생물 정보원에 대한 통합 접근을 위한 랩퍼 모델

박 은 경[†] · 강 동 완[†] · 정 채 영[†] · 배 종 민[††]

요 약

이질의 생물 정보원을 통합하기 위해서는 각 정보원의 이질적인 내용을 숨기고 하나의 관점으로 표현하는 뷰를 정의해야 한다. 본 논문은 생물 정보원 통합 미들웨어를 설계함에 있어서 XML 기반의 뷰 정의모델을 제시하고 그 동작 원리를 보인다. 제시된 모델은 통합시스템 구축을 위한 융통성을 증대시키고, 보다 추상화된 수준에서 통합 질의를 수행할 수 있도록 하기 위하여 사용자 정의 XML 뷰를 지원한다. 제시된 뷰 정의 모델을 기반으로 관계형 데이터베이스와 웹 자원에 대한 랩핑 모델을 제시하고, 아울러 응용 프로그램에 대해서도 같은 모델을 사용한 랩핑 결과를 제시한다.

A Wrapper Model for Integrated Access to Biological Information Sources

Eun-Koung Park[†] · Dong-Wan Kang[†] · Chai-Young Jung[†] · Jong-Min Bae[††]

ABSTRACT

In order to integrate heterogeneous biological information sources, it is necessary to define the view that represents unified viewpoint for the multiple sources by hiding heterogeneity of the data. We present an XML-based view definition model and show its operating principles in designing the middleware system to integrate biological information sources. This model supports the user-defined XML view to increase flexibility in constructing the integration system and execute integrated queries on higher level abstraction. Based on the view-definition model, we present a wrapping model for relational database systems and web resources as well as an application program.

키워드 : 생물 정보원 통합(Integration of Biological Information Sources), XML, 뷰(View), 랩퍼(Wrapper), 미디에이터(Mediator), BLAST, Entrez

1. 서 론

생물 정보원의 통합은 연구자들이 원격 혹은 지역적으로 분산되어 있는 여러 정보원들에 개별적으로 접근해야 하는 시간소비 문제와, 생물 정보원에 접근하기 위한 인터페이스를 모두 숙지해야 하는 어려움을 덜어준다[1, 2]. 생물학적 발견은 다른 분야와는 달리 여러 생물정보원들의 결과를 연합해서 최종 결과를 얻는 경우가 많다. 이전에는 이 작업을 수작업으로 진행하여도 큰 어려움이 없었으나, 생물학적 실험 환경의 급격한 발전 덕택에 수많은 데이터가 생성되고 인터넷의 활발한 보급으로 인해서 생물 정보원들을 수작업으로 연합하여 결과를 검색하는 일은 많은 노력을 필요로 한다. 이에 따라 하나의 사용자 질의를 통하여 질의 수행에 필요한 데이터들이 저장되어 있는 여러 정보원에 접근하여 결과를 검색하고 조작하는 과정을 자동적으로 수행할 수 있는 생물 정보원의 통합에 대한 필요성이 대두되었다.

생물 정보원 통합을 어렵게 하는 요인은 정보원들의 이질적인 성격 때문이다. 우선, 데이터를 저장하고 관리하는 모델이 다양하다. GenBank[3]와 같이 데이터를 파일 시스템을 기반으로 한 텍스트 기반의 정보원, 혹은 관계형, 객체형, 객체 관계형 모델 기반의 정보원, 웹 기반의 정보원, 그리고 반 구조적 데이터 모델 기반의 정보원도 있다. 또한, 생물 정보원들에 접근하기 위한 질의 능력도 다양하다. 파일 시스템일 경우 인덱스를 이용하여 해당 데이터를 검색할 수 있고, SQL이나 OQL과 같은 질의어를 이용하여 데이터를 검색할 수도 있으며, 웹 자원일 경우 링크 기반의 인터페이스를 제공하는 정보원도 있다.

생물 정보원 통합 방법은 다음과 같이 크게 세 개의 부

※ 본 연구는 한국과학재단 특정기초연구(No. R01-2001-000-00151-0) 지원으로 수행되었음.
† 준 회 원 : 경상대학교 대학원 컴퓨터과학과
†† 종신회원 : 경상대학교 컴퓨터과학과/컴퓨터 정보통신연구소 교수
논문접수 : 2004년 2월 16일, 심사완료 : 2004년 4월 13일

류가 있다. 첫째, SRS[4]와 같은 링크 기반의 통합 방법이 있다. SwissProt[5]이나 GenBank와 같이 구조적인 레코드를 포함하고 있는 플랫 파일을 파싱하여 데이터를 저장하고, 인덱스 기반의 질의를 수행한다. 조인의 기능은 정보원간에 서로 유지하고 있는 교차참조(cross-reference)를 통해 수행된다. SRS는 웹 상에서 이미 정해져 있는 질의 플랜을 바탕으로 질의가 수행되기 때문에 정보원에 대해서 제한된 접근만 허용하는 제약점이 따른다. 둘째, Informax사의 Genomax 시스템과 같은 데이터 웨어하우스 기술을 기반으로 한 통합 방법이 있다[6]. 서열 데이터, 유전자 발현데이터, 단백질 3차원 구조정보 등 다수의 정보를 정보원에서 물리적으로 추출하여 하나의 지역 저장소인 웨어하우스에 저장한다. 이는 네트워크와 관련된 병목 현상의 문제점을 감소시켜 주며, 정보들이 단일 형식으로 저장되어 있기 때문에 질의 처리와 분석, 최적화를 처리하기가 비교적 쉽다. 데이터 웨어하우스의 특징상, 원 저장소와는 별도로 개별적인 지역 저장소를 유지하고 있으므로 데이터가 증가할수록 비용도 비례할 것이며, 원 저장소에 저장되어 있는 데이터의 내용이 변경될 때마다 갱신 문제도 발생한다. 셋째, IBM에서 개발한 DiscoveryLink[7]와 같이 미디에이터(Mediator) 기반의 통합 방법이 있다. 미디에이터는 각 정보원에 대한 전역 스키마를 유지하고, 사용자 질의를 분해하여 랩퍼(Wrapper)에게 전달하며, 랩퍼를 통해 얻은 결과를 통합하여 사용자에게 전달하는 미들웨어 시스템이다.

미디에이터 기반의 통합 모델에서 랩퍼는 각 정보원의 이질적인 내용을 숨기고 하나의 관점으로 표현하는 뷰를 정의해야 한다. 이를 위해 정보원을 하나의 관점으로 표현할 수 있는 데이터 모델이 필요하다. 그 대표적인 데이터 모델로 OPM(Object Protocol Model) 모델[8]과 XML 기반 모델 등이 있다. 이질적인 데이터를 쉽게 표현할 수 있는 XML을 기반으로 뷰를 정의할 경우, 사용자는 모든 정보원을 가상적인 XML 문서로 구성되어 있는 XML 정보원으로 간주하게 되어, 이후 XML 스키마와 XML 질의어를 사용해서 통합 질의 처리가 가능하다.

생물 정보는 그 특성상 사용자의 요구가 매우 다양하게 나타나기 때문에, 다양한 요구가 쉽게 수용될 수 있도록 통합 미들웨어를 설계해야 한다. 예를 들어 사용자는 유전자 정보, 단백질 정보, 서열 유사성비교, 문헌 정보를 동시에 필요로 할 수 있다. 이때, 각 정보원에서 제공하는 정보의 일부를 바탕으로 다음 정보원에 접근하는 경우, 사용자에게 정보원에서 제공하는 모든 정보를 제시할 필요가 없다. 이와 같이 통합 시스템 설계의 융통성을 증대시키기 위해서는 정보원에서 제공하는 데이터의 일부만을 대상으로 질의를 할 수 있어야 한다. 이를 위하여 정보원에 대한 기본적인 뷰를 바탕으로 사용자 정의 뷰를 지원할 필요가 있다. 통합 미들웨어 시스템 사용자는 사용자 정의 뷰를 통해서 정보원을 추상화시킬 수 있으며, 이를 통해 동일 정보원에 대하여 다양한 뷰를 가질 수 있어서 보다 상위 수준에서의 통합 질의 수행이 가능하다.

한편, 데이터를 관리하는 정보원뿐 만 아니라 서열유사도 검색 도구 BLAST(Basic Local Alignment Search Tool)[9]와 같은 응용프로그램도 함께 통합되어야 그 유용성이 증대될 것이다. 응용 프로그램 통합을 위하여 이에 대해서도 랩핑이 필요한데, 그 모델이 정보원의 랩핑 모델과 동일해야 그 융통성이 증대된다.

본 논문에서는 생물 정보원 통합 미들웨어를 설계함에 있어서, XML 기반의 뷰 정의모델을 제시한다. 제시된 모델은 사용자 정의 XML 뷰를 지원한다. 제시된 뷰 정의 모델을 기반으로, 관계형 데이터베이스와 웹 자원에 대한 XML 뷰 기반의 랩핑 모델을 제시하며, 특히 응용 프로그램에 대해서도 같은 모델을 사용하여 랩핑한 결과를 제시한다. 그리고 정의된 뷰에 대한 질의 처리 방법을 보인다.

논문의 구성은 다음과 같다. 2장에서는 뷰 기반의 통합 모델을 사용하는 기존의 연구를 살펴보고, 3장에서는 XML 뷰 기반의 통합 시스템 구조를 설명한다. 4장에서는 관계형 데이터베이스 기반의 지역 생물정보원과 응용 프로그램, 그리고 웹 자원에 대한 XML 뷰 기반의 랩핑 모델을 제시하고, 5장에서 결론 및 향후 연구과제에 대해서 논한다.

2. 관련 연구

생물 정보원들 사이의 이질성을 극복하기 위해서는 각 정보원에 대한 가상의 뷰를 정의함으로서 공통의 데이터 모델로 추상화하여 사용자에게 각 정보원의 이질성을 숨기고 동일한 관점으로 정보원에 접근할 수 있도록 한다. 여기서는 정보원과 응용프로그램에 대한 기존의 뷰 정의 모델을 살펴본다.

2.1 OPM 기반의 뷰 정의 모델

OPM 멀티데이터베이스 시스템[8, 10]에서는 통합하고자 하는 생물정보원들을 OPM(Object-Protocol Model) 기반으로 모델링 한다. OPM은 크게 두 개의 클래스로 구분이 되어 각 정보원을 표현하는데, 하나는 객체 클래스이고 다른 하나는 프로토콜 클래스이다. 각 정보원은 하나의 객체로 표현되며, 객체 id와 애트리뷰트들을 이용하여 객체 클래스를 기술한다. 프로토콜 클래스는 생물학적 실험을 모델링하기 위한 클래스이다. 객체 클래스와 프로토콜 클래스는 BNF 기반의 OPM schema로 표현되는데. 이것이 OPM 멀티데이터베이스 시스템의 전역 뷰가 된다. 그런데 객체 클래스는 다른 객체 클래스로 파생될 수 있다. 즉, 여러 개의

객체 클래스를 모두 합쳐서 하나의 객체 클래스로 구성할 수 있으며, 클래스 애트리뷰트의 일부를 추출하여 객체 클래스를 구성할 수 있다. 클래스에 정의되어 있는 애트리뷰트를 이용하여 수식계산이나 집단 함수를 적용하여 새로운 애트리뷰트를 정의할 수 있다. 그러나, OPM에서는 멀티미디어 데이터 타입이나 특정 응용 프로그램에 대해서 모델링하기 위해 ASDT(Application Specific Data Type)[11]라는 특수한 OPM 클래스를 제공한다. ASDT 클래스에는 통합하고자 하는 응용 프로그램의 특정 기능을 수행할 수 있는 메소드가 정의되어 있다.

2.2 ODL 기반의 뷰 정의 모델

K2[12]는 Kleisli 시스템을 기반으로 한 미디에이터 기반의 시스템이다. K2는 통합하고자 하는 각 생물 정보원들을 ODL(Object Definition Language)을 이용하여 표현하며, ODL로 기술된 각 클래스에서는 원시 데이터 타입에서부터 집합 개념의 복잡한 데이터 타입을 지원하는 애트리뷰트와 다른 클래스와의 관계를 정의하고 있다. K2에서는 정보원의 내용과 ODL로 기술된 클래스 사이의 사상을 위해 K2MDL (K2 Mediator Definition Language)이라는 언어를 사용한다.

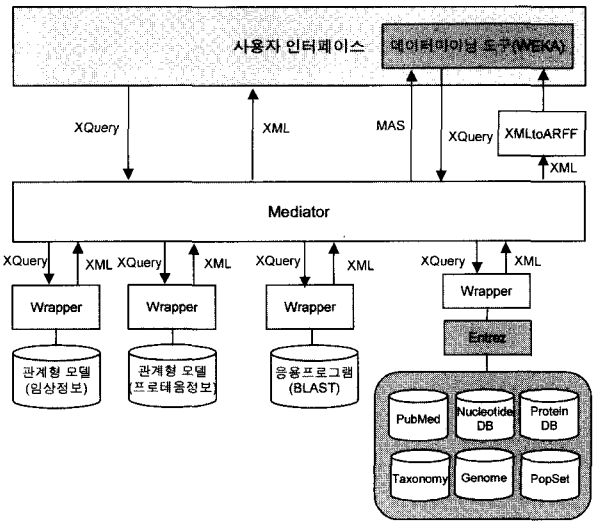
2.3 Search View 기반의 뷰 정의 모델

Search View를 기반으로 웹 기반의 생물 정보원을 객체로 모델링한 연구에서는 하나의 생물 정보원은 하나의 클래스로 표현되며, 각 클래스에는 해당 정보원에 대한 URL 기반의 검색 능력을 세 가지 유형의 애트리뷰트를 통해서 표현한다[13]. 예를 들어 해당 id를 만족하는 완전한 하나의 문서를 반환하는 URL을 Key 애트리뷰트를, 특정 키워드를 만족하는 문서들의 리스트를 반환하는 URL을 Search 애트리뷰트로, 그리고, 특정 응용 프로그램을 수행할 경우, 해당 응용 프로그램에 대한 URL과 수행 시 필요한 여러 인자들에 대해서 기술한 Abstract 애트리뷰트로 표현한다. 각 클래스에는 해당 정보원에 대한 모든 질의 능력을 세 가지 유형의 애트리뷰트를 통해 표현한다. 그러나 이 Search View 기반의 모델링은 애트리뷰트에 명시되어 있는 고정된 URL 형태의 검색만을 제공하며, 검색되어진 문서를 분석하기 위하여 각 정보원마다 별도의 파서를 구현해야 한다.

이와 같이 기존의 연구에서는 객체 모델을 사용해서 뷰를 정의하거나, 혹은 별도의 뷰 정의 언어를 사용하는데 반하여, 본 논문에서는 XML 기반의 뷰를 정의함으로써, 모든 생물정보원을 XML 정보원으로 간주하여, XML 스키마와 XML 질의어 등 XML에서 제공하는 표준 도구를 사용할 수 있도록 한다. 그리고 사용자 정의 XML 뷰를 지원하여 사용자의 목적에 따라서 스키마를 재구성할 수 있어 질의 설계에 큰 융통성이 있다. 게다가 이 모델은 응용프로그램에 대한 랩핑에도 그대로 적용되기 때문에, 응용 프로그램 통합도 쉽게 구현할 수 있다.

3. 시스템 구조

본 논문에서 제시하는 통합 시스템은 실험실에서 생성된 자료를 저장하고 있는 관계형 혹은 객체 관계형 데이터베이스, GenBank, PubMed 등과 같이 Entrez[14]에 의해서 접근 가능한 데이터베이스들, 응용 프로그램 도구인 BLAST, 그리고 데이터마이닝 도구인 WEKA(Waikato Environment for Knowledge Analysis)[15]를 XML 기반으로 통합한다. (그림 1)은 통합 시스템의 개략적인 구조이다.

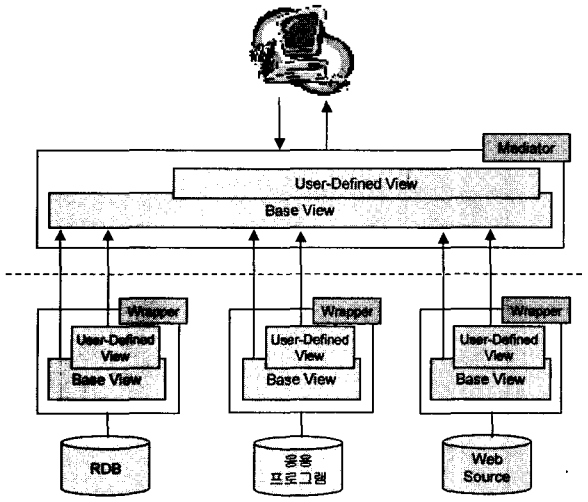


(그림 1) 생물학적 자원의 통합시스템 구조

3.1 XML 뷰 정의

랩퍼는 스키마 관리부, 질의합성과 변환을 담당하는 질의처리부, 그리고 질의결과를 XML문서로 변환하는 태거부로 구성된다. 이 중에서 스키마 관리부에서는 정보원의 전체 내용을 가상의 XML 문서 형태로 표현한 XML 뷰를 생성하는데 이를 랩퍼의 기본 XML 뷰(WBV : Wrapper Base XML View)라고 한다. 기본 XML 뷰는 실체화되지 않은 가상의 XML 문서이다. 따라서 기본 XML 뷰에 대응되는 스키마 정보가 필요하다. 이것을 랩퍼의 기본 XML 스키마(WBS : Wrapper Base XML Schema)라고 하며, 가상의 XML 문서에 대한 구조 정보와 데이터 타입 정보를 담고 있다. 스키마 표현은 W3C 표준안인 'XML Schema 1.0'[16, 17, 18]을 이용한다. 사용자 정의 XML 뷰 정의어로는 W3C의 'XQuery 1.0'[19]를 사용한다. 사용자 정의 뷰에 의해서 재구성된 가상의 XML 문서에 대한 스키마를 랩퍼의 응용 XML 스키마(WAS)라고 한다. 스키마는 미디에이터에게 전달해서 전역 스키마를 구성하도록 한다.

미디에이터는 각 랩퍼에서 제공된 기본 XML 뷰(WBV)를 통합한 전역 뷰를 구성한다. 이것을 미디에이터의 기본 XML 뷰(MBV)라고 한다. 미디에이터의 기본 XML 뷰(MBV)는 전체 정보원에 대한 뷰이다. 미디에이터의 기본 XML 뷰는 각 랩퍼에서 제공한 기본 XML 뷰(WBV)를 통합해서 구성된다. 또한, 랩퍼에서 정의된 사용자 정의 XML 뷰도 미디에이터의 기본 XML 뷰(MBV)에 포함된다. 미디에이터에서도 사용자 정의 XML 뷰를 지원하기 때문에, 사용자의 요구에 따라서 뷰의 재구성이 가능하다. (그림 2)는 랩퍼와 미디에이터에서의 뷰 구성 관계를 보인 것이다.



(그림 2) 랩퍼와 미디에이터간의 뷰 구성 관계

3.2 질의처리

기본 XML 뷰와 사용자 정의 XML 뷰, 그리고 사용자 질의는 모두 XQuery로 표현되며, 내부적으로 동일한 데이터 모델인 트리로 표현한다. 랩퍼 질의 처리부는 미디에이터에서 전달받은 사용자 정의 XML 뷰나 사용자 질의를 뷰 트리로 구성한다. 이 과정에서 합성이 이루어져서 기본 XML 뷰에 대한 질의로 재구성된다[20]. 그 결과로써 각 정보원에서 수행 가능한 질의어로 변환하고, 질의 결과를 XML 문서로 변환하기 위한 XML 탬플릿을 생성한다. 랩퍼가 각 정보원에서 처리된 수행 결과를 전달받으면 XML 템플릿을 이용하여 XML 문서로 변환하여 미디에이터에게 전달한다. 스키마 관리부에서는 합성된 뷰 트리의 구조 정보와 시스템 카탈로그나 메타 데이터를 통해 각 엘리먼트의 데이터 타입을 제공받아서 기본 XML 스키마(WBS) 혹은 응용 XML 스키마(WAS)를 생성한다.

미디에이터에서도 랩퍼와 마찬가지로 사용자 정의 뷰를 지원하기 때문에, 합성을 통하여 기본 XML 뷰(MBV)에 대한 질의로 재구성해야 한다. 따라서 그 처리 과정은 랩퍼의 경우와 동일하게 뷰 트리 기반으로 이루어진다. 합성된 사용자 질의에 대하여 정보원에 관한 메타 데이터를 이용하여 각 랩퍼로 보내질 하위질의를 인식한다. 이 메타 데이터는 미디에이터의 기본 XML 뷰(MBV)와 랩퍼간의 대응관계를 정의한다. 각 랩퍼로 보내질 하위 질의가 결정이 되면, 미디에이터는 사용자 질의를 분해하여 완전한 하나의 XQuery 문으로 재작성 한다. 사용자 질의는 질의에 관련된 랩퍼의 개수만큼 분리가 되어 재작성 된다. 재작성되는 각각의 하위 질의는 질의에 사용된 기본 XML 뷰(MBV) 혹은 사용자 정의 XML 뷰(MAV)의 구조를 가지는 결과로 반환할 수 있도록 작성한다. 그 이유는 각 랩퍼에서 전달해 준 XML 문서를 통합하여 재구성할 때 필요한 정보를 추출하기 위한 경로탐색이 용이하기 때문이다. 하위 질의로 분리된 각 질의는 해당 랩퍼로 전달되고, 각 랩퍼에서 수행된 결과를 통합하여 사용자 질의에서 요구한 형태대로 XML 문서를 재구성하여 사용자에게 최종적으로 결과 값을 전달한다.

3.3 데이터마이닝 도구의 통합

데이터마이닝 도구 WEKA는 뉴질랜드의 Waikato 대학교에서 개발한 데이터 마이닝 시스템이다. WEKA를 미디에이터와 연동시킴으로서 다수의 분산된 정보원에 속한 데이터에 대하여 데이터마이닝을 수행할 수 있다. 미디에이터와 WEKA간의 상호 운용을 가능하게 하기 위해서는 미디에이터와 WEKA 시스템 사이에 통신할 수 있는 인터페이스를 설계해야 하는데, 그 원리도 XML 뷰 기반의 랩핑 모델과 동일하다. WEKA 시스템 사용자는 미디에이터에서 제시해 주는 뷰를 기반으로 미디에이터를 XML 정보원으로 간주한다. 따라서 사용자는 미디에이터에 등록되어 있는 정보원에 상관없이 미디에이터에서 제시하는 응용 XML 스키마(MAS)를 이용하여 사용자 질의를 생성시킨다. XQuery로 표현된 사용자 질의를 미디에이터로 전송하면, 미디에이터는 랩퍼와 연동하여 사용자 질의를 처리하여 그 결과로 XML 문서를 반환한다. WEKA 시스템은 ARFF(Attribute-Relation File Format) 형식으로 표현된 양식만을 인식하기 때문에 미디에이터에서 전송된 XML 형식을 ARFF 형식으로 변환해 주는 기능이 필요하다. (그림 1)에서 표현된 'XMLtoARFF' 모듈이 그 역할을 담당한다.

4. 뷰 정의

본 장에서는 생물학적 실험 데이터를 저장하고 있는 관계형 데이터베이스, BLAST와 같이 서열 유사도 검색을 수행하는 도구, 그리고 GenBank나 PubMed와 같이 생물학적 웹 자원에 대하여 XML 뷰 기반의 랩핑 모델을 제시하고 웹 자원과 BLAST에 대한 질의 처리 과정을 보인다.

4.1 관계형 모델에 대한 XML 뷰 정의

관계형 모델에 대한 뷰 정의의 기본 개념은 이미 제시된 기존 모델[21]과 유사하다. 관계형 데이터를 XML 문서로 사상하는 방법에는 여러 가지가 있을 수 있는데, 본 논문에서는 데이터베이스 전체를 하나의 XML 문서로 보는 기존 연구[21]와는 달리, 하나의 테이블을 하나의 XML 문서로 간주한다. 이때, 테이블의 이름이 XML 문서의 최상위 엘리먼트가 되고 테이블의 필드들은 그 하위 엘리먼트로 표현한다. 엘리먼트들은 XML 문서상에 반복되어 나타날 수 있으므로 이를 구분해 주기 위해 'tuple' 엘리먼트를 임의로 추가한다. 예를 들어 (그림 3)과 같이 단백질의 발현에 관련된 정보가 저장되어 있는 데이터베이스를 생각하자. 'PatientGel' 테이블에는 특정 환자의 gel에 대한 실험 정보를 유지하고, 'GelSpot' 테이블에는 'Gel Spot'의 발현 데이터를 담고 있다.

PatientGel

id	chartno	normal	minhH	maxpH	reagent	testDate	image
P01	1234	Y	10	20	AA	20031225	a.jpg
P02	5678	N	30	40	BB	20020606	b.jpg

GelSpot

id	p_id	sspNo	x	y	sizex	sizey	quantity	normalQty	sequence
G01	P01	2468	10	15	1	2	5000	50	a.txt
G02	P01	1357	30	55	3	4	3000	30	b.txt

(그림 3) 관계형 데이터베이스의 예

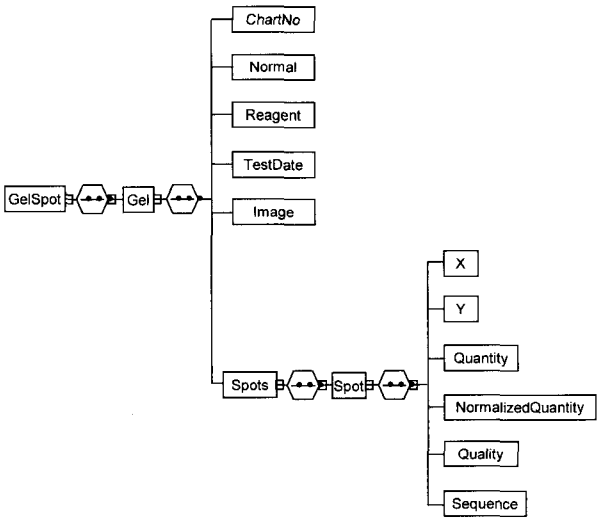
```
<? xml version = "1.0" ?>
< PatientGel >
 < tuple >
        < id > P01 </ id >
        < chartno > 1234 </ chartno >
        < normal > Y </ normal >
        <testData>
        < minpH > 10 </ minpH >
        < maxpH > 20 </ maxpH >
        < reagent > AA </ reagent >
        20031225 </ testDate >
        < image > a.jpg </ image >
</ tuple >
        < tuple >
        < id > P02 </ id >
        ...
        </ tuple >
        </ PatientGel >
        <? xml version = "1.0"? >
        < GelSpot >
        ... similar to < PatientGel >
</ GelSpot >
```

(그림 4) 기본 XML 뷰에 대한 XML 인스턴스 예

(그림 4)는 (그림 3)의 관계형 데이터를 XML 문서로 표현한 가상의 기본 XML 뷰(WBV) 인스턴스이다. 이는 랩퍼가 생성하는 기본 XML 뷰(WBV)에 의해 생성되는 가상의 XML 문서이지 실제로 이 문서가 명시적으로 생성되는 것은 아니다.

기본 XML 뷰(WBV)는 'XML Schema 1.0'로 자동으로 변환되어서 미디에이터에게 전달된다. 기본 XML 스키마(WBS)는 XML 문서에 대한 구조 정보와 엘리먼트에 대한 데이터 타입을 표현한다. 구조 정보는 정의된 기본 XML 뷰(WBV)에서 얻을 수 있고, 엘리먼트에 대한 데이터 타입은 데이터베이스의 시스템 카탈로그나 테이블의 메타정보를 통해서 얻을 수 있다. 이 때, 데이터베이스에서 제공하는 데이터 타입과 'XML Schema 1.0'에서 제공하는 데이터 타입간의 적절한 사상이 필요하다. 뷰로부터 스키마 생성 알고리즘은 이미 연구된 바 있어서 여기서는 생략한다[21].

미디에이터 관리자는 랩퍼가 제공한 기본 XML 스키마(WBS)를 이용하여 사용자 정의 XML 뷰를 정의할 수 있다. 예를 들어 (그림 4)에서 생성되는 스키마로부터 (그림 5)와 같은 XML 스키마를 가진 새로운 뷰를 정의한다고 가정하자.



(그림 5) 사용자 정의 XML 뷰에 대응되는 스키마

이를 위하여 사용자는 랩퍼가 생성해준 기본 XML 스키마(WBS)를 이용하여 XQuery로 표현된 사용자 정의 XML 뷰를 정의한다. (그림 6)은 (그림 5)와 같은 구조로 이루어진 가상의 뷰를 구성하기 위한 사용자 정의 뷰이다. 사용자 정의 XML 뷰도 랩퍼의 기본 XML 뷰(WBV)와 마찬가지로 가상적인 XML 문서이다. 관계형 모델에 대한 질의 처리 과정은 기존 연구[22]를 참고하고 여기서는 생략한다.

```
<GelSpot>
{
    FOR $PatientGel IN document("PatientGel")/tuple
    RETURN
        <Gel ID={$PatientGel/id/text()}>
            <ChartNo>$PatientGel/chartno/text()</ChartNo>
            <Normal>$PatientGel/normal/text()</Normal>
            <Reagent>$PatientGel/reagent/text()</Reagent>
            <TestDate>$PatientGel/testDate/text()</TestDate>
            <Image>$PatientGel/image/text()</Image>
            <Spots>
            {
                FOR $GelSpot IN document("GelSpot")/tuple
                WHERE $GelSpot/id = $PatientGel/id
                RETURN
                  <Spot>
                      <X>$GelSpot/x/text()</X>
                      <Y>$GelSpot/y/text()</Y>
                      <Quantity>$GelSpot/quantity/text()
                      </Quantity>
                      <NormalizedQuantity>$GelSpot/
                        normalQty/text()</
                        NormalizedQuantity>
                      <Quality>$GelSpot/quality/
                        text()</Quality>
                      <Sequence>$GelSpot/sequence
                        /text()</Sequence>
                </Spot>
            }
            </Spots>
        </Gel>
}
</GelSpot>
```

(그림 6) 사용자 정의 뷰의 예

(그림 7) 사용자 질의 수행 결과

(그림 7)은 관계형 모델에 대해서 XML 뷰 기반의 랩핑 모델을 적용한 후, 이 미들웨어 시스템을 사용할 수 있도록 설계한 사용자 인터페이스이다. 이는 관계형 데이터베이스에 접근하여 기본 XML 스키마(WBS)를 조회하고, 사용자 정의 XML 뷰를 생성, 수정, 삭제할 수 있으며, 사용자 질의를 수행할 수 있다. (그림 7)은 사용자 질의를 수행한 결과 화면이다. 왼쪽 상단에 있는 사용자 정의 XML 뷰의 리스트 중 하나를 선택하면 이에 해당하는 스키마(WAS)가 화면에 출력된다. 사용자는 이를 기반으로 왼쪽 하단의 사용자 질의 창에서 XQuery를 이용하여 사용자 질의를 기술한다. 수행된 질의 결과는 오른쪽 창에서 확인할 수 있다.

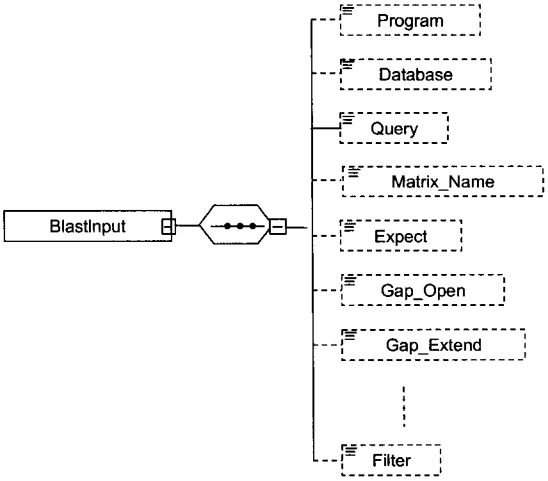
4.2 응용 프로그램에 대한 XML 뷰 정의

응용 프로그램에 대한 통합의 예를 들기 위하여 여기서는 서열 유사도 검색 도구인 BLAST 통합에 대하여 논한다. 제시된 통합 모델은 다른 도구들에 대해서도 그대로 적용될 수 있다. 만약 응용 프로그램 통합 모델이 앞에서 논한 정보원 통합 모델과 서로 상이하다면 설계와 구현상에 이중의 노력을 들여야 한다. 본 논문에서 제시하는 모델은 응용 프로그램과 정보원 모두에 대하여 동일한 개념으로 통합할 수 있는 모델이다. 즉, 응용 프로그램에 대해서도 앞에서 논한 XML 뷰 기반으로 랩핑한다.

응용 프로그램에 대하여 XML 뷰를 정의하기 위해서는 응용 프로그램도 데이터를 가지는 하나의 정보원으로 간주하는 관점이 필요하다. 사용자가 BLAST에 질의할 때는 BLAST가 질의 양식에 대한 정보를 보유하고 있는 것으로 간주될 수 있다. 또한, BLAST의 출력 결과에 대해서도 BLAST가 출력 양식에 대한 정보를 보유하고 있는 것으로 간주할 수 있다. 따라서 응용 프로그램의 입력 양식과 출력 양식이 바로 응용 프로그램이 제공하는 데이터에 대한 뷰가 된다. 이 뷰를 XML 스키마로 간주할 때 그것이 기본 XML 뷰(WBV)로 정의되어, BLAST를 XML 문서로 구성된 XML 정보원으로 간주될 수 있다.

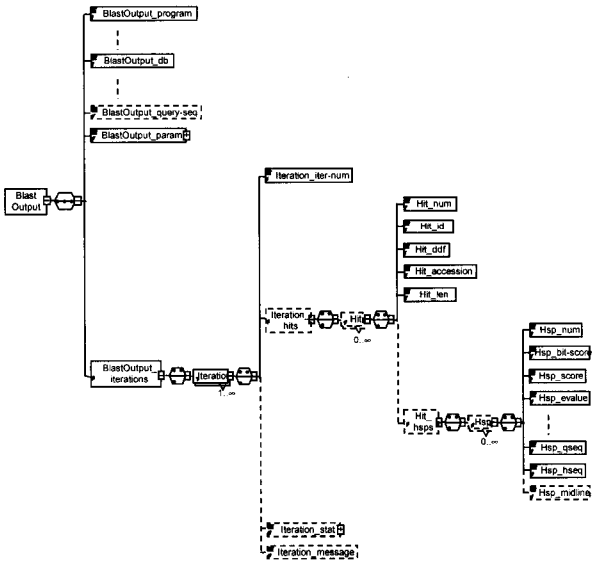
BLAST의 입력양식으로는 5종류의 BLAST 프로그램 중에서 하나의 수행 프로그램을 선택하는 'Program'과 시퀀스에 대한 검색 대상이 되는 'Database', 그리고 실질적으로 유사도를 비교하게 될 시퀀스인 'Query'가 있다. 'Query'는 크게 3가지 양식을 지원하는데, FASTA 양식, Bare Sequence 형식, 그리고 바(bar)형식으로 구분된 NCBI의 시퀀스 식별자 양식 중 하나를 선택하여 'Query'로 기술한다.

위의 기본적인 입력양식 외에도 보다 더 세부적인 제약조건을 명시할 수 있는 세부사항들을 'Matrix_Name', 'Expect' 등의 옵션을 통해 나타낼 수 있다. (그림 8)은 BLAST 입력 정보에 대한 기본 XML 뷰(WBV)를 구조적으로 표현한 것이다.



(그림 8) BLAST 입력에 대한 기본 XML 뷰

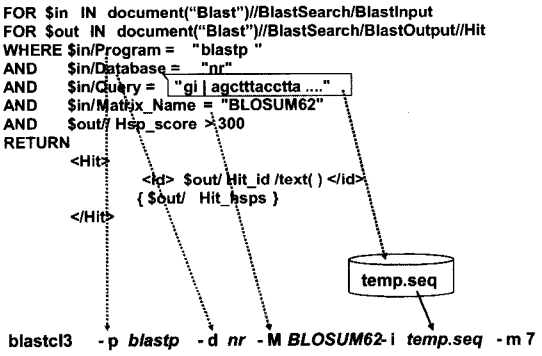
BLAST의 출력 정보에 대한 스키마로 입력 정보와 비슷하게 구성될 수 있다. (그림 9)는 BLAST의 출력정보에 대한 기본 XML 뷰(WBV)를 구조적으로 표현한 것이다.



(그림 9) BLAST 출력에 대한 기본 XML 뷰

이와 같이 기본 XML 뷰(WBV)가 정의된 이후에는 사용자 정의 XML 뷰와 질의 처리 방법은 앞에서 논한 정보원에서의 처리 방법과 동일하게 이루어지기 때문에 전체적으로 일관된 모델로서 설계될 수 있다. (그림 10)은 이와 같이 정의된 XML 뷰에 대하여 질의의 예와 그 변환 과정을 나타낸 것이다. FOR 절에서는 입력에 관련된 엘리먼트인 'BlastInput'과 출력에 관련된 'BlastOutput'의 자손 엘리먼트인 'Hit'를 각각 변수 $in과 $out에 바인딩 시킨다. WHERE 절에서는 입력 양식에서 사용자가 입력해야 할 BLAST의 프로그램 명, 데이터베이스, 그리고 시퀀스를 지정하였으며, BLAST 출력 결과 중 'Hsp_score'가 300이 넘는 내용으로 필터링하는 질의이다. 이 질의에 대한 수행 결과는 Return 절에서 명세한 XML 문서이다.

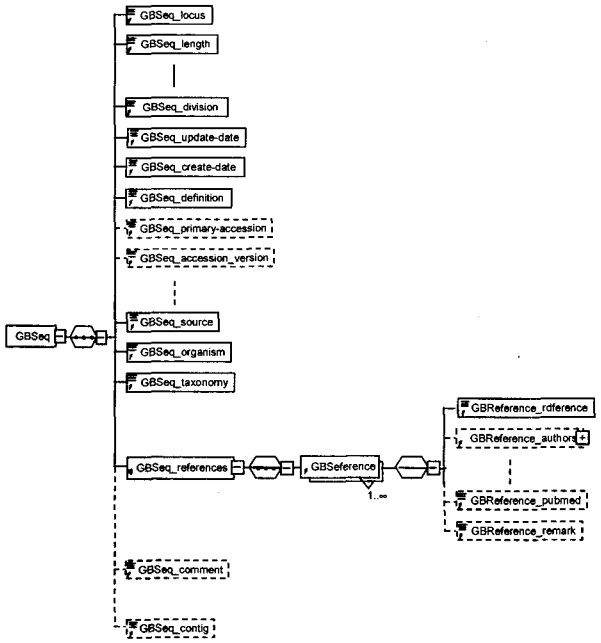
XQuery 질의를 BLAST 명령어 라인으로 사상되는 과정을 살펴보면, BLAST의 입력에 대한 기본 XML 뷰(WBV)의 바인딩 변수인 '$in' 하위의 Program 엘리먼트는 -p 옵션으로 사상되어 5종류의 BLAST 프로그램 중에서 하나의 수행 프로그램을 선택할 수 있도록 한다. '$in' 하위의 Database 엘리먼트는 -d 옵션으로 사상되어 시퀀스에 대한 검색 대상이 되는 데이터베이스를 기술한다. 또한 Query 엘리먼트는 실질적으로 유사도를 비교하게 될 시퀀스를 'temp.seq' 파일에 저장하여 -i 옵션으로 사상한다. 'Matirx_Name' 엘리먼트는 측정 행열에 대한 정보를 기술하고 있으며 이를 -M 옵션으로 사상한다. 마지막으로 -m 옵션을 '7'로 지정함으로서 BLAST의 수행결과를 XML 형식으로 반환한다.



(그림 10) XQuery 질의를 BLAST 명령어라인으로 사상

4.3 Entrez에 정보원에 대한 XML 뷰 정의

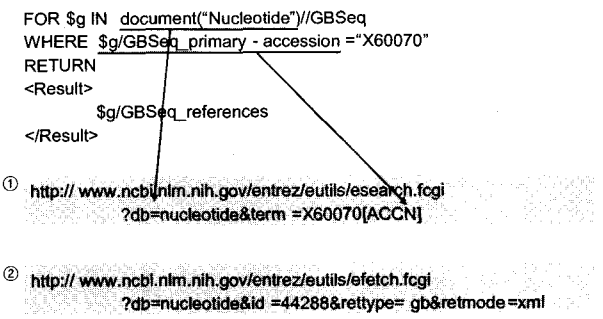
이 장에서는 Entrez 검색 엔진을 대상으로 XML 뷰를 정의하는 방법에 대해서 기술한다. NCBI에서 제공하는 Entrez 검색엔진을 통하여 염기서열, 단백질 서열, 단백질의 3차원 구조 정보, 그리고 이와 관련된 문헌정보에 대한 통합 검색을 수행할 수 있다. Entrez에 대한 XML 기반의 뷰 정의는 BLAST와 유사하게 이루어진다. Entrez의 출력 양식을 기본 XML 뷰(WBV)로 정의하고, Entrez의 수행 결과로 만들어진 출력 양식을 가상의 XML 문서로 사상하여 XML 정보원으로 표현한다. Entrez의 수행 결과에 대한 양식은 데이터베이스마다 서로 다르기 때문에, 각 데이터베이스의 출력 구조가 Entrez의 기본 XML 뷰(WBV)가 된다. 따라서 염기 서열, 단백질 서열, 참고 문헌 데이터베이스 등의 정보원에 대한 출력 구조는 각각 하나의 XML 문서구조로 간주되며, 그 전체가 Entrez의 기본 XML 뷰(WBV)가 된다. (그림 11)은 염기 서열 데이터베이스에 대한 기본 XML 뷰(WBV) 스키마 다이어그램이다.



(그림 11) 염기 서열 데이터베이스에 대한 기본 XML 뷰

이와 같이 정의된 기본 XML 뷰(WBV)를 바탕으로 앞에서 제시한 사용자 정의 XML 뷰를 정의할 수 있으며 사용자 질의를 작성할 수 있다. 랩퍼는 Entrez가 수행할 수 있는 질의어로 사용자 질의를 변환해야 한다. NCBI에서는 사용자가 Entrez에 직접 접근하여 검색할 수 있는 웹 기반의 인터페이스를 기본적으로 제공하지만, 한편으론 Entrez 인터페이스가 아닌 외부의 환경에서 간접적으로 Entrez에 접근할 수 있는 도구인 'Entrez Programming Utilties'를 제공한다. 그 기능으로 EFetch와 ESearch 등이 있다. EFetch는 id와 같이 하나의 완전한 데이터나 문서를 직접적으로 반환하는 질의 유형이고, ESearch는 주어진 키워드에 관련된 문서의 리스트를 반환하는 유형이다. 이들은 외부 환경에서 URL을 통해 Entrez의 기능을 수행할 수 있도록 다양한 파라미터들을 제공한다. Entrez 랩퍼는 미디에이터에서 전달받은 XQuery로 표현된 사용자 질의를 EFetch나 ESearch에서 제공하는 파라미터에 맞게 사상시켜 완전한 하나의 URL을 구성한다. (그림 12)는 (그림 11)에서 정의된 기본 XML 뷰(WBV)에 대한 XQuery 질의 예와 해당 질의가 Entrez에서 수행할 수 있도록 EFetch의 파라미터를 이용하여 완전한 하나의 URL 형식으로 변환되는 과정을 나타낸 것이다.

FOR 절에서는 염기 서열 데이터베이스에 대한 기본 XML 뷰(WBV)의 엘리먼트인 'GBSeq'를 변수 '$nucleotide'에 바인딩 시킨다. WHERE 절에서는 바인딩 변수의 하위 엘리먼트인 'GBSeq_primary-accession'의 값이 'X60070'을 만족하는 레퍼런스 결과를 얻을 수 있도록 조건을 명시하고 있다.



(그림 12) XQuery 질의를 Entrez 질의로의 사상

이와 같은 사용자 질의를 Entrez 질의로 사상하는 과정을 살펴보면, 우선 ESearch의 기본 URL이 명시되고, XQuery의 WHERE절에 기술되어 있는 엘리먼트들을 ESearch에서 제공하는 파라미터를 이용하여 사상한다. FOR 절에 명시되어 있는 'Nucleotide'는 파라미터 'db'를 이용하여 염기 서열 데이터베이스를 의미하는 'nucleotide'에 사상되며, WHERE절에 명시되어 있는 내용은 파라미터 'term'에 사상된다. ESearch의 'term' 파라미터는 검색하고자 하는 세부항목명과 해당 값으로 표현되는데, 'GBSeq_primary-accession' 엘리먼트는 'ACCN' 항목과 사상된다. ①번과 같이 완성된 ESearch URL을 수행하면 그 조건을 만족하는 'id'의 목록들이 출력된다. 이 'id'들을 이용하여 EFetch URL을 만든다. ②번의 EFetch URL을 살펴보면 우선, EFetch의 기본 URL이 명시되고, 이후 파라미터 'db'는 그대로 염기 서열 데이터베이스를 의미하는 'nucleotiode'를 기술하며, 파라미터 'id'의 값으로 ①번 URL의 검색결과로 추출된 'id'값을 기술한다. 이 외에도 'rettype'을 통해 수행 결과의 템플릿을 지정할 수 있는데 그 값으로 'gb'를 선택하여 Entrez에서 제공하는 출력 템플릿 중 'GenBank View' 형태로 결과 값을 출력하도록 명시한다. 또한 'retmode'를 통해 질의의 수행 결과를 XML 문서 형태로 출력하도록 명시하고 있다. 이 경우 랩퍼에서는 미디에이터로 전달하기 위한 통일된 양식인 XML 문서로 변환하는 일을 생략할 수 있다.

5. 결 론

이질적인 생물 정보원을 통합하기 위해서는 각 정보원의 이질적인 내용을 숨기고 하나의 관점으로 표현하는 뷰를 정의해야 한다. 뷰 정의 모델은 통합 시스템의 설계 방향에 큰 영향을 미친다.

본 논문에서는 사용자 정의 XML 뷰를 지원하는 XML 기반의 뷰 정의 모델을 제시하였다. 이를 기반으로 관계형 데이터베이스와 웹 자원, 그리고 응용 프로그램에 대해서도 같은 모델을 사용하여 랩핑한 결과를 제시하고 정의된 뷰

에 대한 질의처리 방법을 간단히 소개하였다. 본 논문에서 제시한 랩핑 모델은 통합하고자 하는 자원에 대해서 동일한 랩핑 모델을 사용하고 있기 때문에 융통성이 증대되며, 새로운 정보원 추가에 대해서도 확장성을 고려한 범용적인 랩핑 모델이다.

앞으로, 개발된 통합 미들웨어를 활용하여 워크플로우 분석을 통하여 분자 생물학자의 다양한 요구를 수용할 수 있는 사용자 인터페이스 도구를 개발할 필요가 있고, 미디에이터에서 보다 최적화된 통합 질의 처리에 대한 연구가 필요하다.

참 고 문 헌

[1] Thomas Hernandez, Subbarao Kambhampati, "Integration of Biological Sources : Current Systems and Challenges Ahead," ASU CSE TR-03-005, pp.3-5, October, 2003.

[2] B. Eckman, Z. Lacroix, L. Raschid, "Optimized Seamless Integration of Biomolecular Data," Proc. of the IEEE 2nd International Symposium on Bioinformatics and Bioengineering Conference, pp.1-3, 2001.

[3] D. Benson, I. Karsch-Mizrachi, D. Lipman, J. Ostell, D. Wheeler, GenBank, Nucleic Acids Res, Vol.31, No.1, January, 2003.

[4] SRS Documentation, [Online]. Available : http://srs.hgmp.mrc.ac.uk.

[5] Swiss-Prot Documentation, [Online]. Available : http://www.ebi.ac.uk/swissprot

[6] "Practical Data Integration in Biopharmaceutical R&D : Strategies and Technologies," White Paper, 3rd Millennium Inc, http://www.3rdmill.com, pp.9-12, May, 2002.

[7] L. M. Haas, P. M. Schwarz, P. Kodali, E. Kotlar, J. E. Rice, W. C. Swope, "DiscoveryLink : A System for Integrated Access to Life Sciences Data Sources," IBM systems journal, Vol.40, No.2, pp.5-9, 2001.

[8] I. A. Chen, V. M. Markowitz, "An Overview of the Object-Protocol Model(OPM) and OPM Data Management Tools," Inform. Syst., Vol.20, No.5, pp.5-10, April, 1995.

[9] BLAST, [Online]. Available : http://www.ncbi.nlm.nih.gov/BLAST, 2003.

[10] A. S. Kosky, I. A. Chen, V. M. Markowitz, E. Szeto, "Exploring Heterogeneous Biological Databases : Tools and Application," Proc. of the 6th International Conference on Extending Database Technology, pp.3-5, 1998.

[11] T. Topaloglou, A. S. Kosky, V. M. Markowitz, "Seamless Integration of Biological Applications within Database Framework," pp.5-6, ISMB, pp.272-281, 1999.

[12] Val Tannen, "The Information Integration System K2," white paper, http://db.cis.upenn.edu/K2/papers.html," pp.4-5.

[13] Zoe Lacroix, "Biological Data Integration : Wrapping Data and Tools," IEEE Transactions on information technology in biomedicine, Vol.6, No.2, pp.2-4, June, 2002.

[14] Entrez-Search and Retrieval System, [Online].Available : http://www.ncbi.nlm.nih.gov/Entrez, 2003.

[15] Ian H. Witten and Eibe Frank, "Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations," Morgan Kaufmann, pp.265-271, 2000.

[16] World-Wide Web Consortium, "XML Schema Part 0 : Primer," [Online].Available : http://www.w3c.org/TR/xmlschema-0/, W3C Recommendation, 2001.

[17] World-Wide Web Consortium, "XML Schema Part 1 : Structures," [Online].Available : http://www.w3c.org/TR/xmlschema-1/, W3C Recommendation, 2001.

[18] World-Wide Web Consortium "XML Schema Part 2 : Datatypes," [Online].Available : http://www.w3c.org/TR/xmlschema-2/, W3C Recommendation, 2001.

[19] World-Wide Web Consortium, "XQuery 1.0 : An XMl Query Language," [Online].Available : http://www.w3c.org/TR/xquery/, W3C Working Draft, 2003.

[20] 정채영, 최규원, 김영옥, 김영균, 강현석, 배종민, "관계형 데이터베이스에서 XML 뷰 기반의 질의 처리 모델", 정보처리학회논문지D, 제10-D권 제2호, pp.221-232, April, 2003.

[21] 정채영, 김영옥, 이미영, 강현석, 배종민, "사용자 정의 뷰 기반의 XML 스키마 관리 시스템", 정보처리학회논문지D, 제10-D권 제3호, pp.367-374, June, 2003.

박 은 경

e-mail : pek1028@hanmail.net
1999년 경상대학교 컴퓨터과학과 학사
2002년 경상대학교 컴퓨터과학과 석사
2002년~현재 경상대학교 대학원 컴퓨터과학과 박사과정
관심분야 : XML, 데이터베이스 통합, 바이오인포매틱스

강 동 완

e-mail : wanni76@naver.com
2002년 경상대학교 컴퓨터과학과 학사
2004년 경상대학교 컴퓨터과학과 석사
관심분야 : XML, 데이터베이스 통합, XML 질의처리

정 채 영

e-mail : lockey@orgio.net

1997년 경상대학교 전자계산학과 학사

2001년 경상대학교 대학원 컴퓨터과학과 석사

2004년 경상대학교 대학원 컴퓨터과학과 박사

관심분야 : XML, WWW, 데이터베이스, 정보검색

배 종 민

e-mail : jmbae@nongae.gsnu.ac.kr

1980년 서울대학교 수학교육과 학사

1983년 서울대학교 대학원 계산통계학과 석사(전산학)

1995년 서울대학교 대학원 계산통계학과 박사(전산학)

1982년~1984년 한국전자통신연구소 연구원

1997년~1998년 Virginia Tech. 객원연구원

1984년~현재 경상대학교 전임강사, 조교수, 부교수, 교수

관심분야 : XML, 데이터베이스 통합, 생물정보, 데이터마이닝