XML 뷰 기반의 생물 정보원 통합 시스템 개발

정재훈* 박은경* 정채영* 김현주** 배종민*
*경상대학교 컴퓨터과학과 **진주산업대학교 컴퓨터공학부
dnachiu@nate.com pek1028@hanmial.net lockey@orgio.net khj@jinju.ac.kr jmbae@gsnu.ac.kr

Development of an Integration System for Biological Information Sources based on XML Views

Jae-Hoon Jung* Eun-Koung Park* Chai-Young Jung* Hyun-Ju Kim** Jong-Min Bae*
*Dept. of Computer Science, Gyeongsang National University
**Dept. of Computer Science & Engineering, Jinju National University

요 약

생물정보원은 이질성이 높고 사용자의 요구사항이 다양하다. 본 논문은 이러한 이질성을 해결하고 사용자의 다양한 요구사항에 쉽게 대처할 수 있는 XML 기반의 생물정보원 통합시스템의 설계개념과 구조 및 구현결과를 제시한다. 제시하는 통합시스템은 관계형테이블, 객체, XML, 플랫파일 등 다양한 자료형을 지원하면서, 관계형, 객체관계형, 웹자원, 응용프로그램 등 데이터 관리모델에 무관한 뷰 정의 및 질의처리모델이다. 그리고 사용자정의 XML 뷰 기반의 뷰 관리 및 질의처리를 통하여 사용자의 다양한 요구사항에 쉽게 적응할 수 있는 미디에이터 질의처리 기반의 생물정보원 통합시스템을 제시한다.

1. 서론

생물학적 실험환경의 급격한 발전으로 인하여 많은 생물정보원이 개발되고 또한 방대한 생물데이터를 분석하고 처리하기 위한 소프트웨어 도구들이 개발되었다. 이에 따라 여러 생물 정보원에 접근하여 결과를 검색하고 조작하는 과정을 자동으로 수행할 수 있는 생물 정보원 통합 시스템의 필요성이 대두 되었다.

생물정보원들은 역사적인 이유로 인하여 이질성이 매우 높다. 이에 따라 정보원 통합에 관한 연구는 그 역사가 오래되었음에도 불구하고, 생물정보원의 통합문제는 새로운 이슈가 되었다[1]. 생물정보원의 데이터는 관계형 모델, 객체형 모델, 혹은 플랫(flat) 파일로 관리되기도 하고 웹 데이터베이스로 제공되기도 한다. 또한 정보원에 접근하기 위해서는 정보원이 제공하는 프로그래밍 인터페이스를 활용할 수도 있고, SQL과 같은 표준 질의어를 사용할 수도 있으며, 웹 검색엔진을 사용할 수도 있고, 폼(form)기반의 인터페이스를 사용할 수도 있다. 그리고 질의에 대한 결과는 관계형 튜플(tuple)이나 객체일 수 있고, XML 혹은 HTML 문서일 수도 있으며, ASN.1과 같은 데이터교환 양식일 수도 있다. 분산된 생물정보원을 물리적으로 혹은 가상적으로 통합하기 위해서는 이와 같은 정보원들의 이질성을 해결해야 한다.

생물 정보원을 통합하는 대표적인 방법에는 크게 링크 기반의 통합방법, 데이터 웨어하우스 기반의 통합 방법, 그리고 미디에이터 기반의 통합 방법 등이 있는데, 본 논문은 미디에이터 모델 하에서 XML 기반의 통합 방법에 기초를 두고 있다. 이는 이질의 각 정보원을 모두 하나의 XML 정보원으로 보는 관점 즉 XML 뷰를 제공하여 XML 기반의 질의처리를 수반한다. 즉 통합시스템은 사용자에게 가상의 XML 정
보원을 제공한다. 이때 XML 뷰를 정의하기 위한 언어로는 XQuery를 사용하고, 개별 정보원에서 정의된 스키마를 기반으로 만들어지는 통합정보원의 스키마는 XML-Schema를 사용하여 표현하며, 사용자질의어는 XQuery를 사용하고, 질의수행결과는 XML 문서로 제공된다.

한편, 보다 편리한 통합시스템이 되기 위해서는 정보원 통합 뿐 아니라, 유전자 서열의 상동성 비교와 같은 생물학적 분석도구도 통합된 정보원과 연동되어야 하며, 데이터마이닝 도구도 통합정보원과 연동되어야 한다. 이때 연동모델은 정보원 통합모델과 일관성을 유지하여 설계해야만 새로운 정보원이 추가될 때 유연하게 대처할 수 있다.

또한, 생물 정보원의 특성상 사용자의 요구가 매우 다양하다. 이러한 다양한 요구를 수용하기 위해서는 개별 사용자의 요구에 맞는 통합시스템을 쉽게 구축할 수 있는 기반이 필요하다. 이를 위하여 본 논문에서는 관계형 데이터베이스에서 사용자 정의 뷰를 지원하는 것과 마찬가지로 가상적인 XML 정보원에 대해서도 사용자 정의 XML 뷰를 지원한다. 사용자는 사용자 정의 뷰를 통해서 정보원을 추상화 시킬 수 있으며, 사용자의 목적에 따라서 정보원의 스키마를 재구성할 수 있어서 질의 설계에 큰 융통성을 제공할 수 있다.

마지막으로, 본 통합 시스템에서는 새로운 정보원이 추가된다 하더라도 최소한의 비용으로 통합이 가능하도록 확장성을 고려하여 설계한 미들웨어 시스템이다.

본 논문에서는 위에서 제시한 요구사항, 즉 첫째, 정보원의 이질성 극복, 둘째, 일관된 통합모델을 기초로 한 데이터베이스, 응용프로그램, 그리고 데이터마이닝 도구의 통합, 셋째, 사용자정의 뷰 기반의 질의처리를 통한 개별화된 통합시스템 구축의 용이성, 넷째, 시스템 확장의 용이성, 마지막으로 사용자가 쉽게 통합질의를 가능하게 하는 사용자 중심의 질의인터페이스 등을 만족시키는 범용 통합시스템에 대한 설계개념을 제시한다.

2. 관련연구

그동안 다양한 생물정보원 통합시스템이 개발되어 왔는데, 여기서는 그 중에서 대표적인 몇 개의 시스템을 제시한다.

SRS[2]는 EMBL/EBI&Lion Bioscience 사에서 개발한 링크 기반의 대표적인 생물 통합 시스템이다. SRS는 SwissProt이나 GenBank와 같이 구조적인 레코드 형식으로 구성되어 있는 플랫파일을 파싱하여 내부적으로 데이터를 저장함과 동시에 데이터 간의 인덱스를 생성해 놓고, 후에 이를 기반으로 질의를 수행한다. 또한 SRS는 정보원들 간에 서로 연관이 있는 데이터들을 연결시켜 두고, 질의 수행시 사용자에게 관련된 정보들을 제공해 주기 위하여 교차참조(cross-reference) 기능을 제공한다. SRS는 웹을 이용한 링크 기반의 통합 시스템이기 때문에 사용자가 시스템에 대한

전문적인 지식 없어도 편리하게 사용할 수 있는 장점이 있는 반면에 이미 정해져 있는 질의 플랜을 바탕으로 수행되기 때문에 정보원에 대한 제한된 접근을 허용하는 제약점이 따른다. 또한, 통합 대상이 주로 데이터 정보원이다.

OPM*QS[3][4]는 뷰 기반의 통합 방법론을 사용하고 있다. 생물 정보원들을 OPM 기반으로 모델링하기 때문에 각 정보원은 하나의 객체로 표현되며, 객체 id와 속성을 이용하여 객체 클래스를 기술한다. 질의어로는 OPM*QL을 사용하며, CORBA API를 이용하여 원격지의 데이터 베이스 연결을 수행한다. OPM*QS에서는 멀티미디어 데이터 타입이나 특정 응용 프로그램에 대해서도 OPM 기반의 뷰를 정의할 수 있도록 ASDT(Application Specific Dat Type)라는 별도의 타입을 제공한다. ASDT 타입의 OPM 클래스에서는 특정 기능을 수행할 수 있는 별도의 메소드들이 정의되어 있고, 사용자가 OPM*QL 질의어에 해당 메소드를 기술함으로서 기능을 수행할 수 있다.

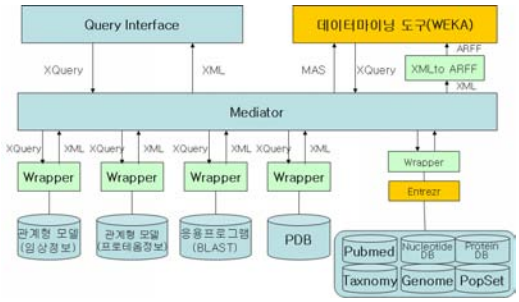
IBM에서 개발한 DiscoveryLink[5]는 멀티미디어 데이터 통합 시스템인 Garlic과 DB2 기반이 통합 상품인 DataJoiner를 생물 데이터베이스 통합에 맞게 확장하였다. 각각의 DB 내용에 대한 전역 스키마를 정의하기 위하여 SQL 문법을 모방한 별도의 언어를 사용하고 있다. K2 시스템은 Kleisli 시스템을 기반으로 한 미디에이터 기반의 시스템이다. K2에서는 통합하고자 하는 각 생물 정보원들을 ODL(Object Definition Language)을 이용하여 표현하며, 정보원의 실제적인 내용과 ODL로 기술될 클래스 사이이 사상을 위해 K2MDL(K2 Mediator Definition Language)이라는 언어를 사용한다.

이와 같이 기존의 연구에서는 객체 모델을 사용해서 뷰를 정의하거나 혹은 별도의 뷰 정의 언어를 사용하는데 반하여, 본 논문에서는 XML 기반의 뷰를 정의함으로서, 모든 생물 정보원을 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. 특히, 본 논문에서 제시하는 모델은 특정 정보원모델에 독립적이어서 관계형, 객체형은 물론이고, 웹 자원과 응용 프로그램에 대한 통합도 일관된 모델로 설계된다.

3. 통합시스템 구조

개발된 통합시스템의 구조는 (그림 1)과 같다. 통합 대상으로는 임상정보와 프로테옴 정보를 자체에서 전용으로 관리하는 관계형 데이터베이스, 단백질 서열 비교 프로그램 BLAST, NCBI에서 제공하는 Entrez로 검색가능한 GenBank, Pubmed, Nucleotide 데이터베이스와 단백질의 3차원 구조를 검색할 수 있는 PDB등 웹 정보원, 그리고 데이터마이닝 도구인 WEKA(Waikato Environment for Knowledge Analysis)이다. 통합 시스템은 크게 미디에이터, 랩퍼, 사용자 질의인터페이스, 데이터마이닝 도구인 WEKA, 4부분으로 구성되어 있다.

랩퍼는 개별 정보원과 직접 대화하는 시스템이다. 랩퍼는 개별 정보원의 스키마를 XML 뷰로 정의하여 그 결과를 XML Schema로 변환하여 미디에이터에게 전달하며, 미디에이터로부터 받은 XQuery 질의어를 개별 정보원의 질의어로 변환하여 정보원에 대한 검색을 행하며, 검색결과를 XML 문서로 변환하여 미디에이터에게 전달한다. 미디에이터는 지역정보원에 대한 XML Schema를 랩퍼로부터 받아서 통합된 가상의 XML 스키마를 관리하면서 사용자에게 제시하고, 사용자가 제시한 XQuery 통합질의를 분해하여 각각의 랩퍼에게 전달하며, 랩퍼로부터 받은 질의결과를 통합하여 사용자에게 검색결과를 제시한다. 사용자는 질의인터페이스에서 제공하는 스키마를 기반으로 질의하며, 사용자 질의는 XQuery로 표현되어서 미디에이터에게 전달한다. 미디에이터는 필요한 스키마를 데이터마이닝 도구 WEKA에 전달하며, WEKA는 이 스키마를 기반으로 미디에이터에게 정보를 검색한다. 검색결과인 XML 문서는 WEKA 고유의 입력양식인 ARFF 양식으로 변환되어서 전달된다.



(그림 1) 통합시스템 구조

3.1 XML 뷰 정의

이질의 정보원을 통합하기 위해서는 각 정보원이 관리하는 데이터에 대하여 그 모델에 무관한 통일된 모델로 표현해야 한다. 본 시스템은 모든 정보원을 하나의 가상적인 XML 정보원으로 간주하여 각 정보원의 내용은 모두 XML Schema로써 표현한다. 이때, 정보원 스키마와 SML 스키마 사이에 사상이 필요한데, 관계형의 경우 2차원 테이블을 임의의 차원인 XML 스키마로 사상하는 방법은 다양하게 있는데, 본 시스템은 [6]와 유사한 개념으로 사상한다. 객체형의 경우는 임의의 객체가 객체를 가질 수 있기 때문에 XML 스키마와 가장 쉽게 사상할 수 있다.

정보원의 스키마가 알려져 있지 않은 웹 자원의 경우는 웹 자원 자체의 검색기능을 그대로 활용할 수 있도록 XML 스키마를 정의한다. 이를 위하여 대부분의 생물정보원은 검색결과에 대한 출력양식이 일정하게 정해져 있어서 그 출력양식을 그 정보원이 관리하고 있는 데이터의 스키마로 간주할 수 있다. 이때 NCBI에서 제공하는 정보원과 같이 정보원에서 제공하는 출력문서가 XML파일이면 그 파일로부터 XML 스키마를 유도한다. 만약 출력문서 양식이 PDB와 같이 플랫(flat)파일 형식이면, 플랫파일도 일정한 양식을 갖추고 있기 때문에 플랫파일로부터 XML 스키마를 유도한다.

또한, BLAST와 같은 응용프로그램의 경우는 입력데이터의 내용이 출력데이터의 양식에 포함된다는 보장이 없다. 따라서 응용프로그램의 경우는 입력데이터도 정보원이 관리하는 데이터로 간주해야만, 나중에 질의변환이 가능하다. 따라서 응용프로그램의 경우는 출력양식과 입력양식 모두를 하나의 스키마로 간주하여 XML 스키마로 변환한다. 이러한 개념은 [7]에서 제시한 XML 뷰 정의 방법에 따른 것이다.

이렇게 각 정보원을 XML 스키마로 정의한 것을 기본 XML 뷰(view)라 한다. 일단 기본 XML 뷰가 정의된 이후에는 시스템의 융통성과 개별화된 통합시스템 구축을 지원하기 위하여 사용자정의 XML 뷰를 정의한다. 이는 관계형 모델에서 SQL로써 뷰를 정의하는 개념과 유사하다. 즉, 기본 XML 뷰로부터 XQuery를 이용하여 다양한 사용자정의 XML 뷰가 정의될 수 있으며 각 뷰는 사용자의 필요에 따라서 제시될 수 있다.

기본 XML 뷰와 사용자정의 XML 뷰는 자체 개발한 XQuery 파서를 통하여 통합시스템 내에서 뷰 트리 형태로 유지된다. 미디에이터와 랩퍼 모두 사용자정의 XML 뷰를 지원하기 때문에, 뷰 트리는 미디에이터와 랩퍼 양쪽에서 모두 유지되며, 미디에이터 경우는 질의분해를 통하여 각 랩퍼로 보낼 XQuery 질의어를 구성할 수 있도록 뷰 트리노드 상의 정보가 추가된다. 뷰 트리는 사용자

질의어를 분해하고 변환하기 위한 핵심자료구조이며, 또한 결과문서 생성을 위한 템플릿으로도 활용된다.

3.2 랩퍼

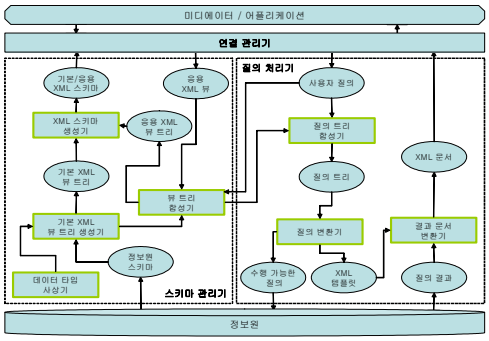
랩퍼는 그림 2에서 보는 것과 같이 크게 연결 관리기, 스키마 관리기, 질의 처리기로 구성되어 있다. 연결 관리기는 미디에이터와 랩퍼의 대화를 위한 인터페에스를 제공한다. 랩퍼는 미디에이터와 같은 기계에 있을 수도 있고 다른 기계에서 동작할 수 있기 때문에, 스키마 전달, 질의전달, 결과문서전달 등과 같은 대화를 가능하도록 하기 위하여 자바 RMI로 구현되었다.

스키마 관리기는 기본 XML 뷰 트리 생성기, 뷰 트리 합성기, XML 스키마 생성기, 데이터 타입 사상기, XQuery 파서로 구성된다. 기본 XML 뷰 트리 생성기는 정보원으로부터 기본 XML 뷰를 생성하는 모듈이다. 관계형이나 객체관계형 경우는 시스템 카탈로그로부터 자동으로 생성되는데, 실제 XML 문서를 생성하지 않고 내부적으로 트리로 표현해서 처리한다. 이때 데이터 타입 사상기는 정보원에서 제공하는 데이터 타입과 XML Schema에서 제공하는 데이터 타입의 불일치에 의해 생기는 문제점을 해결한다. 웹 자원에 대해서는, XML 스키마를 제공하는 웹 자원의 경우는 이를 우리의 요구에 맞게 수작업으로 약간 수정하여 기본 XML 스키마를 만들어서 이로부터 기본 XML 뷰 트리를 생성한다. 플랫파일을 제공하는 웹 자원에 대해서는 개별 정보원마다 플랫파일을 XML 스키마로 변환하는 도구를 개발한 후, 이를 다시 기본 뷰 트리로 변환한다.

사용자정의 XML뷰 또한 뷰 트리로 표현된다. 사용자정의 XML 뷰는 XQuery로써 작성되기 때문에, XQuery 파서를 기반으로 뷰 트리가 생성된다. 이때 정보원에 대한 질의변환을 위하여, 사용자정의 XML 뷰 트리 생성시, 기본 XML 뷰 트리 노드와의 관계를 유지하고 있어야 한다. (그림 2)에서 트리 합성기의 일부 기능이 이 역할을 담당한다.

XML 스키마 생성기는 기본 XML 뷰 트리로부터 기본 XML 스키마를 생성하고 사용자 정의 XML 뷰 트리로부터 사용자 정의 XML 스키마를 생성한다. 그리고 그 결과는 미디에이터에게 전달된다.

질의 처리기는 질의 트리 합성기, 질의 변환기, 경과 문서 변환기로 구성된다. 사용자 질의는 사용자 정의 XML 뷰 혹은 기본 XML 뷰에 대하여 질의가 이루어진다. 따라서 사용자질의어가 정보원의 질의어로 변환되기 위해서는 사용자질의 도메인, 사용자정의 XML 뷰 도메인, 기본 XML 뷰 도메인 사이의 관계를 설정하는 기능, 즉, 질의어 합성이 필요하다. 질의 트리 합성기는 사용자 질의와 사용자 질의에 사용된 XML 뷰를 합성하여 질의 트리를 생성하는 역할을 한다. 질의트리에는 정보원에 대한 질의변환을 위한 모든 정보가 포함되어 있다.



(그림 2) 랩퍼 시스템 구조

질의 변환기는 질의 트리를 순회하면서 정보원에서 수행 가능한 질의어를 생성한다. 사용자질의는 합성을 통하여 이미 기본 XML 뷰와의 관계가 설정되어 있고, 기본 XML 뷰는 정보원의 스키마와 거의 대응관계에 있기 때문에 질의어를 쉽게 변환할 수 있다. 생성된 질의어는 SQL일수도 있고, OQL일 수도 있으며, 웹 URL, 프로그램실행명령어 등이 될 수 있다.

정보원으로부터 질의결과를 받으면 결과문서변환기는 사용자가 요구한 XML 문서 양식에 맞도록 XML 문서를 생성한다. 사용자가 요구한 XML 문서 양식은 이미 질의트리에 반영되었기 때문에 질의트리가 XML 문서 생성을 위한 템플릿 역할을 한다. 실제로 결과문서변환기는 질의어 합성, 변환, 실행 등을 시키는 드라이버 역할을 한다. 결과문서는 관계형테이블일 수도 있고, 객체일 수 있으며, XML 문서일 수 도 있고, 플랫파일일 수도 있다. 현재 각 데이터 타입에 대하여 모두 구현되어 있는데 각각에 대한 상세한 알고리즘은 생략한다.

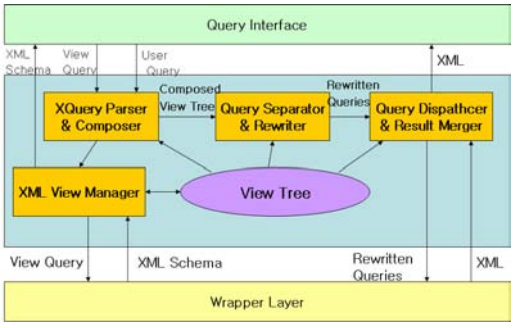
3.3 미디에이터

미디에이터는 (그림 3)과 같이 크게 XML 뷰 관리기(XML View Manager), XQuery 파서 및 합성기, 질의분해 및 재작성기(Query Separator & Rewriter), 결과생성기(Query Dispatcher & Result Merger)로 구성된다. 각 모듈의 상세한 내부구조는 랩퍼와 유사한 부분이 많다. XML 뷰 관리기는 랩퍼로부터 받은 XML Schema를 통합해서 전역 뷰를 구성한다. 전역 뷰를 구성하는 방법으로서 본 시스템은 지역 스키마를 그대로 통합하되 각 스키마의 원천을 보관한다. 또한 미디에이터도 랩퍼와 마찬가지로 사용자정의 XML 뷰를 지원하기 때문에 전역 뷰를 관리하는 모델은 랩퍼와 유사하게 XML 뷰 트리로 관리되며, 랩퍼 스키마 관리기의 기능이 대부분 지원된다.

XQuery 파서 및 합성기는 사용자 질의를 파싱해서 질의에 대한 유효성검사를 하게 되고, 질의에 문제가 있을 시에는 사용자가 정확한 질의를 할 수 있도록 도와준다. 질의합성기는 랩퍼합성기와 완전히 동일하다.

합성이 끝나면 랩퍼에 대한 질의로 분해하는데, 이때 분해는 질의트리의 합성에 이용된 XML 뷰 별로 나누어진다. 즉, 사용자가 작성한 통합 질의는 각각의 뷰 트리에 대한 질의로 나누어지며 이를 새로운 각각의 질의 트리로 재구성한다. 질의 재작성기는 분해된 질의트리로부터 각 랩퍼로 보낼 XQuery 문으로 재작성한다.

재작성된 질의어는 질의분배기(Query Dispatcher)에 의해서 각 랩퍼에게 전달된다. 그리고 결과통합기는 각 랩퍼에서 수행된 결과를 통합하여 사용자 질의에 맞게 XML 문서를 재구성하여 사용자에게 결과문서를 보여주는데 한 랩퍼에서 나온 결과가 다른 랩퍼의 결과에 영향을 미치는 경우도 함께 처리한다.



(그림 3) 미디에이터 시스템 구조

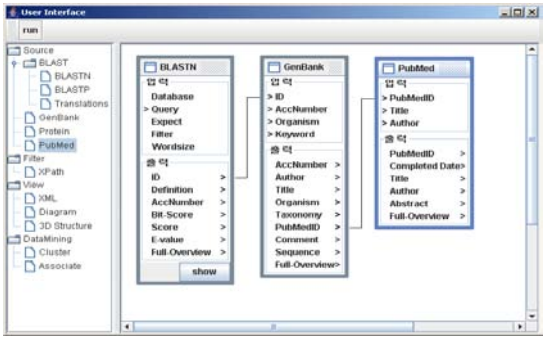
3.4 데이터마이닝 도구의 통합

통합시스템을 데이터마이닝 도구와 연동하기 위하여 뉴질랜드 Waikato 대학에서 개발한 데이터마이닝 도구 WEKA를 활용한다. WEKA는 기계학습 알고리즘과 데이터마이닝 알고리즘을 구현한 JAVA 클래스 라이브러리로 구성되어 있으며, 각자의 시스템 환경과 목적에 맞게 이용할 수 있도록 그 소스를 공개하고 있다. WEKA를 통합시스템에 연동하기 위해서 미디에이터는 WEKA에게 XML 스키마를 트리구조로 제공하고, 질의방법은 일반 미디에이터 사용자와 동일하다. WEKA 질의인터페이스는 질의를 XQuery로 변환하여 미디에이터에게 전달한다. 미디에이터는 질의결과로써 WEKA에게 XML 문서를 전달하고, 이를 (그림 1)에서 보는 바와 같이 XMLToARFF 모듈이 받아서 WEKA가 처리할 수 있는 형식인 ARFF(Attribute-Relation File Format) 형식으로 변환하여 데이터마이닝 모듈에 전달한다.

3.5 질의 인터페이스

많은 분야에서 XML이 정보교환의 중요한 수단으로 활용되면서 XML 데이터에 대하여 질의하고 결과를 제시하는 도구개발이 점점 중요해지고 있다. 본 시스템은 두 가지 형태의 질의인터페이스를 제공한다. 첫째는 통합시스템의 스키마를 트리구조로 제시하고, 사용자는 그래픽 도구로써 드래그 앤 드랍 형식으로 원하는 문서의 구조와 속성을 계층구조를 가진 다이어그램을 작성하면 질의인터페이스는 XQuery로 표현된 통합질의를 생성하여 미디에이터에게 전달하는 것이다. 두 번째 형태의 인터페이스는 생물정보원 특성을 고려한 것이다. 생물학자들은 하나의 정보원으로부터 얻은 검색결과의 일부를 바탕으로 다른 정보원에 접속하여 새로운 검색결과를 얻는 일을 반복한다. 이를 편리하게 지원하기 위하여 정보원과 정보원 사이의 항목을 연결하여 새로운 질의를 구성하도록 인터페이스를 설계 구현 하였다. 예를 들어, (그림 4)는 BLAST를 실행시켜서 유전자 서열의 상동성을 검사한 다음, 검색된 유전자 중에서 관심 있는 유전자의 특성을 파악하기 위하여 GeneBank Id를 추출하여 GeneBank에 접근하고, 계속해서 GeneBank로부터 얻은 결과를 토대로 관련된 문헌정보를 얻기 위하여 PubMed에 접근하는 경우를 보인 것이다.

사용자는 왼쪽 프래임으로부터 원하는 정보원인 BLAST, GeneBank, PubMed을 선택하면 오른쪽 프래임에 자동으로 그 정보원의 입력자료(윗부분)와 출력자료(아랫부분)이 제시된다. 계속해서 사용자는 한 정보원의 출력자료 항목을 다른 정보원의 입력자료로 보내기 위하여 해당 항목을 마우스로 드래그한다. 그 후에 실행을 시키면 XQuery가 생성되어서 전달된다. 검색결과는 최종 정보원에 대한 요구를 제시하지만, 중간 단계의 정보원이 보내 준 검색결과도 XML 문서로 확인할 수 있다.



(그림 4) 사용자 질의 인터페이스

4. 결론

생물정보원은 이질성이 높고 연구자에 따라서 다양한 요구사항이 있다. 본 논문은 사용자의 다양한 요구에 대한 해결책으로 개별화된 통합시스템을 쉽게 구축할 수 있도록 사용자정의 XML 뷰 기반의 뷰 관리모델, 질의처리, 사용자인터페이스를 설계 구현한 결과를 제시하였다. 또한 관계형 테이블, 객체, XML문서, 플랫파일 등 다양한 데이터 타입을 지원하며, 관계형모델, 웹 자원, 응용프로그램, 데이터마이닝 도구를 쉽게 통합하는 특정모델에 종속적이지 않은 통합시스템 구조를 제시하였다.

현재 실험대상으로 구현된 정보원은 임상정보와 프로테옴 정보를 관리하는 관계형모델, GeneBank, PubMed 등 NCBI에서 제공하는 대부분의 정보원과 단백질의 3차원 구조에 대한 정보를 제공하는 PDB와 같은 웹 정보원, 서열분석도구 BLAST, 그리고 데이터마이닝 도구 WEKA이다. 향후, 온톨로지 기반의 의미기반 통합시스템 구축, 미디에이터의 효율적인 질의처리 알고리즘, 그리고 자동화된 질의인터페이스 개발 등에 대한 연구가 더 필요하다.

참고 문헌

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

[3] 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.

[4] A. S. Kosky, I. A. Chen, V. M. Markowitz, E. Szeto, "Exploring Heterogeneous Biological Database: Tools and Application," Proc. of the 6th International Conference on Extending Database Technology, pp3-5, 1998.

[5] 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

[6] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan, J. Funderburk, "Querying XML Views of Relational Data", VLDB Conference, pp.261-270. 2001

[7] 박은경, 강동완, 정채영, 배종민, "생물 정보원에 대한 통합 접근을 위한 랩퍼 모델", 정보처리학회논문지 D, 제11-D권, 제4호, pp.768-772, August, 2004