SCOPML과 SCOPBrowser에 관한 연구

안 건 태[†] · 윤 형 석[†] · 황 의 윤[†] · 김 진 홍[††] · 이 명 준[†††]

요 약

*포스트지놈 시대*에 있어서 가장 주된 연구는 단백질의 구조적 유사성이나 분류학적인 연관성을 밝히는 것이다. *SCOP 단백질 구조 분류*는 이러한 목적을 위한 대표적인 데이터베이스로서, 3차원 구조가 알려진 단백질에 대한 구조적 분류학적 관계에 대한 상세한 기술을 제공한다. 하지만, SCOP 데이터는 단순 텍스트 형식의 자료로만 제공되고 있어서 이를 이용한 다른 분석 도구나 자원을 개발할 경우 그 작업이 번거로우면서도 오류 발생의 소지가 높다. 따라서 이러한 데이터를 연구자들이 보다 효과적으로 이용할 수 있도록 표준화된 구조적인 형식으로 제공하는 것이 바람직하다. 이러한 요구를 충족시키기 위하여, 본 논문에서는 SCOP 데이터의 XML 표현인 SCOPML과 SCOP 데이터를 해당 SCOPML 문서로 변환하기 위한 변환기를 개발하였고, 또한 SCOP 데이터베이스에 대한 효율적인 검색을 지원하는 브라우징 도구인 SCOPBrowser를 구현하였다. SCOPBrowser는 SCOP 사이트에서 제공되는 기본정보 및 단백질 구조 분류 정보에 대한 트리보기, 전체 단백질 도메인에 대한 검색, 특정 도메인에 대한 XML 내용 보기, 그리고 단백질 구조에 대한 유용한 통계 등 다양한 정보를 얻을 수 있다.

SCOPML and SCOPBrowser

Geon-Tae Ahn[†] · Hyeong-Seok Yoon[†] · Eui-Yoon Hwang[†]
Jin-Hong Kim[††] · Myung-Joon Lee[†††]

ABSTRACT

The major challenge for *post-genomic* study is to identify structural similarity and relationships of proteins. *SCOP* (*Structural Classification Of Proteins*) is a typical database for this purpose, providing a detailed description of the structural and functional relationships of the proteins whose three-dimensional structures have been determined. Unfortunately, since the SCOP data is only available as a plain text format, it is cumbersome and error-prone to develop tools and resources for the different kinds of analysis of the data. Therefore it is desirable to provide the data in a standard structured format, allowing researchers to utilize the data more effectively. To meet these requirements, we have developed an XML representation for the SCOP data, named SCOPML and the translator to convert the SCOP data into the associated SCOPML documents. Also, we have implemented a browsing tool, named *SCOPBrowser*, for effective search of SCOP database. In addition to the information available from the SCOP site, users of the tool can obtain various information such as viewing the tree hierarchy of structure classification of proteins, searching into whole protein domains, showing XML contents of a specific domain, and some useful statistics about protein structures.

키워드 : SCOPML, SCOPBrowser, 단백질 도메인(Protein Domain), SCOP, XML

1. 서 론

분자 생물학 기술의 발달과 *인간유전체사업(Human Genome Project)*의 완성은 대량의 생물 분자 데이터 및 새로운 형태의 생물학 정보들을 산출하였다. 이와 더불어 이러한 정보들을 저장하고 관리하기 위한 다양한 형태의 데이터베이스들과 시스템들이 생겨나게 되었다. 특히, 인간 유전체 지도의 완성으로 포스트지놈 시대가 열리게 되면서 이제는 유전체 정보의 최종산물인 단백질의 구조와 기능을 규명하는 연구에 관심이 집중되고 있다[1]. 단백질은 대부분 분류학적으로 공통의 기원을 가지거나 구조적으로 유사한 경우가 많다. 단백질 사이의 구조와 분류학적인 관계 정보는 인간유전체사업의 결과 생성된 대량의 서열정보를 번역하고 단백질의 기능을 밝혀내는 데 중요한 역할을 담당할 것이다. 이러한 구조적인 정보를 이해하고 접근할 수 있도록 하기 위하여 *SCOP 단백질 구조 분류 데이터베이스*[2]가 구축되었다. SCOP은 단백질 도메인 정보를 기반으로 3차 구조가 결정된 단백질들을 구조적이고 분류학적인 관계에 따라 분류하고 있다. 이러한 단백질 구조 분류 정보는 단백질 3차 구조에 내포된 규칙이나 단백질 설계상의 관계를 추론하는 데에도 활용될 수 있다. 현재 SCOP 데이터베이스는 단백질 분류 데이터들을 단순 텍스트 형태로 제공하고 있으며 HTML 기반으

※ 이 논문은 2002년 울산대학교의 연구비에 의하여 연구되었음.
† 준 회 원 : 울산대학교 컴퓨터정보통신공학부
†† 준 회 원 : 울산대학교 컴퓨터정보통신공학부
††† 정 회 원 : 울산대학교 컴퓨터정보통신공학부 교수
논문접수 : 2002년 9월 2일, 심사완료 : 2002년 11월 18일

로 웹을 통하여 서비스하고 있다. 이러한 분류정보에 대한 연구는 생물학에 있어서 중요한 의미를 가지지만 자료의 제공형식이 텍스트 파일이고 검색 또한 HTML 기반이어서 기존 데이터를 활용하려고 하는 경우 데이터 가공의 번거로움이 많으며 정형화되지 않아 문서의 파싱시 에러가능성을 내포하고 있다. 이러한 문제를 해결하기 위하여 단백질 구조 분류 정보를 보다 효과적으로 표현하고 교환하기 위한 접근 방법이 요구되며, XML(eXtensible Markup Language)[3, 4]은 이러한 문제를 해결하기 위한 이상적인 해결책을 제시해준다. XML 문서는 문법적인 구별이 가능해 유효성 검사를 통하여 문서상의 오류를 쉽게 판단할 수 있고 문서의 내용과 표현을 기본적으로 분리하고 있어 다양한 표현 방식이 가능하도록 지원하므로 재사용적인 측면에서도 상당한 장점을 가진다.

본 논문에서는 단백질 구조 도메인 정보를 효과적으로 활용할 수 있는 방안으로 단순 텍스트 기반의 SCOP 데이터를 XML 문서로 변경하기 위한 언어인 *SCOPML*을 설계하고 구현하였다. 추가로, SCOPML의 응용 시스템으로 SCOP 정보를 보다 효율적이고 용이하게 검색할 수 있는 뷰어 프로그램인 *SCOPBrowser*를 제공함으로써 단백질 도메인 정보에 대한 브라우징 기능, 단백질 구조 도메인간의 연관성 분석 기능, 그리고 구조 분류 기반의 도메인 통계정보추출 기능 등 다양한 추가 기능을 관련 연구에 활용할 수 있도록 지원하였다. SCOPML은 SCOP의 구조 분류를 *XML DTD* (*Document Type Definition*)로 정의하여 SCOP데이터 구조가 가진 정보를 구조화된 문서로 나타내어 질 수 있도록 한다. SCOPBrowser는 SCOPML에 의하여 생성된 XML 데이터를 이용하여 SCOP 기반 단백질 분류 구조를 트리형식으로 볼 수 있는 기능을 제공한다. 이처럼 SCOPBrowser를 이용하는 경우 SCOP 사이트[5]보다 빠르고 효율적인 도메인 검색이 가능하며, 시스템이 제공하는 추가적인 기능을 이용하는 경우 관련 연구 진행에 상당한 도움을 줄 것으로 기대된다.

본 논문의 구성은 다음과 같다. 1장 서론에 이어 2장에서는 XML 기반의 생물정보 관련 연구에 대하여 기술한다. 3장에서는 단백질 구조 분류 데이터베이스인 SCOP의 XML 표현인 SCOPML에 대하여 설명한다. 4장에서는 SCOPML을 이용한 단백질 구조 분류 탐색기인 SCOPBrowser의 구현에 대하여 기술한다. 마지막으로 5장에서는 결론과 향후과제에 대하여 기술한다.

2. XML과 생물 정보

XML은 웹상에서 구조화된 문서를 전송하도록 설계된 표준 마크업 언어이다. 인터넷이 대중화되고 정보의 양이 급속하게 증가함에 따라 정보를 표준화된 형태로 저장할 필요성이 증대되었다. 생물정보학 분야에서도 생물정보 데이터베이스나 데이터의 교환을 위하여 XML 기술을 이용한 연구가 활발해지고 있다[6]. 대표적인 연구로는 유전자 정보에 대한 XML 표준 스펙인 *BSML*(*Bioinformatic Sequence Markup Language*)[7], 분자 구조 기술을 위한 언어로서 단백질 명세를 확장하기 위한 이상적인 기초를 제공하는 *CML*(*Chemical Markup Language*)[8], 단백질 서열, 구조, 패밀리(families)등에 관한 명세언어인 *ProML*(*Protein Markup Language*)[9], 그리고 온톨로지(ontology) 기반의 객체 메타모델인 *OpenMMS*(*Open Macromolecular Structure*)[10] 등이 있다.

2.1 BSML

BSML은 생물정보 데이터에 대한 언어 스펙 및 컨테이너를 지원한다. 이 마크업 언어는 SGML과 XML을 기반으로 DNA 정보를 인코딩하거나 보여주기 위한 방법을 제시하고 있다. XML 표준을 따르는 이 언어의 궁극적인 목적은 유전자 서열의 특징이나 이러한 특징을 표현하기 위해 사용되는 비주얼 객체 정보를 기술할 수 있도록 지원하며, 또한 서열정보 및 그래픽 정보를 저장하고 전송하는 방법을 제공하는 것이다. 따라서 BSML 형식을 이용하여 작성된 문서는 내용이 내포하는 생물적인 의미와 구성요소들 사이의 관계 등을 체계적으로 잘 표현할 수 있게 된다. 또한, 연구 생산물과 공개 생물데이터베이스와의 통합작업이 용이하게 되어 생물정보 데이터의 *상호운용성*(*interoperability*), 교환 및 관리를 효과적으로 지원할 수 있다. 그러나 BSML은 유전자 정보를 중심으로 설계되어 단백질의 도메인 정보나 *구조 모티프*(*structural motif*) 같은 정보에 대한 표현은 지원하지 않는다.

2.2 CML

CML은 XML과 JAVA 기술을 이용하여 분자구조를 기술하는 언어이다. CML은 고분자 서열에서부터 무기화합물 및 양자화학의 연구에 이르기까지 광범위하게 이용되고 있다. 이 언어는 분자관련 문서에 포함된 다양한 이산 객체 정보를 완벽하게 처리하고 단백질 명세를 확장하기 위한 이상적인 기초를 제공한다. CML 파일에는 화학적 MIME 유형과 같은 특정 파일을 포함할 수 있다. 따라서 단백질에 대한 하나의 CML 문서에는 하이퍼텍스트와 함께 *PDB*(*Protein Data Bank*)[11]와 *SWISS-PROT*[12] 파일들을 포함시킬 수 있는 장점이 있다. CML은 단백질 분자의 물리화학적인 구조를 표현하기엔 적합하지만 주석처리나 서열관련 데이터 및 SCOP과 같은 구조적 분류 데이터들에 대한 정보를 표현하기가 어렵다.

2.3 ProML

ProML은 단백질 *서열*, *구조*, *패밀리(families)* 등에 관한 명세 언어이다. 이 언어는 단백질 기본 정보를 표현하는데 있어서 *이식성(portability)*이 강하고 시스템 독립적이며, 기계적 파싱이 가능하여 *가독성(readability)*이 높은 특징을 가진다. ProML은 단백질을 이루는 속성들을 하위레벨의 또 다른 속성들로 상세 정의해 놓아서 단백질들을 패밀리별로 그룹화하고 각각의 패밀리들이 내포한 공통적인 특성들을 표현하는 데에 성공적으로 적용되어 왔다. 특히, ProML은 *쓰레딩(threading)*이나 그룹화에 사용되는 단백질 속성들을 잘 표현해 준다. 이들 속성에는 아미노산 서열, *PROSITE* 패턴[13], *CATH* 구조분류[14], 이차구조 구성요소(나선, 판상조각, 루프), 삼차구조 정보(3차원 좌표), 그리고 *이황화 결합* 정보 등이 포함된다. ProML은 PDB 데이터를 ProML 포맷으로 변경할 수 있도록 웹 기반 변환기를 제공하고 있다.

2.4 OpenMMS

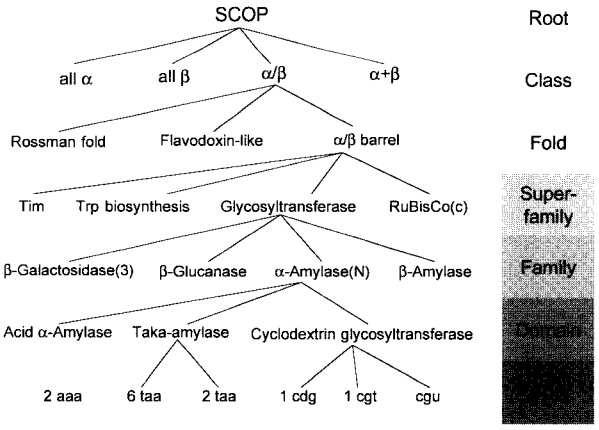
OpenMMS는 *RCSB(Research Collaboratory for Structural Bioinformatics)* 컨소시엄이 단백질 관련 연구자들을 위하여 발표한 객체 메타모델로서 PDB 정보를 *CORBA 스펙*을 이용하여 보다 효과적으로 이용할 수 있도록 설계한 것이다. PDB는 단백질 구조 정보를 저장하고 있는 대표적인 데이터베이스로서 *X-Ray*나 *NMR(Nuclear Magnetic Resonance)*과 같은 실험적인 구조 규명 기법에 의하여 밝혀진 공간적인 3차원 구조 정보의 제공 및 가공을 담고 있다. 표준 온톨로지 기반의 이 객체모델은 *고분자 구조(Macromolecular Structure)* 데이터에 대하여 CORBA 인터페이스, SQL 스키마, 그리고 XML 표현기법을 지원한다. 또한, 이 메타모델을 이용할 경우 CORBA 서버나 JDBC 데이터베이스 로더에 대한 핵심 구성요소들을 생성할 수 있다. OpenMMS의 CORBA 인터페이스는 PDB에서 제공하는 고분자 정보나 상세 실험정보에 원거리에서 직접 접근할 수 있는 프로그래밍 인터페이스를 제공한다. 이처럼 OpenMMS는 고분자 데이터에 대한 CORBA, SQL, 및 XML 표현들을 제공함으로써 PDB에서 제공되는 분자 구조 정보를 다양한 표준 방식으로 접근할 수 있도록 지원하고 있지만, 단백질 구조에 대한 전체적인 정보를 포함하고 있어서, 도메인 연구에 활용시 추가적인 작업이 필요하며, 특히, 단백질 구조 분류에 대한 정보가 누락되어 있어서 다른 데이터베이스와의 연동 작업이 필요하다.

3. SCOPML(Structural Classification of Proteins Markup Language)

SCOPML은 구조적인 웹 문서 표준인 XML 기술을 이용하여 SCOP 데이터를 기술하기 위한 마크업 언어이다. SCOP 데이터베이스가 표현하는 단백질 구조 분류를 XML로 표현하기 위하여 SCOPML DTD를 정의하였다. SCOPML DTD는 SCOP 데이터베이스가 제공하는 단백질의 계층적인 분류를 효과적으로 기술하고 있으며, 이를 이용하여 생성된 SCOPML 문서를 이용할 경우 단백질 구조 분류와 관련된 응용프로그램들을 보다 용이하게 개발할 수 있다.

3.1 SCOP 데이터베이스

SCOP은 단백질이 지닌 구조적인 유사성과 분류학적인 관계를 기반으로 단백질들을 체계적으로 분류해 놓은 데이터베이스이다. SCOP에는 이미 구조가 밝혀진 모든 단백질들에 대한 자료가 저장되어 있어 미지의 단백질에 대한 구조를 밝히거나 기능을 예측하는 연구에 많이 활용되고 있다. (그림 1)에서 보는 바와 같이, SCOP의 계층적인 구조 분류는 분류 기준에 따라 11개의 *클래스(class)*로 나뉘어지며, 각각의 클래스는 2차 구조의 구성과 *토폴로지(topology)*에 의해 *폴드(fold)*로 나뉘어 진다. 폴드는 다시 *수퍼패밀리(superfamily)*로, 수퍼패밀리는 *패밀리(family)*로, 그리고 패밀리는 *도메인(domain)*으로 나뉘어 진다. 이처럼 PDB의 모든 단백질은 다른 단백질들과 비교되어 *구조적 유사성(structural similarities)*을 가지는 그룹으로 분류된다.



(그림 1) 단백질 구조 분류 데이터베이스인 SCOP 계층구조의 예

SCOP이 단백질 연구에 있어서 중요한 역할을 담당하고는 있지만, 사실, SCOP이 제공하는 데이터들은 그들 고유의 형식에 의하여 정형화된 명세없이 단순 텍스트 파일로 저장되어 있어서 데이터의 활용 및 가공에 어려움이 있다. 따라서 데이터베이스로부터 원하는 자료를 효과적으로 검색하고 활용할 수 있는 방안이 필요하다.

3.2 SCOPML DTD

SCOPML 문서는 SCOP 데이터베이스에서 나타내고 있는 계층구조를 Element로 표현하였고, 각각의 하위 Element

를 식별하기 위해 sunid 혹은 sccs라는 속성 값을 이용하여 구분하였다. sunid는 SCOP에서 계층구조 분류에 대한 식별자이며, sccs는 SCOP 계층구조에 대한 간략한 표현이다(예 : PDB 엔트리 1g61, 체인 A의 sccs는 d.126.1.1).

(그림 2)는 SCOPML DTD의 구조를 나타낸다.

```
< !ELEMENT root (class + ) >

< !ELEMENT class ( # PCDATA | fold ) * >
< !ATTLIST class sunid ID # REQUIRED sccs CDATA
  # REQUIRED >

< !ELEMENT fold ( # PCDATA | superfamily ) * >
< !ATTLIST fold sunid ID # REQUIRED sccs CDATA
  # REQUIRED >

< !ELEMENT superfamily ( # PCDATA | family ) * >
< !ATTLIST superfamily sunid ID # REQUIRED sccs
  CDATA # REQUIRED >

< !ELEMENT family ( # PCDATA | protein ) * >
< !ATTLIST family sunid ID # REQUIRED sccs CDATA
  # REQUIRED >

< !ELEMENT protein ( # PCDATA | species ) * >
< !ATTLIST protein sunid ID # REQUIRED sccs CDATA
  # REQUIRED >

< !ELEMENT species ( # PCDATA|domain ) * >
< !ATTLIST species sunid ID # REQUIRED sccs CDATA
  # REQUIRED >

< !ELEMENT domain (scopid, pdbid, pdbregion ) >
< !ATTLIST domain sunid ID # REQUIRED sccs CDATA
  # REQUIRED >

< !ELEMENT scopid ( # PCDATA ) >

< !ELEMENT pdbid ( # PCDATA ) >

< !ELEMENT pdbregion ( # PCDATA ) >
```

(그림 2) SCOPML DTD 구조

- **ELEMENT**

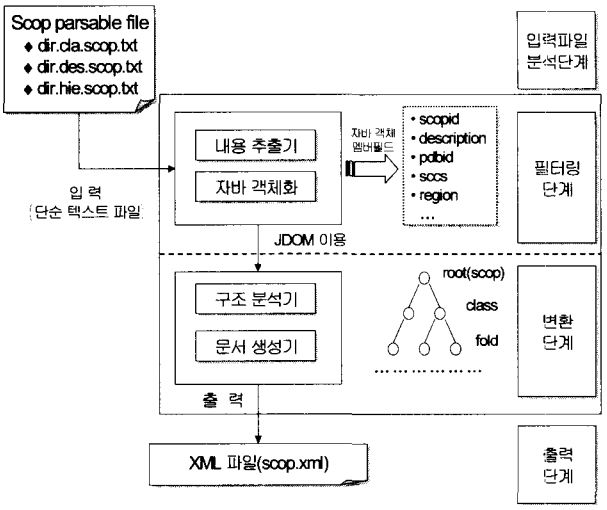
XML 문서의 논리적 구조를 표현하는 기본 단위로서 모든 문서는 이 구성요소의 트리 형태로 구성된다. SCOPML에서는 SCOP의 7가지 하위분류와 4개의 PDB 관련 정보(root, class, fold, superfamily, family, protein, species, domain, scopid, pdbid, pdbregion)가 ELEMENT로 기술된다.

- **Attribute**

ELEMENT의 추가정보를 기술하기 위한 단위이다. SCOPML에서는 각각의 ELEMENT 사이의 구별을 위하여 SCOP 분류계층의 식별자인 sunid와 그룹 정보를 나타내는 sccs를 이 Attribute에 기술하였다.

3.3 SCOPML 변환기

SCOPML 변환기는 비구조적인 문서를 구조적인 문서인 XML 문서로 변환하기 위한 프로그램이다. 이 변환기는 크게 입력파일 분석부분, 데이터 필터링 부분, XML 변환 부분, 그리고 XML 문서파일 생성 부분으로 구성된다. 입력파일 분석 단계에서는 SCOP의 텍스트 파일을 입력받아 내용 정보를 추출하고, 추출된 정보는 필터링 단계에서 자바 객체로 저장된다. XML 변환 단계에서는 필터링 단계에서 생성한 자바 객체로부터 XML을 위한 자바 API인 *JDOM(JAVA DOM)* [15]을 이용하여 XML로 변환하게 되며 최종 단계에서 XML 문서를 생성하게 된다. SCOPML 변환 시스템의 전체 구성은 (그림 3)과 같다.



(그림 3) SCOPML 변환기 처리과정

3.3.1 입력파일 분석 단계

SCOP 데이터베이스는 자신의 계통 분류학적 정보를 다음과 같은 3개의 파일(dir.class.scop.txt, dir.des.scop.txt, dir.hie.scop.txt)로 저장하여 제공한다. 입력파일 분석 단계에서는 이들 파일들을 분석하여 어떤 데이터를 이용할 것인지를 결정하게 된다. SCOP에서 제공하는 파일들을 간단히 살펴보면 다음과 같다. 먼저, dir.cla.scop.txt 파일은 SCOP 분류의 기본 단위인 도메인에 대한 전체 분류 계층구조에 대한 정보를 저장하고 있다. (그림 4)는 dir.cla.scop.txt 파일의 내부 구성을 보여준다.

```
# dir.cla.scop.txt
# SCOP release 1.57 (January 2002) [File format version 1.00]
.....①

d1dlwa_ 1dlw  A : a.1.1.1 14982 .....②
        cl = 46456, cf = 46457, sf = 46458, fa = 46459, dm = 46460,
        sp = 46461, px = 14982 .....③
• • •
```

(그림 4) dir.cla.scop.txt 파일 내용

① # : file header를 표시함. 파일 이름, Release Version, SCOP Server주소, Copyright 정보를 나타낸다.

② a.1.1.1 14982

- sccs(a.1.1.1) : class, fold, superfaminly, family 정보를

하나의 문자열로 간략하게 표현한다.

- sunid(14982) : 계층 분류를 구분하기 위한 식별자이다

③ 나머지 부분

- class(cl), fold(cf), superfamily(sf), family(fa), protein (dm), species(sp), domain(px) 등 각각의 분류에 해당하는 sunid이다.

(그림 5)의 dir.des.scop.txt 파일은 각 계층 분류에 해당하는 설명정보를 가지고 있다. 실제 각각의 계층 분류가 가지는 의미를 이 파일을 통하여 참조하게 된다.

```
# dir.des.scop.txt
# SCOP release 1.57 (January 2002) [File format version 1.00]

14982   px   a.1.1.1   d1dlwa_   1dlw A :
46456   cl   a         -         All alpha proteins
46457   cf   a.1       -         Globin-like
46458   sf   a.1.1     -         Globin-like
46459   fa   a.1.1.1   -         Truncated hemoglobin
46460   dm   a.1.1.1   -         Truncated hemoglobin
46461   sp   a.1.1.1   -         Ciliate (Paramecium caudatum)
• • •
```

(그림 5) dir.des.scop.txt 파일 내용

dir.hie.scop.txt에는 (그림 6)에서 보는 바와 같이 SCOP 계층구조를 sunid를 이용하여 표현하고 있다. 계층 구조상 최상위는 0으로 표시되며 두 번째 필드의 sunid는 첫 번째 필드 sunid의 부모이고, 세 번째 필드 sunid는 첫 번째 sunid의 자식들이 나열된다(예 : ①번 행이 나타내고 있는 정보는 sunid가 46460인 계층 분류는 부모의 sunid가 46459이고 46461, 46462, 그리고 63437의 3개의 자식 분류를 가지고 있다).

```
# dir.hie.scop.txt
# SCOP release 1.57 (January 2002) [File format version 1.00]

0       -       46456, 48724, 51349, 53931, 56572, 56835, 56992,
                57942, 58117, 58231, 58788
14982   46461   -
46461   46460   14982
46460   46459   46461, 46462, 63437 ..... ①
46459   46458   46460
46458   46457   46459, 46463, 46532
46457   46456   46458, 46548
46456   0       46457, 46556, 46625, 46688, 46928, 46954, 46965,
                46996, 47004, 47013,
• • •
```

(그림 6) dir.hie.scop.txt 파일 구성 내용

3.3.2 필터링 단계

입력파일 분석 결과를 바탕으로 필터링 단계에서는 SCOP의 모든 정보를 XML 문서로 나타내기 위하여 각각의 파일들을 참조하게 된다. 구조정보는 dir.cla.scop.txt로부터 얻어오며, dir.des.scop.txt는 DTD에 정의된 형식에 따라 생성 태그들의 내용정보(구조 분류에 대한 설정정보)를 제공한다. 각각의 파일정보를 자바 객체의 멤버필드로 표현하여 단백질 구조 분류 정보를 저장할 객체를 생성하고, 이를 ArrayList 자료구조에 임시 저장한다.

3.3.3 XML 변환 단계

필터링 단계를 통하여 생성된 자바 객체를 JDOM 인터페이스를 이용하여 처리한다. DTD에 정의된 Element와 Attribute에 따라 내부구조를 생성하고 내용정보를 추가하여 XML 문서를 생성한다. 각각의 정보는 자바 객체로부터 얻어오게 되며, "scop"이라는 이름으로 최상위 태그를 생성하였다. Element에 대한 속성은 SCOP 데이터베이스에서 나타내는 유일한 값인 sunid와 sccs를 가진다. (그림 7)은 생성

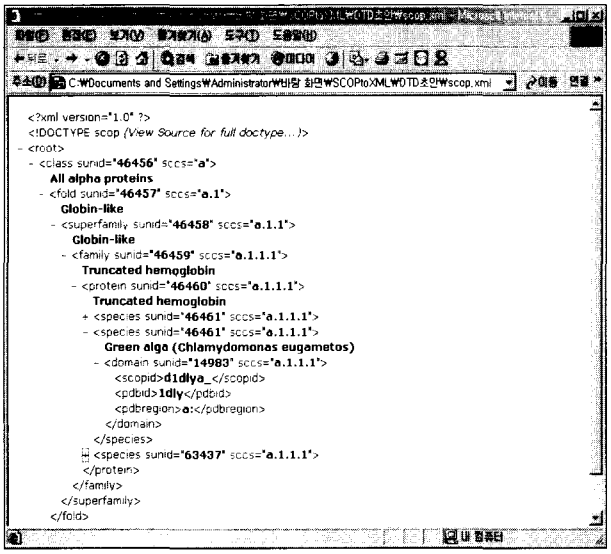
```
public class ScopToXML {
  ArrayList cla, des ;
  String xmlFileName ;
  boolean empty = true ;
  public ScopToXML (ArrayLIst cla, ArrayList des) { // 생성자
    this.cla = cla ;                          // ArrayList를 초기화
    this.des = des ;
    xmlFileName = "scopml.xml" ;              // convertToXML 메소
                                                 드 호출
    convertToXML( ) ;                         // 실제 XML 파일을
                                                 생성하는 메소드
  }
  public void convertToXML( ) {               // 결과 XML 생성 메소드
    Element root = new Element ("scop") ; // Root Element를 생성
    Document scopml = new Document (root) ; // Document 객체
                                              를 생성
    // cla의 길이만큼 loop
    for (int i = 0 ; i < cla.size ( ) ; i++ ) {
    // cla에서 한개의 Object를 가져온다 (Dir_cla_scop Type)
    Dir_cla_scop s = (Dir_cla_scop) cla.get (i) ;
    List children = root.getChildren ("class") ; // Root Element의
                                                   자식을 구함.
    ..... (중략)
  try {
    // 주어진 파일로 문서를 출력한다.
    XMLOutputter outputter = new XMLOutputter (" ", true) ;
    FileOutputStream output = new FileOutputStream (xml
                                        FileName) ;
    outputter.output (scopml, output) ;
    } catch (Exception e) { }
  }
  // 태그를 생성하기 위한 메소드
  public void tagGeneration (Dir_cla_scop s, String level,
        Element root ) { }
  // 생성된 태그에 내용 정보(구조설명)를 넣기 위한 메소드
  public String getEngDescription (String sunid ) { }
  // SCCS를 얻어오기 위한 메소드
  public String getSccs (String sunid ) { }
  // 이미 생성되어 있는 태그일 경우 그 자식 Element에 대한 태그
     생성
  public void childToXML (Dir_cla_scop s, Element cl,
        String level ) { }
}
```

(그림 7) SCOPML 문서를 생성하는 클래스

된 자바 객체로부터 XML로 변환하는 클래스의 일부를 나타낸다.

3.3.4 XML 파일 생성 단계

(그림 8)은 SCOPML 변환 시스템이 생성한 scop.xml의 일부를 웹 브라우저를 통해 나타낸 것이다. 이 문서는 SCOP ID가 'd1dlya_'인 정보를 나타내고 있는데, class는 All alpha proteins, fold는 Globin-like, superfamliy는 Globin-like, family는 Truncated hemoglobin, protein은 Truncated hemoglobin, 그리고 species는 Green alga (Chlamydomonas eugnmetos)인 단백질 도메인의 구조를 표현하고 있다.



(그림 8) XML 변환기를 통해 생성한 XML 문서

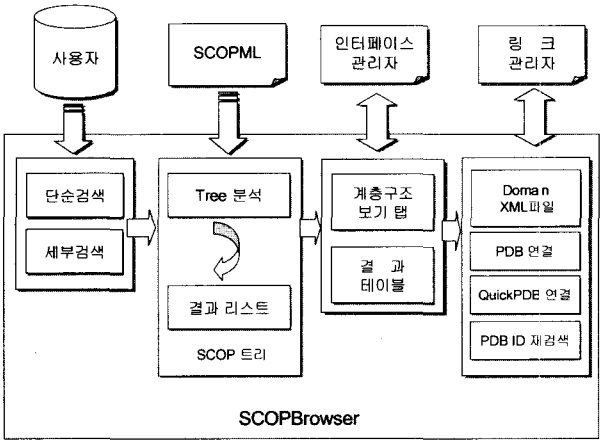
4. SCOPBrowser

SCOPBrowser는 SCOPML의 응용 시스템으로서, SCOPML 문서를 기반으로 SCOP 정보를 효율적으로 검색하고 브라우징할 수 있는 도구이다. 이 시스템은 자바 언어와 JDOM 인터페이스를 이용하여 SCOPML 문서를 분석하여, SCOP 단백질 구조 분류를 트리 구조로 한 눈에 볼 수 있는 기능을 지원한다. 또한, 트리로부터 사용자가 원하는 데이터를 효율적으로 추출하여 보여줄 수 있는 검색기능을 제공한다. 이러한 검색 기능은 SCOP ID나 PDB ID를 이용하여 원하는 정보를 조회하는 단순 검색기능과 사용자가 옵션별로 원하는 세부 정보를 설정할 수 있도록 한 세부검색기능 등 두 가지로 구분된다. 상세 검색의 경우, 사용자는 구조 분류 단계별로 원하는 항목을 선택할 수 있어서 각각의 분류가 가지는 연관성을 추가한 검색연산을 할 수 있는 장점을 가진다. 출력결과는 테이블 형식으로 보여지며, 결과 내 검색도 지원한다.

4.1 SCOPBrowser의 구조

SCOPBrowser는 내부적으로 SCOPML을 이용하여, SCOP 트리를 생성한다. SCOPML이 가지는 모든 데이터를 JDOM을 사용하여 분석한 다음, 트리로 변환하고 트리로부터 모든 기능을 수행한다.

(그림 9)는 SCOPBrowser의 전체적인 구조와 데이터의 흐름을 표현한 것이다. 전체적으로는 SCOP 트리를 중심으로 모든 기능이 수행되어진다. 즉, 사용자로 부터의 검색 요청은 SCOP 트리를 분석하여 결과 리스트를 얻고, 결과 리스트로부터 Table을 생성하게 된다. 또한, 검색한 결과가 나타나는 Table로 부터 다른 부가적인 기능을 수행할 수 있는 구조로 구성되어 있다.

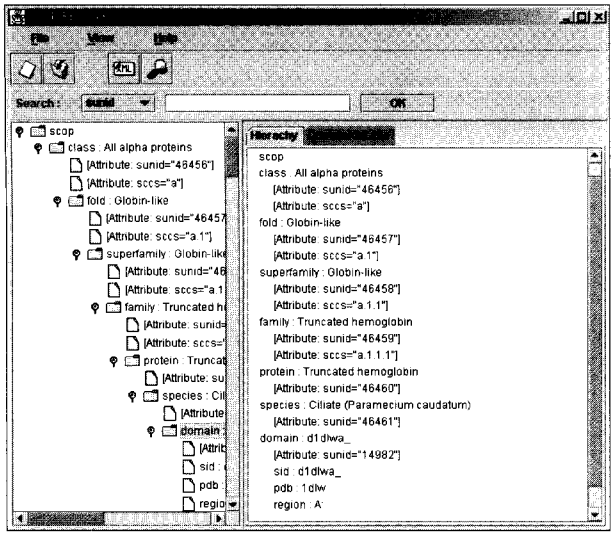


(그림 9) SCOPBrowser 내부 구성

4.2 SCOPBrowser의 구현 및 기능

SCOPBrowser는 SCOPML을 효율적으로 이용하기 위한 도구이다. 따라서 SCOPML의 효과적인 검색에 중점을 두어 개발되었다. JDOM을 사용하여 SCOPML로부터 자바객체인 JTree를 생성해내고, 이 트리로부터 원하는 데이터를 검색할 수 있게 인터페이스를 구현하였다.

SCOPBrowser는 (그림 10)과 같이 초기에는 Menu, Tool Bar, 검색 패널, 트리, Tab 부분으로 구성되어 있다. 메뉴는 Browser에 대한 전체적인 제어를 할 수 있는 File Menu, 검색을 위한 Search 프레임을 표시할 수 있는 기능을 제공하는 View메뉴, 그리고 SCOP Database에 대한 간단한 도움말과 SCOPBrowser의 사용법을 표시하는 Help Menu로 구성된다. 또한, Menu의 목록들에 대해서 한번의 움직임으로 실행을 할 수 있게 각 Menu와 연결된 아이콘이 있는 Tool Bar가 있다. 검색 패널은 단순검색시 이용하는 부분으로 sunid, sccs, scopid, pdbid를 이용하여 SCOP Database를 검색할 수 있다. 트리부분은 SCOPML을 트리형식으로 표현한 것이고, 마지막으로 Tab 부분은 트리에서 선택된 한 노드의 정보를 표시하는 부분이다.

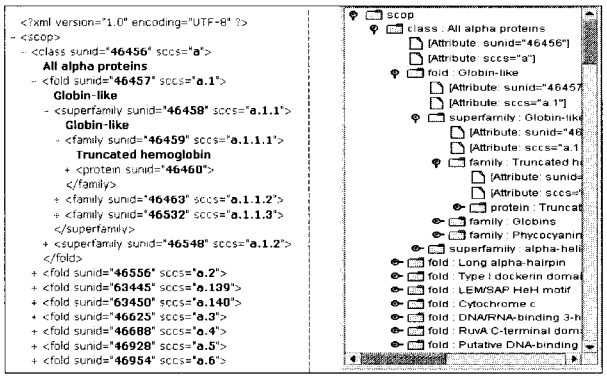


(그림 10) SCOPBrowser 인터페이스

4.2.1 트리 뷰어

SCOPML은 SCOP의 단백질 구조 분류를 계층적으로 담고 있다. 따라서 SCOPBrowser는 SCOPML을 트리형태로 변환시켰고, 생성된 트리로부터 SCOP에서 제공하는 모든 정보를 효과적으로 조회할 수 있도록 하였다.

(그림 11)은 SCOPML이 트리에서 어떻게 표현 되는지를 나타낸 것이다. SCOPML에서 각각의 Element들은 트리에서 노드로 나타나고, *Attrebute*들은 트리에서 해당 노드의 식별자 정보를 담고 있다. 그리고 SCOPML에서는 Text Element를 각 트리 노드에 포함시켜서 노드에 대한 추가 설명을 기술하고 있다.



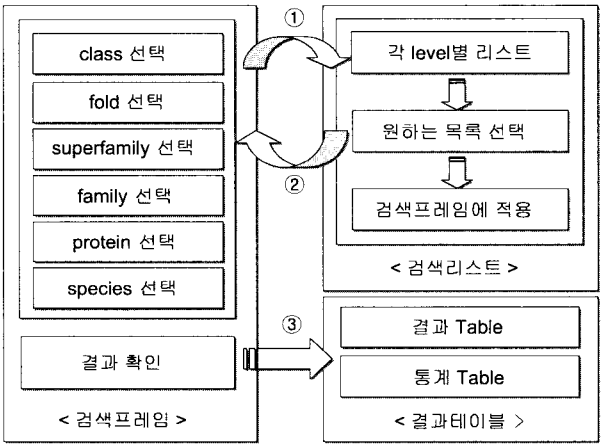
(a) SCOPML Hierarchy (b) SCOP Tree Hierarchy

(그림 11) SCOPML문서 구조와 SCOPBrowser의 트리

4.2.2 검 색

검색은 단순검색과 세부 검색으로 나누어 구현되었다. 단순검색은 간단하게 sunid, sccs, scopid, pdbid를 이용하여 검색하는 것이고, 세부검색은 세부적으로 여러 가지 조건을 통하여 검색할 수 있도록 하여 사용자가 원하는 데이터와 더불어, 결과에 대한 통계적 수치도 제공하고 있다. 이러한 검색 결과에 해당하는 단백질 구조에 대한 계층적인 표현과 통계적 수치를 테이블 형태로 보여주고 있다.

단순검색은 sunid, sccs, scopid, pdbid중 하나를 이용하여 SCOP 데이터베이스를 검색하는 기능이다. 사용자가 어떤 정보를 검색하면, 트리의 노드를 순차적으로 탐색하여, 원하는 결과를 Tab 부분에 표시하거나 테이블로 표시한다. 세부검색은 단순검색 보다 좀더 확장된 검색기능을 제공한다. sunid나 scopid와 같이 어느 특정한 데이터를 검색하는 것이 아니라, 결과 데이터가 여러 개가 나올 수 있는 SCOP Database에서의 공통된 특징을 가지는 부분을 검색할 수 있다. 세부검색을 하기 위해서 별도로 검색프레임을 사용한다. (그림 12)는 검색프레임으로부터 검색 결과가 나오는 결과테이블까지의 과정을 설명한 그림이다. 사용자는 각 level에서 검색을 원하는 데이터를 선택하는 과정(①, ②)이 있고, 선택이 끝난 후 검색 결과를 보는 과정(③)이 있다.



(그림 12) SCOPBrowser 검색 기능의 수행 절차

4.2.3 결과 테이블에서의 기능

사용자는 검색한 결과와 관련된 다른 정보를 원하는 경우가 많다. 따라서 SCOPBrowser는 좀더 세부적이고 다양한 기능을 결과 테이블에서 팝업 메뉴를 통해 접근할 수 있도록 하였다. 검색한 결과 테이블에서 해당 열을 선택한 다음 SCOPBrowser에 적용, 도메인 정보를 XML 파일로 보기, PDB 연결, QuickPDB 연결, PDB ID로 재검색 등의 기능을 제공하고 있다.

팝업 메뉴에서 제공하는 주요기능은 다음과 같다.

● 도메인에 대한 XML 파일보기

SCOPML에서는 SCOP Database에서 하나의 도메인에 대한 정보를 모두 XML 파일로 생성하였다. SCOPBrowser는 세부검색의 결과에서 어느 특정한 줄의 도메인에 대한 XML 파일을 볼 수 있게 하는 기능을 가지고 있다. 이 파일은 주어진 도메인에 대한 모든 SCOP Database의 정보를 XML 형식으로 제공한다.

• PDB와의 링크

단백질 구조 연구에서 가장 많이 사용하는 데이터베이스 중의 하나인 PDB와의 링크를 제공한다. SCOPBrowser를 이용할 때, PDB와의 링크를 사용해서 PDB에서의 PDB ID로 검색 결과를 알 수 있으며, 또한 QuickPDB와의 링크를 통해 해당 PDB ID에 대한 QuickPDB 도구와 연결할 수 있다. QuickPDB는 PDB 데이터베이스에 저장된 단백질의 서열 및 3차 구조 정보를 검색하여 보여주기 위한 그래픽 도구이다.

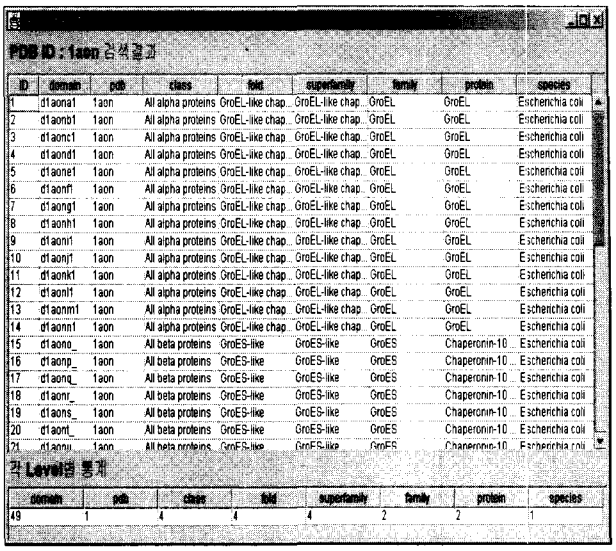
• PDB ID로 결과 재 검색

결과 테이블에서의 PDB ID로 팝업 메뉴를 통한 재 검색이 가능하다. 이러한 기능은 SCOPBrowser의 검색선택 판넬에서 PDB ID로 검색을 하는 것과 동일한 효과를 가지며, 검색 결과 역시 여러 개가 나올 수 있기 때문에 결과 테이블 형식으로 출력하도록 하였다.

4.3 SCOPBrowser 실험 및 분석

SCOP 데이터베이스에서 단백질 구조를 분류하는 기본 단위는 도메인(domain)이다. 단백질에서 도메인은 하나 혹은 그 이상의 구조적 의미를 가지는 부분이 모여 이루어진 독립적인 단위체로 일반적으로 고유의 기능을 가진다. 따라서 우리가 일반적으로 일컫는 단백질 구조 내부에는 이러한 도메인들이 한 개에서부터 수십개씩 존재할 수 있다. 현재 SCOP 사이트에서 제공하는 정보를 이용할 경우 단순히 특정 분류에 대한 도메인 조회나 해당 단백질 도메인에 대한 정보 확인만이 가능하며, 전체 구조 분류들 사이의 연관성이나 다중 도메인 검색 같은 작업은 매우 복잡한 정보처리과정을 수행하는 응용 시스템을 구현하여야만 한다. 하지만, 개발된 SCOPBrowser를 이용하는 경우 개개의 도메인정보 뿐만 아니라 하나의 단백질을 이루는 도메인의 종류, 분류학적인 연관관계, 구조적으로 유사한 도메인들의 존재 유무, 그리고 여러 단백질에 공통으로 나타나는 도메인의 존재 등 다양한 통계자료를 산출할 수 있다.

간단한 예로 (그림 13)은 PDB ID가 '1aon'인 단백질을 SCOPBrowser에 질의하여 얻은 결과 출력 화면이다. 검색 결과는 두 개의 테이블로 제공이 되는데 상단에는 이 단백질을 구성하고 있는 도메인들과 도메인들이 속해있는 전체 구조 분류 계층구조 리스트가 나타나며, 하단에서는 결과리스트에 대한 통계 수치가 나타나게 된다. 그림 하단의 통계 수치는 PDB에 등록된 하나의 단백질에 대한 구조 분류 정보를 보여주는데, 이 단백질이 49개의 도메인을 가지며 이들 도메인들은 각각 4개의 Class, 4개의 fold, 그리고 4개의 superfamily로 분류되어 있으며 하나의 species 분류에만 나타난다는 것을 의미한다. 따라서 여러 단백질에 공통적으로 나타나는 도메인들을 분석할 경우 구조분류학적인 연관성을 이해할 수 있다. <표 1>은 현재 SCOP 사이트에서 제공하는 서비스와 SCOPBrowser와의 효율성을 비교한 표이다.


PDB ID : 1aon 검색결과

ID	domain	pdb	class	fold	superfamily	family	protein	species
1	d1aona1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
2	d1aonb1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
3	d1aonc1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
4	d1aond1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
5	d1aone1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
6	d1aonf1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
7	d1aong1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
8	d1aonh1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
9	d1aoni1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
10	d1aonj1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
11	d1aonk1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
12	d1aonl1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
13	d1aonm1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
14	d1aonn1	1aon	All alpha proteins	GroEL-like chap...	GroEL-like chap...	GroEL	GroEL	Escherichia coli
15	d1aono_	1aon	All beta proteins	GroES-like	GroES-like	GroES	Chaperonin-10...	Escherichia coli
16	d1aonp_	1aon	All beta proteins	GroES-like	GroES-like	GroES	Chaperonin-10...	Escherichia coli
17	d1aonq_	1aon	All beta proteins	GroES-like	GroES-like	GroES	Chaperonin-10...	Escherichia coli
18	d1aonr_	1aon	All beta proteins	GroES-like	GroES-like	GroES	Chaperonin-10...	Escherichia coli
19	d1aons_	1aon	All beta proteins	GroES-like	GroES-like	GroES	Chaperonin-10...	Escherichia coli
20	d1aont_	1aon	All beta proteins	GroES-like	GroES-like	GroES	Chaperonin-10...	Escherichia coli
21	d1aonu_	1aon	All beta proteins	GroES-like	GroES-like	GroES	Chaperonin-10...	Escherichia coli

각 Level별 통계

domain	pdb	class	fold	superfamily	family	protein	species
49	1	4	4	4	2	2	1

(그림 13) 검색 결과 테이블

<표 1> SCOP 서비스와 SCOPBrowser의 비교

특 징	SCOP 서비스	SCOPBrowser
외부DB Link	O	O
Statistics	X	O
검 색	단순검색	단순 + 세부검색
브라우징 형태	단순계층구조	트리계층구조
제공파일타입	단순 텍스트	XML 문서, RDB(예정)

현재 단백질 구조 분류 서비스를 제공하는 SCOP에서는 분류 정보를 활용할 수 있는 텍스트 파일과 단순 조회만 가능한 컨텐츠를 제공하고 있어서, 효용성이 다소 떨어진다고 볼 수 있다. 따라서 이러한 정보를 활용하는 차원에서 보다 사용이 편리하면서 실용성이 높은 시스템을 제공하는 것은 대단히 의미있는 일이다. 단백질 구조에 대한 연구가 점점 활발해지고 있는 시점에서 이러한 중요데이터의 활용성을 극대화 할 수 있는 SCOPBrowser는 관련연구자들에게 많은 도움이 될 것이다. 또한 단백질 구조 분류 응용 시스템을 개발하는 경우에도 본 논문에서 구현한 SCOPML을 이용하여 보다 편리하게 시스템을 개발할 수 있다.

5. 결 론

오랜 연구 결과 축적된 생물정보를 효과적으로 관리하고 데이터의 교환을 용이하게 하기 위한 방법들이 최근 활발하게 연구되고 있다. 그 중 대표적인 것이 XML 기술을 기반으로 한 생물정보의 구조적 문서화 작업이다. 단순 텍스트 형식으로 저장된 자료들을 웹상의 구조적 문서 표준인 XML을 이용하여 재구성함으로써 데이터의 재활용성과 가용성을 높일 수 있다. 본 논문에서는 단백질 구조 분류 데이터베이

스인 SCOP 데이터베이스에서 제공하는 데이터의 XML 표현 기법인 SCOPML과 관련 도구의 개발에 대하여 기술하였다.

SCOPML은 단순 텍스트 형식으로 제공되는 SCOP 데이터를 표준적인 구조화된 문서로 표현하기 위한 마크업 언어이다. SCOP의 단백질 구조 분류를 표현하기 위하여 SCOPML DTD를 정의하였으며, SCOP 데이터에 대한 XML 문서 생성을 위하여 새로운 변환기를 구현하였다. 추가로, SCOPML을 활용하여 XML 기반의 SCOP 데이터 뷰어인 SCOP-Browser를 개발하였다. SCOPBrowser는 단백질 도메인을 중심으로 전체 SCOP 의 단백질 구조 분류를 한눈에 볼 수 있는 트리보기 기능, 다양한 옵션에 의하여 도메인에 대한 통계를 구할 수 있는 검색기능, 그리고 해당 도메인과 단백질 구조 데이터베이스인 PDB와의 링크 기능 등을 제공한다. 개발된 SCOPML과 SCOPBrowser는 서열정보의 번역으로 생성되는 단백질에 대한 연구 및 새로운 단백질 구조가 밝혀졌을 경우 단백질간의 연관관계를 규명하기 위한 도구로 널리 활용될 수 있으리라 기대된다.

향후 연구로는 SCOPML의 재사용성과 확장성을 고려하여 DTD를 *스키마(Schema)*[16] 구조로 재구성하고 SCOP-Browser의 검색기능을 보다 세분화하여 보다 다양한 단백질 구조 및 도메인 정보에 대한 통계자료를 산출할 수 있는 도구로 확장할 예정이다.

참 고 문 헌

[1] Frederic, A., Guy, V. and Emmanuel, B., "XML, bioinformatics and data integration," Bioinformatics, Vol.17, pp. 115-125, Feb., 2001.

[2] Alexey, M., Steven, B., Tim, H. and Cyrusm Ch., "SCOP : A Structural Classification of Proteins Database for the Investigation of Sequences and Structures," J. Mol. Biol., 247, pp.536-540, 1995.

[3] Mateosian, R., "XML : Learning XML : Creating Self-describing data," IEEE micro, Vol.21, No.2, pp.37-40, 2001.

[4] Gottesman, B. Z., "Why XML Matters," PC magazine, Vol. 17, No.17, 1998.

[5] http://scop.berkeley.edu, Scop site-Structural Classification of Proteins.

[6] Guerrini, VH. and Jackson, D., "Bioinformatics and Extended Markup Language," Online Journal of Bioinformatics, 1, pp.12-21, 2000.

[7] http://www.bsml.org, "XML Data Standard for Genomics : The Bioinformatic Sequence Markup Language (BSML) DTD".

[8] P. Murray-Rust, and H. Rzepa, "Chemical Markup Language and XML Part I. Basic principles," J. Chem. Inf. Comp, Sci, Vol.39, No.6, pp.928-942, 1999.

[9] Dniel, H., Ralf, Z. and Thomas, L.,"ProML-The Protein Markup Language for specification of protein sequences, structures and families," German Conference on Bioinformatics 2001, Oct., 2001.

[10] http://openmms.sdsc.org/, "Corba, Relation Database and XML Software for Macromolecular Structure".

[11] John, W., Zukang, F. and Helen, M., "The Protein Data Bank : unifying the archive," Nucleic Acids Research, Vol. 30, No.1, pp.245-248, 2002.

[12] Bairoch, A. and Apweiler, R., "The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 2000," Nuceic acids research, Vol.28, pt.1, pp.45-48, 2000.

[13] Bairoch, A., Bucher, P. and Hofmann, K., "Nucleic acids research," Vol.25, No.1, pp.217-221, 1997.

[14] Orengo, C. A., Pearl, F. M. G., Bray, J. E. and Todd, A. E., "The CATH Database provides insights into protein structure/function relationships," Nucleic acids research, Vol.27, No.1, pp.275-279, 1999.

[15] Josan, H., "JDOM Makes XML Easy," Sun's JavaOne Conference, session # 2104, 2002.

[16] Loannides, D., "XML schema languages : beyond DTD," Library Hi Tech, Vol.18, No.1, pp.9-14, 2000.

안 건 태

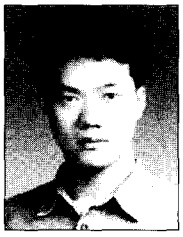
e-mail : java2u@mail.ulsan.ac.kr

1999년 울산대학교 전자계산학과(공학사)

2001년 울산대학교 컴퓨터정보통신 공학부 (공학석사)

2001년~현재 울산대학교 컴퓨터정보통신 공학부 공학박사과정

관심분야 : 생물정보학, 로직프로그래밍, 협업지원시스템, 웹 프로그래밍 등



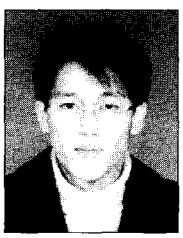
윤 형 석

e-mail : miracle@ulsan.ac.kr

2001년 울산대학교 컴퓨터공학과(공학사)

2001년~현재 울산대학교 컴퓨터정보통신 공학부 공학석사과정

관심분야 : 네트워크 모니터링, 생물정보학, 로직프로그래밍, 컴파일러 등



황 의 윤

e-mail : heyoon@ulsan.ac.kr

2002년 현재 울산대학교 컴퓨터정보통신 공학부 공학사 과정

2003년 울산대학교 컴퓨터정보통신 공학부 졸업예정

관심분야 : 생물정보학, 웹 프로그래밍 등

김 진 홍

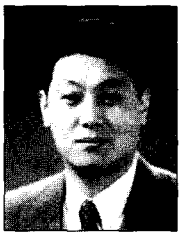
e-mail : karif99@ulsan.ac.kr

1999년 울산대학교 전자계산학과(공학사)

2001년 울산대학교 컴퓨터정보통신 공학부 졸업(공학석사)

2001년~현재 울산대학교 컴퓨터정보통신 공학부 공학박사과정

관심분야 : 생물정보학, 웹 프로그래밍, 로직프로그래밍, 협업지원 시스템 등



이 명 준

e-mail : mjlee@mail.ulsan.ac.kr

1980년 서울대학교 수학과(학사)

1982년 한국과학기술원 전산학과(공학석사)

1991년 한국과학기술원 전산학과(공학박사)

1982년~현재 울산대학교 컴퓨터정보통신 공학부(교수)

1993년~1994년 미국 버지니아대학 교환교수

관심분야 : 프로그래밍언어, 생물정보학, 분산 객체 프로그래밍 시스템, 병행실시간 컴퓨팅, 인터넷 프로그래밍시스템 등