템으로의 잦은 접근으로 전체적인 성능의 하락을 가져오는 요인이 된다.

또한, 단일 데이터셋에 대해 단순한 Lookup 형태의 검색이 아니고, 다중 데이터셋에 대한 정보 추출이 요구될 경우에는 재구성된 결과 데이터셋을 시스템 내부에 유지하고, 이에 대한 순차적인 접근을 지원할 수 있는 체계로 커서(Cursor)가 필요하다[7]. 커서는 데이터셋에 대하여 현재의 질의문맥에 일치하는 데이터들을 순차적으로 접근하기 위해 사용하는 구조이다. 데이터 처리 시스템에서 커서가 필요한 이유는 데이터베이스 시스템이 기본적으로 크기를 한정하지 않는 데이터들의 집합을 다루는 반면, 범용 프로그래밍 언어에서는 이들을 다루는 별도의 수단을 갖지 않기 때문이다. 응용 프로그램에 순차적인 접근을 허용하기 위해, 기존의 커서는 연산 후 얻어진 결과 데이터들을 그 크기만큼 주 메모리에 별도로 저장하는 방식을 사용하기 때문에, 주기억장치의 저장공간을 낭비하는 문제가 있고, 해당 저장공간에 접근하기 위한 별도의 체계를 구축하기 위한 시간과 알고리즘이 부가적으로 필요하다는 문제가 있다. 일반적으로 값을 사용한 커서의 구축은 매우 많은 시간과

메모리를 사용하기 때문에, 시간 제약이 중요한 응용에서는 부가적인 메모리를 사용하지 않고 빠른 응답을 제공할 수 있는 조인(Join) 방법이 제공되어야만 한다. 현재, 데이터세트의 구조가 기본형으로 정형화 된 구조인 경우에는 다양한 해결 방법이 제시되고 있지만, 자신내에 또 다른 데이터세트들을 반복적으로 내포할 수 있는 구조에서는 처리의 비효율성과 복잡성이 매우 크다.

본 논문에서는 이러한 문제점들을 해결하기 위해, 비정형화 된 구조를 쉽게 구조화할 수 있는 데이터 모델을 소개한다. 구체적으로, 각 데이터에 시스템 내에서 유일화 된 식별자를 갖도록 하고, 1) 이를 사용하여 통신환경에서 요구되는 복잡한 관련성과 제약들을 일관되게 표현하며, 2) 다중 데이터의 요구에 대해 데이터세트의 물리적인 재구성 없이 식별자간의 항해를 통해 빠른 처리 성능을 보장하는 방법을 기술한다.

3. 데이터세트에서 관계성 정의

데이터 모델은 시스템에서 데이터를 표현하고 운용하기 위한 방법들의 모임이다. 따라서, 이 장에서는 이에 대한 설명을 통해 시스템이 제공하는 기능적 측면에 대해 기술한다. 즉, 커서 운용을 위한 기본 데이터세트의 구성과 이들에 대해 커서를 통해 관련 데이터들을 재구성하는 과정들을 자세히 설명하도록 한다.

데이터세트는 동일한 특성(혹은 속성)을 갖는 데이터들의 모임이다. 데이터는 고유 의미를 가지며 동일한 형태의 데이터들의 집합을 의미한다. 각 데이터는 시스템에서 유일하며, 데이터의 물리적인 위치를 갖는 식별자(RID, Record Identifier)에 의해 구분된다. 데이터들의 관계성은 데이터내에 관련 있는 식별자를 속성으로 포함함으로써 성립된다. 예를 들어, (그림 2)에 나타낸 바와 같이 데이터세트 C1, C2간에 관계성 Rc1_c2이 존재한다면, 관계성이 있는 두 데이터 $oc1 \in C1, oc2 \in C2$ 는 각각 $oc2, oc1$ 의 식별자를 속성의 값으로 갖게 된다. 또한, 이러한 관계성과 관계성의 다중성(Multiplicity)을 명시하기 위해, REF, SET 및 inverse를 정의하였으며, 이를 사용하여 정의한 예는 다음과 같다.

<pre>create dataset C1 { Attrc1 Rc1_c2 REF C2, }</pre>	<pre>create dataset C2 { Rc2_c1 SET inverse C1.Rc1_c2, Attrc2 }</pre>
--	---

(그림 2) 데이터세트와 관계성의 정의

예에서, $oc1$ 은 C2내 데이터와 유일하게 하나의 관계성을 명시할 수 있으며, 반면에 $oc2$ 는 Rc2_c1이 SET으로 선언되었기 때문에 $oc1$ 이외에도 다수개의 C1내 데이터들과 관계성을 명시할 수 있다. 또한, inverse의 의미는 데이터세트

C1, C2간에 Rc1_c2와 Rc2_c1을 사용한 관계성은 항상 양방향성을 가지는 것으로 C1에서 Rc1_c2를 사용하여 C2를 참조할 수 있으며 역으로 C2에서 Rc2_c1으로 C1을 다시 참조할 수 있음을 의미하며, 보다 세부적으로는 두 데이터간의 참조 무결성 보장을 의미한다. 관계성의 명시는 해당 데이터의 식별자를 가짐으로서 성립된다. 예를 들어, C1 데이터 $oc1$ 의 Rc1_c2는 관련된 C2 데이터 $oc2$ 의 식별자를 가짐으로서 구체화된다. 여기에서 $oc1$ 과 $oc2$ 가 Rc1_c2로 구체화되었다는 것은 $oc2.Rc1_c2$ 을 통해 $oc2$ 데이터에 접근 가능하며, 역으로 $oc2.Rc2_c1$ 을 통해 $oc1$ 데이터로의 접근도 가능하다는 것을 의미한다. 이와같이 데이터세트들간에 미리 정의한 관련성은 데이터 추출을 위한 질의에서 사용될 때, 기본 데이터세트로 시작하는 경로로서 명시되고, 실제 추출은 경로에 명시한 속성이 가진 식별자를 사용함으로써 완료된다.

다음의 (그림 3)은 이를 사용한 C1과 C2의 조인 질의를 보이고 있다. 질의에서와 같이 식별자(Rc1_c2 혹은 Rc2_c1 값)를 통해 다른 데이터로의 참조는 곧 그 데이터와의 조인을 의미하는 것으로, 이와같이 식별자를 통한 다른 데이터의 참조를 위해 경로 연산자("->")를 사용한다. 경로 연산자는 자신과 관련된 다른 데이터를 지칭할 때 사용되며, 다시 지칭된 데이터내의 식별자를 통해 또 다른 데이터와 반복적으로 조인될 수 있다.

```
retrieve Attrc1, Rc1_c2 -> Attrc2
from C1

retrieve Rc2_c1 -> Attrc1, Attrc2
from C2
```

(그림 3) 관계성을 사용한 C1, C2 조인

조인이 요구될 경우, C1에서 Rc1_c2(혹은 C2의 Rc2_c1)를 사용해 다른 데이터세트인 C2(혹은 C1)로 직접 접근할 수 있어, 임시 데이터세트의 생성없이 가능하기 때문에 기존의 방식[5,6]에 비해 빠른 검색이 가능하다. 또한, 통신환경에서는 조인 후 전체 데이터세트를 추출하는 연산 보다 조건에 일치하는 소수 데이터의 추출 연산이 주로 요구된다는 점에서 조인 시 이를 위한 작은 데이터세트의 재구성은 데이터 처리 성능에 미치는 영향이 매우 크다[1]. 따라서, 본 논문에서는 C1내 데이터와 연관된 C2내의 데이터를 추출하기 위해 일련의 알고리즘들을 사용하고 그 결과 얻어진 조합을 새로운 데이터세트로 구성하는 기존의 방식과는 달리, C1내 미리 포함된 식별자로 직접 접근할 수 있는 항해 알고리즘을 통해 부가의 데이터세트 구성 없이도 조인 결과를 제시할 수 있는 방법으로 보다 향상된 성능을 제공하고자 한다. 본 논문은 모델 내에서 관계성을 표현하는 식별자를 통해 연관된 데이터들간의 조인을 위해 필요

한 항해 알고리즘과 구조에 대해 주로 기술한다.

4. 항해기반의 데이터셋 커서

커서는 주어진 질의(Query)에 만족하는 데이터들의 순차적인 접근 연산을 위해 사용하는 구조이다. 기존의 커서는 재구성된 데이터들에 대한 순차 접근을 제공하지만, 이 장에서는 식별자를 사용해, 데이터의 물리적인 재구성 없이 항해를 통해 직접 접근하여 보다 융통성 있고 효과적인 커서에 대해 기술한다.

4.1 커서의 구조

커서는 기본적으로 다음의 네 요소로 표현할 수 있다. 먼저, D는 추출될 정보를 포함하는 데이터셋이며, A는 D내의 검색의 대상이 되는 데이터의 식별자 그룹의 집합이다. 식별자 그룹은 경로에 의해 구분되어, 연산 시점에 동시에 연산의 대상이 되는 데이터들의 집합을 의미하는 것으로, 이에 대한 보다 자세한 내용은 4.2절에서 설명한다. C는 결과로 반환되기 위해 만족하여야 하는 조건들로 속성과 연산자, 값으로 구성된다. S는 조건에 만족할 경우, 추출될 속성 정보를 명세한다.

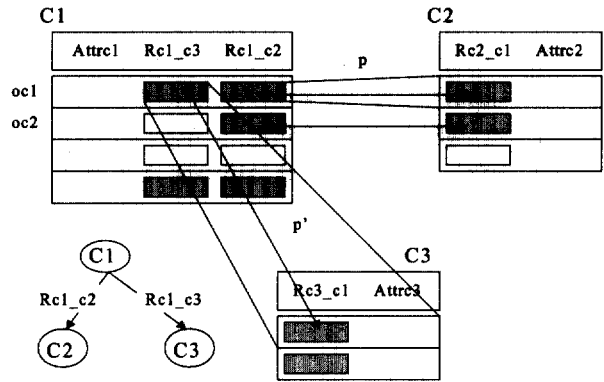
커서 Csr = (D, A, C, S)

- 여기에서, D : dataset의 집합
- A : D내의 검색 대상 데이터 식별자 그룹(= 앨리어스)들의 집합
- C : 조건식들의 집합
- S : 결과 속성들의 집합

예를 들어, (그림 2)의 정의에서 “retrieve Rc1_c2 -> Attrc2 from C1 where Attrc1 = 10”에 대해 구성되는 커서 $c \in Csr$ 는 ({C1, C2}, {{oc1}, {oc1.Rc1_c2}}, {Attrc1 = 10}, {Rc1_c2 -> Attrc2})로 표현할 수 있다. 여기에서, oc1은 C1내 데이터에 대한 식별자라고 하면, oc1.Rc1_c2는 oc1과 관련된 C2의 데이터에 대한 식별자가 된다. 따라서, oc1.Rc1_c2 -> Attrc2는 C2 데이터의 특정 속성을 지칭하게 될 것이다. 질의를 처리하기 위해서는 질의의 분석 과정을 통해 커서를 구성하고, 이의 항해를 통해 결과를 추출하게 된다.

질의에서 기본 데이터셋으로부터 REF나 SET과 같은 관계성을 통해 다른 데이터셋으로의 참조, 즉 경로가 명시된다면, 기본적으로 기본 데이터셋을 루트로 하는 트리(tree)를 형성하게 된다.

(그림 4)에서, C1에서 Rc1_c2는 C2로의 참조를, Rc1_c3는 C3로의 경로가 가능함을 알 수 있다. 따라서, C1을 루트로 하여, C2내의 속성을 참조하기 위해서는 C1의 속성인 Rc1_c2를 사용한 경로식을 통해, C3내의 속성은 Rc1_c3를 사용한 경로식을 통해 참조할 수 있다. 이와 함께, 경로식



(그림 4) 경로의 계층화

을 사용한 조건식의 명세도 가능하다. 여기에서 참조란 C1의 Rc1_c2, Rc1_c3가 각각 C2, C3내의 특정 데이터에 대한 식별자를 가지며, 이를 사용해 C2, C3내의 데이터에 직접 접근하는 방식을 통해 세 개의 데이터셋 C1, C2, C3간의 관련성 표현과 속성의 추출을 경로식이라는 단일의 체계화된 표현으로 가능하다. 다음절에서는 질의에서 데이터셋과 앨리어스를 추출하는 과정을 상세히 설명한다.

4.2 앨리어스(Alias)의 추출

서로 독립적인 데이터셋을 식별자를 통해 서로 연관시키고 연산하기 위해서는 자신 데이터셋을 구분하는 경로를 유일하게 구분하여야 하고, 이를 자신의 식별자 그룹내의 유효한 식별자로 대응하여야 한다. 이것은 C1에서 연산의 대상이 oc1에서 oc2로 변경됨에 따라 C2, C3의 대상 데이터도 함께 변경되어야 하기 때문이다. 즉, $pc_1(Rc1_c2) \rightarrow C2$ 로 C1의 경로 p에 특정 데이터의 속성값을 대응하여 실제 C2의 식별자를 추출하는 과정으로, Rc1_c2이 새로운 Rc1_c2'로 변경됨은 곧 C2의 연산 대상이 $pc_1(Rc1_c2')$ 로 함께 변경되어야 한다는 것이다. 이를 위해서는 데이터셋들간에 존재하는 각 경로들을 분석하여, 대응하는 데이터셋내의 식별자 그룹을 생성하여야 한다. 본 논문에서는 이를 앨리어스라고 지칭하며, 다음과 같이 정의된다.

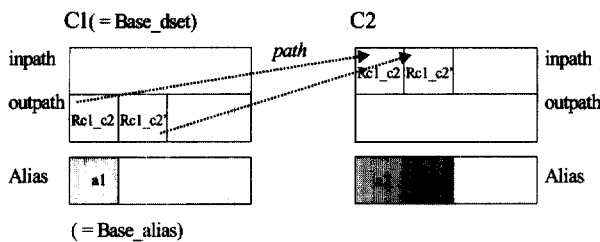
앨리어스 A = (D, P, F)

- 여기에서, D : 현재 데이터셋을 참조하는 축의 데이터셋
- P : 이전 앨리어스와 속성으로 현재 D를 구분하는 경로
- F : REF/SET 속성

앨리어스는 데이터셋을 참조하는 경로들에 의해 구분된다. 즉, C2는 C1의 모든 데이터들로부터 오직 Rc1_c2를 통한 경로가 한개 존재하므로, 한개의 앨리어스가 존재하고 있음을 알 수 있다. 좀 더 일반화된 예로 하나의 데이터셋에서 다수개의 앨리어스가 존재할 수 있는 상황을 가정해 보자. 예로, C3에 대해, Rc1_c3에 의한 식별자와 Rc2_c3에 의해 지칭되는 식별자는 모두 C1내의 데이터들을 각각

지칭하지만 C3를 지칭하는 문맥에 따라 구분되어야 한다 (혹은 자기 참조형 관계도 가능하므로 Rc3_c3와 같은 형태를 갖는 관계성의 명시도 가능). 따라서, 데이터세트내에 다수개의 식별자를 구분하기 위해 경로상에 자신을 지칭하는 이전 데이터세트와 속성, 그리고 경로를 조합하여 구성한 (데이터세트, 속성, 경로)의 세 요소에 의해 유일하게 구분되도록 정의되었다. 단, 기본 데이터세트는 기본적으로 디폴트 경로를 갖도록 정의된다. 예를 들어, (그림 4)에서 C1은 디폴트 경로만을 갖고, C3의 경우에는 C1-C3간의 경로로 (C1, Rc1_c3,0)을 갖는다(여기에서 0의 기본 데이터세트인 C1의 디폴트 경로 ID임). 만일, C1내 자기 참조형의 구조가 있었다면, 디폴트 경로외에 (C1, Rc1_c1,0)을 추가로 가지게 될 것이고, 다시 (C1, Rc1_c1,0)을 통해 C3로의 관계성이 설정되었다면, C3는 (C1, Rc1_c3, (C1,Rc1_c1,0))을 추가로 갖게 될 것이다. 이러한 경로는 계층화된 데이터세트를 식별자를 통해 항해하는 과정에서 현재의 문맥에 맞는 대상 엘리어스(= 데이터세트의 식별자)를 추출하는데 사용된다.

(그림 4)를 엘리어스 측면에서 좀 더 구체화한 (그림 5)를 살펴보자. 그림에서, 기본 데이터세트 C1(=BASE_dset)은 기본 엘리어스 a1(BASE_alias)를 갖도록 초기화된다. 데이터세트 C2의 엘리어스 a2는 (BASE_alias, a1, Rc1_c2)에 의해 생성된다. 만일, C1에서 C2로의 관계성 Rc1_c2'가 정의되어 있다면, 이 경로는 (Base_alias, a1, Rc1_c2')로 새로운 엘리어스 a3가 생성된다.



(그림 5) 엘리어스의 추출

다음의 알고리즘은 기본 데이터세트와 경로식이 주어졌을 때, 경로식을 분석하여 엘리어스를 생성하기 위한 알고리즘이다. 이때, 경로식은 “s->n->...” 형식이며, s, n은 각각 데이터세트를 지칭하는 REF 혹은 SET 타입을 갖는 속성이름이다.

```

algorithm make_cursor ( name : 데이터세트 이름, p : 경로식 )
dset : 데이터세트 정보
alias : 작업 엘리어스
begin
  dset = create_dataset_info ( name )
  alias = dset.active_alias ( )
  while ( 모든 속성 attr in p )
    path = dset.search_out_path ( alias, attr )

```

```

if ( search가 성공 )
  if ( attr is leaf )
    cursor.add_attr_info ( dset, alias, attr )
  end if
else
  next_dset = dset.get_next_dataset ( attr )
  if ( next_dset이 커서내에 존재하지 않으면 )
    create_dset_info ( next_dset.get_name() );
  end if
  next_dset.add_inpath ( path )
  end if
  dset = next_dset ;
  path' = dset.convert ( path )
  if ( not dset.exist_path ( path' ) )
    dset.create_alias ( path' )
  end if
  alias = dset.getAlias ( path' )
end while
end

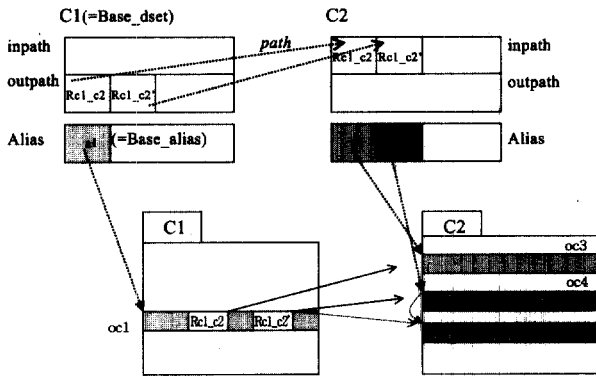
```

알고리즘에서 현재 데이터세트(dset)에 대해, 현재 속성 attr으로의 경로(path')가 존재하지 않는 경로일 경우 새로운 엘리어스를 생성함을 보이고 있다.

4.3 엘리어스를 사용한 항해

구성된 엘리어스는 데이터세트를 항해하기 위해 필요하다. 항해의 방식은 조건식의 평가를 통해 이루어진다. 이를 위해 조건식내 명시된 속성을 포함하는 엘리어스가 지칭하는 데이터와 식을 평가하게 된다. 만일, 모든 식의 평가가 성공적으로 이루어졌다면, 모든 엘리어스가 지칭하는 데이터는 질의의 결과 대상이 된다. 그렇지 않다면, 다음 데이터 식별자를 엘리어스로 대체한 후, 식의 평가를 반복하게 된다. 이러한 과정을 (그림 6)을 통해 살펴보자. 그림에서는 C2내의 두 개의 엘리어스가 존재하고 있다. 만일, 엘리어스 a3는 C1과 SET 관련성으로 정의되었고, 현재 a3가 지칭하는 데이터(식별자가 oc3인)에 대한 조건식의 평가가 true가 아니라면, a3는 가능한 다음 데이터 식별자 oc4로 대체되어 식을 평가하게 된다. 식의 평가 후 결과가 다시 true가 아니라면, a3에 대체할 다음 식별자를 추출하게 되며, 이때, 다음 식별자의 추출이 가능하지 않게 된다면, a3의 경로상 상위의 엘리어스인 C1내의 엘리어스 a1에 대한 대체(알고리즘에서 touch_cursor)를 요구하게 된다. 현재, a1이 REF이거나, SET이지만 대체할 식별자를 갖지 못한다면, a1자체의 대체가 필요하게 된다. 즉 C1에서 oc1이외의 다음 데이터가 a1에 의해 새로이 지칭되며, 이에 따라 경로상 a1의 하위 엘리어스인 a2, a3가 a1에 의해 초기화된 후 앞의 식의 평가를 다시 시작하게 된다.

이와 같은 반복 처리를 통해 C1내의 모든 데이터를 평가한 결과는 사용자에게 반환될 것이다. 다음의 알고리즘은 4.2절에서 추출된 데이터세트와 엘리어스를 바탕으로 주어진 조건식을 평가하는 과정을 보이고 있다.



(그림 6) 데이터세트의 항해

```

algorithm touch_cursor ( dset : 데이터세트, alias : 엘리어스,
                        attrs : 속성들 )
begin
  re-evaluate :
  while ( attrs내의 각 속성등 attr에 대해 )
    dset = attr.get_dset ()
    alias = attr.get_alias ()

    data = dset.get_data(alias)
    result = data.evaluate_expr ( attr, attr.get_operator(),
                                attr.get_value() )
    if ( result = true )
      continue
    end if
    status = touch ( dset, alias )
    if ( status = RequiredReEvaluate )
      goto re-evaluate
    return status
  end if
end while
end
  
```

엘리어스 대체의 기본 방식은 자신 엘리어스가 조건식을 만족하지 못할 때, 경로상의 상위 엘리어스의 대체를 요구하는 방식으로 이루어진다. 다음의 알고리즘은 자신의 대체가 가능한지 검사한 후, 가능하지 않을 경우 상위 엘리어스의 대체를 요구하기 위한 절차를 보이고 있다.

```

algorithm touch ( dset : 데이터세트, alias : 엘리어스 )
begin
  if ( dset = BASE_dset and alias = BASE_alias ) return
    RequiredNext ;
  pre_dset = dset.find_pre_dset ( alias )
  path = dset.find_path ( alias )
  if ( path.get_type() = SET and not path.all_traverse() )
    oid = path.getNextSetElement () ;
    dset.set_identifier ( alias, oid )
    return RequiredReEvaluate ;
  end if
  return touch ( pre_dset, dset.get_prealias() )
end
  
```

알고리즘에서 RequiredReEvaluate 상태는 기본 엘리어스(예에서는 C1내의 a1)의 변경없이 하위의 엘리어스(a2 혹은

a3)의 대체 후, 질의에 주어진 조건식을 재평가해야 함을 의미한다. 만일, 기본 엘리어스가 대체되어야 하는 경우에는 RequiredNext 상태를 반환한다.

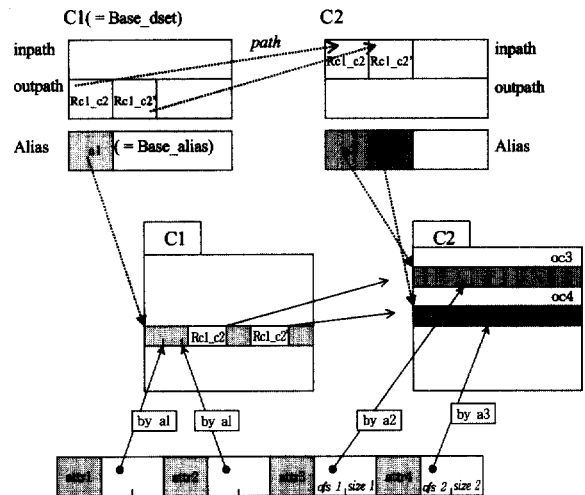
4.4 결과 추출

질의의 모든 조건식이 성공적으로 평가되었을 경우, 각 데이터세트내의 현재 엘리어스들이 지칭하는 데이터들은 사용자에게 반환되어야 한다. 질의는 데이터의 특정 속성이 명시되었다면, 해당 속성들만이 추출되어 반환되어야 한다. 이를 위해, 커서는 결과 속성의 리스트를 가지며, 결과 속성들은 다음과 같은 세 요소로 정의된다.

$$\text{결과 속성 RS} = (A, P, S)$$

- 여기에서, A : 자기 자신 속성이 속할 데이터세트와 엘리어스
- P : 엘리어스로 지칭된 데이터의 시작에서 결과값의 위치 값
- S : 실제 결과값의 크기

예를 들어, (그림 7)에서 attr3에 대해, 실제 값은 데이터 세트 C2내 엘리어스 a2가 지칭하는 데이터(oc3)의 ofs1 만큼의 오프셋(offset) 위치와 size2만큼의 크기로 명시하고 있다. 따라서, attr3는 (a2,ofs1,size1)∈RS로 표현할 수 있으며, attr4는 동일 데이터세트에서 엘리어스 a3에 의해 지칭되는 데이터(oc4)내 속성을 지칭하고 있어 (a3,ofs2,size2)∈RS로 표현할 수 있다. 커서는 이러한 결과 속성들의 리스트에서 각 결과 속성에 대해 해당 엘리어스가 지칭하는 데이터의 오프셋 위치에서 크기 만큼의 값을 연속적으로 추출함으로써(다음 알고리즘에서 get_data()), 다중 데이터세트에 대한 고려 없이 반환될 결과값의 구성이 가능하다.



(그림 7) 데이터의 추출

다음은 이를 위한 알고리즘으로, 알고리즘에서 attrs는 결과 속성 리스트이며, results는 이들 속성의 실제값을 저장하기 위한 버퍼이다. 결과 버퍼에 추가될때, 각 속성은 형(type), 크기(size) 정보등을 추가로 갖도록 한다.

```

algorithm get_info ( attrs : 속성들 )
  results : 결과값들
  begin
    while ( attrs내의 각 속성동 attr에 대해 )
      results.add_value( attr.get_data ( ) )
    end while
    return results ;
  end
    
```

이와 같이 앨리어스의 항해를 통해 데이터 추출 기능은 질의내 모든 데이터세트를 물리적으로 재구성하는 기존의 방식 [1, 5, 6]들에 비해, 물리적인 재구성없이 결과의 추출이 가능하기 때문에 보다 빠른 성능을 제공하며, 다수 결과 추출에 대해서도 균일한 접근 성능을 제공하는 특징을 가지고 있다.

5. 결론 및 향후 연구 과제

다중 데이터세트에 대한 조인시, 기존의 방식들은 비록 메모리내 연산이기는 하지만 다량의 메모리 복사를 사용해 새로운 데이터세트(혹은 인덱스)를 재구성하는 방식을 사용하고 있다. 그 결과, 현재의 하드웨어 구조상, 최소화하여야 하는 메모리 병목 현상[8]을 피하지 못하고 있으며, 또한, 이 방법은 비록 주 메모리간 연산이지만 데이터세트의 크기나 상태에 따라 많은 준비시간이 소요될 수 있다[9]. 그러나, 통신 시스템은 처리 시간에 대한 엄격한 제약을 갖고 있어, 특정 연산이 다수의 시간을 사용하는 경우 해당 이벤트의 완료에 불확실성을 주기 때문에 적용이 어렵게 될 수 있다. 따라서, 본 논문에서는 메모리간의 물리적인 복사 연산을 최소화하면서 안정되게 조인할 수 있는 방법을 제안하였다. 주요 내용으로는 데이터 삽입시 다중 데이터세트들간의 관계성을 미리 설정하고, 이들에 대한 추출 요구시 이 관계성에 따라 물리적인 메모리 복사 없이 메모리 주소를 추적하면서 결과를 얻을 수 있도록 하는 체계이다.

구체적으로, 본 논문에서는 비정형화되고 복잡한 구조를 갖는 데이터세트에 대해, 경로식을 포함하는 데이터 모델을 통해 정형화 된 접근체계를 인터페이스를 제공하고, 실제 이를 구현하기 위한 방법으로 데이터의 식별자를 사용한 항해 기반 커서에 대해 기술하였다. 항해 기반 커서를 운영하기 위해, 질의에서 커서를 생성하는 과정으로 경로를 분석하여 데이터세트 그룹과 앨리어스의 추출하는 방법, 추출된 앨리어스를 기반으로 조건식의 평가를 통해 식별자를 대체하는 방법, 모든 조건식을 만족하였을때 데이터내의 세부 속성 값을 추출하는 방법을 알고리즘과 함께 설명하였다.

본 논문에서 제시한 방법을 사용함으로써, 첫째, 통신환경에서 비정형적인 구조를 쉽게 표현하고 데이터 접근에 용이한 체계를 제시하였으며, 둘째, 다중 데이터세트의 조인시 발생할 수 있는 이벤트의 시간제한을 만족시킬 수 있으며, 셋째, 이를 사용한 응용 프로그램의 복잡성을 감소시킬 수 있는 효과를 얻을 수 있다. 향후, 연구과제로는 현재 고안된

데이터 모델내 경로식을 사용한 조건식의 평가를 최적화하기 위한 방법이 있다. 기본적으로 질의내 경로는 단일 루트의 트리로 구성되기 때문에, 경로의 다중성과 방향성에 대한 고려, 평가의 순서에 따라 많은 비교를 줄일 수 있으리라 사료된다.

참 고 문 헌

- [1] Y. I. Yoon, et. al., "Scalable Distributed Real-time Database Management for Switching System," Int'l Switching Symposium(ISS'97), Toronto, Canada, pp.539-545, September, 1997.
- [2] M. Stonebreaker and P. Brown, "Object-Relational DBMSs : Tracking the Next Great Wave," Morgan Kaufmann Publishers, pp.320, 1998.
- [3] W. Wolfe, J. Prichard, L. Dipippo, and J. Black, "Real-Time Database Systems : The RTSORAC Real-Time Object-Oriented Database Prototype," Kluwer Academic Publishers, pp.279-303, 1997.
- [4] S. K. Kim et. al., "A Design and Implementation of Object-oriented Compiler for the Telecommunications System Software," Int'l Technical Conf. Circuit/Systems, Computers, and Communications(ITC-CSCC'98), Sorak Hanwha Resort, Sokcho, Korea, July, 1998.
- [5] Polyhedra Plc., 'Bloor Research Overview of Polyhedra,' White Paper, [http : //www.polyhedra.com](http://www.polyhedra.com).
- [6] Raima Co., 'Raima Database Manager++ In Real-time and Embedded Systems,' White Paper, [http : //www.mbrane.com /rdm.html](http://www.mbrane.com /rdm.html).
- [7] C. T. Yu and W. Meng, "Principles of Database Query Processing for Advanced Applications," Morgan Kaufmann Publishers, pp.512, 1997.
- [8] Peter Boncz, Stefan Manegold, Martin Kersten, "Data Architecture Optimized for the New Bottleneck : Memory Access," In Proceeding of VLDB Conference, 1999.
- [9] H. Garcia-Molina and K. Salem, "Main Memory Database Systems : An Overview," IEEE Transaction on Knowledge and Data Engineering, Vol.4, No.6, pp.509-516, Dec. 1992.



배 명 남

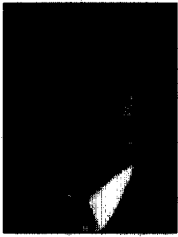
e-mail : mnbae@etri.re.kr

1991년 전북대학교 전산통계학과 졸업(학사)

1993년 전북대학교 대학원 전자계산학과 (석사)

1998년 전북대학교 대학원 전자계산학과 (박사)

1998년~현재 한국전자통신연구원 네트워크 S/W 플랫폼팀 선임 연구원
 관심분야 : 실시간시스템, 객체지향 개발 방법론, UML, CASE



최 완

e-mail : wchoi@etri.re.kr

1981년 경북대학교 전자공학과(전자계산학 전공) 졸업(학사)

1983년 한국과학기술원 전산학과(석사)

1988년 정보처리기술사(전자계산응용)자격 취득

1983년~1985년 한국과학기술원 전산학과 연구조교

1985년~현재 한국전자통신연구원 네트워크 S/W 플랫폼팀장

관심분야 : 실시간 소프트웨어, DBMS, OS, Compiler



이 동 준

e-mail : ldch@sunny.howon.ac.kr

연세대학교 컴퓨터과학과 공학박사

1989년~현재 호원대학교 컴퓨터학부

정교수

JPDC, JHSN, ETT, Performance

Evaluation 논문지심사위원

관심분야 : 무선통신(IMT-2000, Wireless Ip, Wireless ATM)의

위치관리(DB) 기법과성능분석, 무선통신보안