유전자 상호작용 데이타베이스 SOAP 서버 객체 모델의 설계 및 구현

(Design and Implementation of SOAP Servers Object Model for Gene Interaction Databases)

이 호 일 † 유 성 준 †† 김 민 경 †††
(HO IL LEE) (Seongjoon Yoo) (Minkyung Kim)

요 약 최근 주요 생물정보학 데이타베이스 중 DDBJ, ENSEMBL, KEGG, 등의 데이타베이스는 연구자들의 편의를 위해 데이타와 분석용 도구들을 웹 서비스를 이용하여 제공한다. 이와 같이 웹 서비스를 이용하여 서비스를 제공하기 위해서는 SOAP 서버 객체와 메소드 정의가 매우 중요하다. 이 연구에서는 BIND, MINT, DIP과 같은 유전자 상호작용 데이타베이스를 위해서 필요한 SOAP 서버 객체에 대한 요구사항을 도출한다. 이어서 이 요구사항을 만족하는 SOAP 서버 객체와 메소드를 정의하였다. 이를 기반으로 프로토타입을 설계하고 구현한 것에 대하여 기술한다.

키워드 : 웹 서비스, 바이오인포메틱스, 상호작용, 데이타베이스, 통합

Abstract Recently main Bioinformatics databases(DDBJ, ENSEMBL, KEGG, etc.) provide analysis tools and data using web services for the convenience of bioinformaticians. Thus, defining SOAP server objects and their methods are very important to provide services for web services. We define SOAP server objects for interaction databases such as BIND, MINT and DIP.

Key words : Web services, soap, bioinformatics, interaction, database, integration

1. 서 론

대부분의 생물정보 데이타베이스들은 연구자들을 위해 데이타를 flatfiles, XML, Database Dump files 형식으로 제공한다. 동질 혹은 이질적 주제의 데이타베이스들을 통합하여 새로운 발견이 가능하기 때문에 전 세계 많은 연구소에서는 생물정보 통합 데이타베이스들을 만든다. 기존의 생물정보 데이타베이스를 통합하는 방법으로는 링크 기반, 뷰 기반, 데이타웨어하우스 기반 방법 등이 활용되었다. 그러나 이러한 통합 방법은 여러 데이타베이스에서 데이타를 추출/저장하기 위해 스크립트를 작성하는 중복된 노력이 요구되고 시스템 유지/관리 등이 어려운 문제점이 있다. 따라서 기존 방식 외에 데이타의 재사용성을 증가시키고, 주소 혹은 스키마의 변경과 무관하게 그 데이타를 사용 가능하게 하는 다른 접근 방식이 요구되었다. 최근 바이오인포매틱스 연구 분야에서는 이러한 문제를 극복하고자 하는 노력의 일환으로 웹 서비스 기술을 도입하고 있다[1-3].

단백질체학의 대두 등으로 유전자 상호작용 데이타베이스의 중요성이 부각되고 있음에도 불구하고, 이들 데이타베이스를 위한 SOAP 서버는 아직 소개되지 않고 Flatfiles, XML 등 형태의 데이타를 그대로 이용하고 있는 형편이다. 이러한 현실에서 Pathway 등을 포함한 유전자 상호작용에 대한 연구를 하는 데에는 위에서 언급한 바와 같은 어려움이 있음에도 불구하고 아직 상호작용 데이타베이스가 웹 서비스를 기반으로 지원되지 않고 있는 실정이다. 이에 본 연구에서는 이 상호 작용 데이타베이스를 웹 서비스 기술을 기반으로 제공하기 위한 SOAP 서버 객체와 메소드를 설계하게 되었다. 이 논문에서는 유전자 상호작용 데이타베이스의 특징을 고려해서 유용한 SOAP 서버 객체를 정의한 내용을 기술한다.

웹 서비스가 많은 장점을 가지고 있음에도 불구하고 생물정보학에 빠르게 도입하지 못하는 이유는 컴퓨터 공학자들의 생물정보학에 대한 정보 부족으로 유용한

† 정 회 원 : ㈜마크로젠 연구원
headil@hanmail.net
†† 종신회원 : 세종대학교 컴퓨터공학부 교수
sjyoo@sejong.ac.kr
Corresponding author
††† 비 회 원 : 이화여자대학교 공과대학 컴퓨터학과 교수
minkykim@ewha.ac.kr
논문접수 : 2004년 3월 8일
심사완료 : 2004년 10월 15일

SOAP 서버 객체와 메소드를 정의하는 것이 어렵기 때문이다. 우리는 컴퓨터 공학자와 생물학자가 협력하여 지난 수년간 유전자 상호작용에 대하여 연구해오고 있고, 데이타웨어하우스 기반의 유전자 상호작용 데이타베이스를 통합한 경험을 가지고 있다. 이러한 경험을 바탕으로 상호작용 데이타베이스를 활용하기 위한 객체 및 메소드를 도출할 수 있게 되었다. 이는 이 논문의 4장에서 SOAP 서버 객체에 대한 요구사항으로 정리하고 이 요구사항을 만족하는 객체 및 메소드를 정의, 설계한다.

2장에서는 웹 서비스를 이용하여 데이타를 제공하는 생물정보 데이타베이스인 KEGG와 DDBJ의 SOAP 서버 객체에 대하여 살펴본다. 3장에서는 유전자 상호작용 데이타베이스인 BIND, DIP, MINT의 데이타 모델 및 데이타 제공 방법에 대하여 기술한다. 4장에서는 상호작용 데이타베이스에 대한 SOAP 서버 객체에 대하여 기술하고, 5장에서는 우리 시스템의 프로토타입에 대하여 기술한다. 마지막 장에서는 결론 및 향후 연구에 대하여 기술한다.

2. 관련연구

웹서비스를 이용하여 데이타를 제공하는 생물정보 데이타베이스의 대표적인 것으로 KEGG[4,5]와 DDBJ가 있다. 아래 기술한 바와 같이 KEGG는 네 개의 SOAP 서버 객체를 제공한다. 이들은 각각 데이타베이스를 검색하는 데 활용되는 것과 유사한 서열을 검색하는 데 사용되는 객체로 대별된다. DDBJ SOAP 서버 객체는 기존 데이타베이스를 검색하는 기능 외에 Blast, ClustalW, Fasta 등과 같이 유사한 서열을 찾는 기능, SRS 등을 연결하여 검색하는 기능 등을 제공한다.

2.1 KEGG

KEGG는 1995년 5월 일본 휴먼 지놈 프로젝트의 일환으로 구축되기 시작했다. 이 데이타베이스는 기능 유전체학 연구를 위한 대사과정 비교, 대사과정 재구성 및 대사과정 설계 등과 같은 새로운 생물정보학 기술을 개발하기 위한 데이타베이스를 가지고 있고, 2003년부터 웹 서비스로 일부 데이타를 서비스하고 있다.

KEGG의 웹 서비스를 위한 객체는 KEGG, GENES, SSDB(Sequence Similarity DataBase), PATHWAY의 4개로 구분하고, 각각의 객체는 다음과 같은 서비스를 제공한다.

(1) KEGG

KEGG 객체는 KEGG의 데이타베이스를 검색할 수 있는 서비스를 제공한다.

(2) GENES

GENES 객체는 Pathway에 관련된 유전자(Genes)정보를 제공한다.

(3) SSDB

SSDB 객체는 유전자와 유전자사이의 서열 유사성(Sequence Similarity)에 대한 정보를 제공한다.

(4) PATHWAY

PATHWAY 객체는 생물학에 관련된 Pathway정보를 제공한다.

2.2 DDBJ

DDBJ[6]는 미국의 NCBI(National Center for Biotechnology Information)[7], 유럽의 EMBL(European Molecular Biology Laboratory)[8]과 같이 DNA sequence 정보를 제공해 주는 곳이다. DDBJ의 웹 서비스는 Blast, ClustalW, DDBJ, ExClustalW, Fasta, GetEntry, Gib, Gtop, SRS, TxSearch의 객체로 구성되고, 각각의 객체는 다음과 같은 서비스를 제공한다.

(1) Blast

Blast 서비스는 단백질 또는 염기 서열들 간의 유사성을 찾아주는 서비스로 DDBJ, DAD, EPD, PDB, PDBSH, PIR, PRF, PROTEIN, SWISSPROT, WORM DNA, WORM PEPTIDE의 데이타베이스가 제공된다.

(2) ClustalW

ClustalW 객체는 단백질이나 핵산의 서열들을 Multiple Alignment해주는 서비스를 제공한다.

(3) DDBJ

DDBJ 객체는 DDBJ의 데이타베이스를 검색할 수 있는 서비스로 데이타베이스의 접근 번호(accession number)를 입력하여 Flat File이나 XML형식으로 해당 데이타를 제공한다.

(4) ExClustalW

ExClustalW 객체는 단백질이나 핵산의 서열들을 입력하여 ClustalW를 실행하고 결과를 반환하는 서비스를 제공한다.

(5) Fasta

Fasta객체는 단백질 또는 DNA 서열 라이브러리에서 유사한 서열을 검색하고 결과를 반환하는 서비스를 제공하고, DDBJ, DAD, EPD, PDB, PDBSH, PIR, PRF, PROTEIN, SWISSPROT, WORM DNA, WORM PEPTIDE의 데이타베이스가 제공된다.

(6) GetEntry

GetEntry객체는 DDBJ, DAD, EMBL, PDB, PIR, PRF, SWISSPROT들의 데이타베이스를 검색할 수 있는 서비스로 데이타베이스의 접근 번호(accession number)를 입력하여 Flat File이나 XML형식으로 해당 데이타를 제공한다.

(7) Gtop

Gtop객체는 유전자(Gene)와 생물체(Organism)에 대한 정보를 보여주는 서비스를 제공한다.

(8) SRS

SRS 객체는 SRS(Sequence Retrieval Service)를 실행하고 결과를 반환하는 서비스로 DDBJ, DDBJNEW, DAD, DADNEW, SWISSPROT, PIR, PROSITE, PROSITEDOC, BLOCKS, PRINTS, PFAMA, PFAMB, SWISSPFAM, PFAMHMM, PFAMSEED, PRODOM, ENZYME, PDB, HSSP, FSSP, PATHWAY, LENZYME, LCOMPOUND들의 데이타베이스가 제공된다.

(9) TxSearch

TxSearch서비스는 입력한 Taxonomy이름에 따라 분류된 출력 값을 반환하는 서비스를 제공한다.

3. 유전자 상호작용 데이타베이스

3장에서는 유전자 상호작용 데이타베이스의 종류와, 이들의 데이타베이스 모델 및 데이타 제공 방법에 대하여 기술한다.

3.1 BIND

BIND(Biomolecular Interaction Network Database)[9]는 Mount Sinai 병원의 Samuel Lunenfeld 연구소의 연구 프로그램이며, 생체 분자들의 상호작용과 Pathway 정보를 다루는 데이타베이스이다. 이 데이타베이스는 Interaction, Molecular Complex, Pathway 형태로 구분한다. Interaction은 두개의 유전자가 상호작용하는 것을 의미한다. Interaction을 구성하는 유전자에는 protein, DNA, RNA, ligand, molecular complex, gene, photon 혹은 아직 분류되지 않은 개체들이 있다. Complex는 1개 이상의 Interaction이 모여서 특정 기능을 수행하는 집합체를 의미하며, Pathway는 2개 이상의 Interaction이 모여서 만든 물질/신호전달 경로를 의미한다. 현재 BIND는 ASN.1, XML, flatfiles 형식으로 데이타를 제공하고 있다.

그림 1은 Bind의 *Interaction*을 보여주는 그림이다. 하나의 Interaction 데이타는 자신을 구성하는 개체들의 상호 작용 정보와 실험 조건, PubMed 등의 정보를 포함하고 있다. 여기서 개체는 protein, DNA, RNA, ligand, molecular complex, gene, photon, unclassified biological entity가 될 수 있다.

그림 2는 Bind의 Molecular Complex 구조이다.

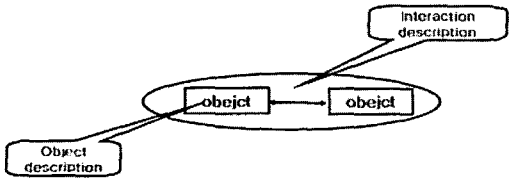

그림 1 BIND의 인터액션 개념도

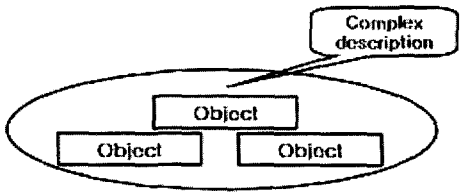

그림 2 BIND의 Molecular Compex 구조도

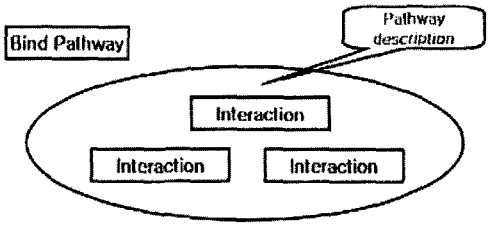

그림 3 BIND의 Pathway 구조도

Complex는 2개 이상의 *Interaction*들이 연결되어 특정 기능을 수행하는 집합체를 말한다. 하나의 Complex 데이타는 자신을 구성하는 개체들과 Interaction들의 정보를 가지고 있으며, Complex에 대한 설명, PubMed 정보등을 포함하고 있다.

그림 3은 BIND의 Pathway 구조이다. Pathway는 2개 이상의 Interaction들을 규정된 순서대로 연결한 생체작용경로를 말한다. 하나의 Pathway 데이타는 Pathway를 이루는 여러 개의 Interaction에 대한 정보와 Pathway에 대한 설명, PubMed 등의 정보를 포함하고 있다.

3.2 DIP

DIP(Database of Interacting Proteins)[10]은 실험적으로 결정된 protein-protein interaction에 대한 자료들을 수집, 제공하는 데이타베이스이다. 이 데이타베이스는 Node와 Edge로 구분하여 정보를 표현한다. Node는 상호작용하는 단백질을 의미하고, 단백질의 이름, 기능, 세포내 위치 등의 기본 정보와 다른 데이타베이스(PIR, SWISSPROT, GENBANK)에 대한 참조 정보를 나타낸다. Edge는 두 단백질의 상호작용에 대한 실험 방법, 특징, 관련문서 정보 등을 설명하는 정보를 나타낸다. 현재 DIP에서는 XML 형식과 탭으로 구분한 flatfiles 형식으로 데이타를 제공한다. 이 XML 문서는 두 종류로 구분할 수 있다. 하나는 DIP에서 정의한 스키마(*dip3.xsd*)에 따르는 문서이고, 다른 하나는 PSI(Proteomics Standards Initiative)에서 정의한 스키마(MIF.xsd)에 따르는 문서이다.

3.3 MINT

MINT(Molecular INTeraction database)[11,12]는 생체 분자들 간의 상호작용 정보를 저장하는 관계형 데이

타베이스이다. 이 데이타베이스는 Interactor와 Interaction으로 구분하여 정보를 표현한다. Interactor는 상호작용하는 단백질 정보를 나타내고, Interaction은 상호작용에 대한 실험 방법, 특징, 관련문서 등의 정보를 나타낸다. 현재 MINT에서는 특정 문자('|')로 구분한 flatfiles 형식과 PSI[13]에서 정의한 XML 스키마(MIF.xsd)에 따르는 XML 형식으로 데이타를 제공한다. 그리고 flatfiles 형식과 XML 형식을 상호 변환 가능하도록 지원하는 툴(maker/flattener)을 제공한다.

4. 상호작용 데이타베이스를 위한 SOAP 서버 객체의 설계

이 장에서는 BIND, MINT, DIP과 참조 데이타베이스의 SOAP 서버 객체에 대하여 기술한다.

4.1 SOAP 서버 객체에 대한 요구사항

당 연구실에서는 BioPathDB[14]라고 하는 Pathway 데이타를 통합 검색하기 위한 데이타웨어하우스를 개발한 경험이 있다. 우리는 이 연구와 KEGG의 예에서 상호작용 데이타베이스를 이용하기 위해서는 ID나 이름 등으로 검색하는 기본적인 기능 외에 다음과 같은 주요 기능이 필요하다는 것을 발견했다. 이 논문에서 정의하는 객체와 메소드는 이러한 요구사항을 기반으로 설계하였다.

1) Interaction ID로 Pathway를 검색할 수 있어야 한다.

2) Pathway를 구성하는 Interaction 리스트를 검색할 수 있어야 한다.

3) 하나의 상호작용을 구성하는 유전자들의 정보를 검색할 수 있어야 한다. 즉, 특정 Pathway에 존재하는 유전자들도 검색할 수 있어야 하며, 반대로 특정 유전자가 존재하는 모든 Pathway를 검색할 수 있어야 한다.

4.2와 4.3, 4.4절에서 각 상호작용 데이타베이스를 위한 SOAP 서버 객체에 대하여 기술한다. 각 메소드에 대한 내용은 참고문헌 [15]의 부록에 수록하고 여기에서는 생략하였다.

4.2 BIND 데이타베이스 SOAP 서버 객체 정의

3장에서 언급한 바와 같이 BIND는 Interaction, Complex, Pathway로 데이타를 구분하여 데이타베이스에 저장한다. Interaction은 2개의 유전자가 상호 작용하는 것을 말한다. 유전자는 단백질, DNA, RNA, Small-molecule, Complex 등이 될 수 있다. Complex는 1개 이상의 Interaction이 모여 하나의 특정기능을 수행하는 Interaction의 집합체를 말하고, Pathway는 2개 이상의 Interaction이 연결된 신호/물질 전달 경로를 말한다.

BIND에서는 Interaction, Pathway, Complex 정보를 제공하므로 이들을 지원하는 SOAP 서버 객체도 이에 따라 세 개로 구분하여 정의한다. Bind_InteractionIF는 Interaction 관련 데이타를 제공해주는 객체이며, DNA, RNA, Complex, small-molecule과 같은 유전자 정보를 이용하여 Interaction 정보를 검색하고, 반대로 Interaction 정보를 이용하여 Interaction에 관련된 유전자 정보, 관련논문, 실험정보 등을 검색하는 메소드를 지원한다. Bind_PathwayIF는 Pathway 관련 데이타를 제공하는 객체이며, Interaction을 이용하여 Pathway정보를 검색하고, 반대로 Pathway 정보를 이용하여 Interaction 정보, 관련논문 등을 검색하는 메소드를 지원한다. Bind_ComplexIF는 Complex 관련 데이타를 제공해주는 객체이며, Interaction을 이용하여 Complex 정보를 검색하고, 반대로 Complex 정보를 이용하여 Interaction 정보를 검색하는 메소드를 지원한다.

그림 4는 BIND의 SOAP 서버 객체들이 정의하는 주요 메소드를 보여준다. 사용자는 메소드 이름에서 메소드들의 입/출력 타입이 무엇인지 알 수 있다. 메소드 이름 규칙은 "By"를 중심으로 좌측은 메소드의 출력 값이 무엇인지 보여주고, 오른쪽은 입력 값이 무엇인지 보여준다. 입력 값이 여러개일 경우가 있는데, 이는 "_"로 구분하여 입력 값을 순서대로 나열한다. 예를 들어 "get_InterIdByName_Gi()" 메소드는 출력 값으로 Interaction ID를 출력하고, 입력 값으로 유전자 이름과 유전자 ID(NCBI의 Gene ID)를 갖는다. "By"가 없는 메소드는 입력 값이 없는 경우이다. 예를 들어 "get_All_Interaction()" 메소드는 입력 값이 필요 없고, BIND 데이타베이스의 모든 Interaction 정보를 출력한다.

이상의 메소드 중 "get_PubmedByPathwayId()" 하나를 예를 들어본다. 그림 5는 Pathway ID를 이용하여 PubMed 정보를 검색하는 이 메소드의 pseudo code이다. 입력 값으로 Pathway ID를 읽어서 데이타베이스를 검색하고 검색된 PubMed 정보 리스트를 리턴하는 구조이다. 다른 메소드들도 이와 유사한 구조로 되어있다.

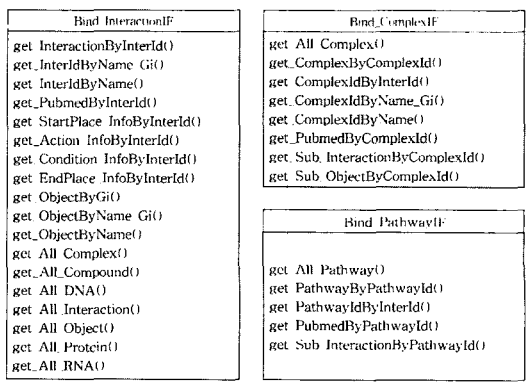

그림 4 BIND의 SOAP 서버 메소드

```
i=0
READ_PARAMETER pathway_id
query_string = "select * from pathway_source where pathway_id = " + pathway_id
EXECUTE_QUERY query_string
WHILE ( SQL_HAS_RESULT )
        result(i) = GET_PUBMED_INFORMATION
        i = i + 1
ENDWHILE
RETURN result
```

그림 5 "get_PubmedByPathwayId()"의 Pseudo Code

4.3 MINT의 SOAP 서버 객체

MINT는 Interaction, Interactor, Experiment로 구분하여 데이타를 저장한다. 하나의 Interaction은 2개의 Interactor와 1개의 실험 정보를 가지고서 유전자 상호작용 정보를 나타낸다.

MINT에서는 Interaction 정보만을 제공하므로 하나의 객체만을 정의한다. MintInteractionIF는 Interaction 데이타를 제공해주는 객체이며, 유전자 정보를 이용하여 Interaction 정보를 검색하고, 반대로 Interaction 정보를 이용하여 Interaction에 참여하는 유전자 정보, 관련논문, 실험정보 등을 검색하는 메소드를 지원한다.

그림 6은 MINT의 SOAP 서버 객체들이 정의하는 주요 메소드를 보여준다. 메소드 이름 규칙은 앞에서 언급한 BIND와 동일하다.

MINTInteractionIF
getMintInteractionByInterId()
getMintInteractionBySpId()
getMintInteractionBySpIds()
getMintInteractDomainByInterId()
getMintPubmedByInterId()
getMintExperimentByInterId()
getCommentByInterId()
getNegationByInterId()
getInteractionTypeByInterId()
getOrganismBySpId()
getShortNameBySpId()
getMintInteractorFeatureByInterId()

그림 6 MINT의 SOAP 서버 메소드

4.4 DIP의 SOAP 서버 객체

DIP은 Node와 Edge로 구분하여 데이타를 저장한다. Edge는 두 유전자 간의 상호작용을 나타내며, Node는 상호 작용하는 유전자를 나타낸다. 즉, 하나의 Edge는 2개의 Node를 가지고 상호작용 정보를 나타낸다.

DIP에서는 Interaction 정보만을 제공하므로 하나의 객체를 정의한다. DipInteractionIF는 Interaction 데이타를 제공해주는 객체이며, 유전자 정보를 이용하여 Interaction 정보를 검색하고, 반대로 Interaction 정보를 이용하여 Interaction에 참여하는 유전자 정보, 관련논문, 실험정보 등을 검색하는 메소드를 지원한다. 그림 7은 DIP의 SOAP 서버 객체들을 정의하는 주요 메소드를 보여준다. 메소드 이름 규칙은 앞에서 언급한 BIND와 동일하다.

DipInteractionIF
getDipInteractionByNodesId()
getDipInteractionByEdgeId()
getDipInteractionByNodeId()
getDipNodeByNodeId()
getNodeNameByNodeId()
getSwissprotIdByNodeId()
getNcbiIdByNodeId()
getPirIdByNodeId()
getDescriptionByNodeId()
getOrganismByNodeId()
getTaxonByNodeId()
getNodeIdByNodeName()
getNodeIdBySwissprotId()
getNodeIdByNcbiId()
getNodeIdByPirId()
getPubmedIdByEdgeId()
getExperimentByEdgeId()
getClassByEdgeId()
getEdgeFeatureIdByEdgeId()
getEdgeFeatureByEdgeId()
getEdgeFeatureByEdgeFeatureId()

그림 7 DIP의 SOAP 서버 메소드

5. SOAP 서버 객체의 효용성 분석

이 장에서는 4장에서 정의한 SOAP 서버 객체들의 효용성을 증명하기 위해서 프로토타입을 구현한 것에 대해 기술한다. 5.1에서는 전체 프로토타입 시스템 구조와 정의된 객체의 활용 예를 SOAP 메시지의 요청 및 응답을 통하여 기술한다. 5.2, 5.3, 5.4에서는 상호작용 데이타베이스를 관계형 데이타베이스 형태로 구축한 것에 대하여 기술한다. 프로토타입 시스템 개발 환경으로

인텔 펜티엄 4 1.9MHz, 메모리 512MB의 하드웨어 시스템을 사용하였고, 소프트웨어 시스템은 윈도우 2000 프로페셔널, JWSDP 1.3, JDK 1.4를 각각 활용하였다. 데이타베이스는 MS ACCESS를 사용하였다.

5.1 프로토타입 시스템 구조 및 객체 활용 예

그림 8은 우리가 개발한 프로토타입 시스템의 구조이다. 이 시스템에는 BIND, DIP, MINT의 데이타를 제공하는 SOAP 서버 객체와 참조 데이타베이스의 SOAP 서버 객체가 있고, 이들 객체들을 이용하는 사용자가 있다. 이 시스템의 전체 흐름은 일반적인 웹 서비스 구조(Service provider, Service Requestor, Registry)와 유사하다. 일반적인 웹 서비스 구조에서는 Service Requestor가 자신이 원하는 서비스를 Registry에서 검색하고, 검색 결과로 서비스의 위치 정보를 받아서 원하는 서비스를 이용한다. 이와 비슷하게 우리의 시스템은 사용자가 참조 데이타베이스의 SOAP 서버 객체에서 Interaction을 검색하고, 검색 결과로 SOAP 서버 객체 이름과 Interaction ID 값을 받아서 해당 SOAP 서버 객체로부터 원하는 Interaction 정보를 찾는다. 즉, 참조 데이타베이스의 SOAP 서버 객체는 Registry 역할을 하고, BIND, MINT, DIP의 SOAP 서버 객체는 Service Provider 역할을 하고, Interaction 정보를 찾는 사용자는 Service Requestor 역할을 한다.

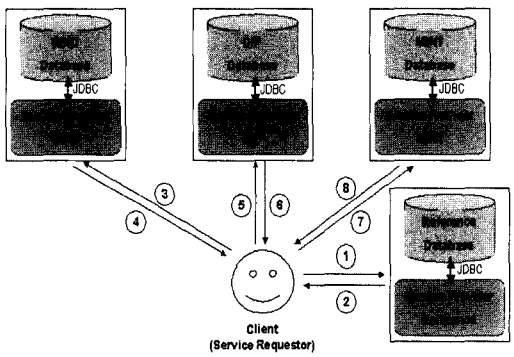

그림 8 프로토타입 시스템 구조도

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
xmlns:env="http://www.w3.org/2001/06/soap-envelope"
xmlns:xsd"http://www.w3.org/2001/XMLSchema"
xmlns:xsi"http://www.w3.org/2001/XMLSchema-instance"
xmlns:enc"http://schema.xmlsoap.org/soap/encoding/"
xmlns:bind="http://bind.ca"
env:encodingStyle="http://www.w3.org/2001/06/soap-encoding">
        <env:Body>
                <bind:get_PathwayIdByInterId>
                        <Int_1
xsi:type="xsd:string">1653</Int_1>
                </bind:get_PathwayIdByInterId>
        </env:Body>
</env:Envelope>
```

그림 9 "get_PathwayIdByInterId"의 SOAP 요청

그림 9와 그림 10은 BIND의 SOAP 서버 객체에서 정보를 요청/응답하는 SOAP 메시지 예제이다. 그림 9는 "get_PathwayIdByInterId()" 메소드를 통하여 Pathway 정보를 요청하는 SOAP 메시지이다. 이와 유사하게 다른 SOAP 서버 객체도 위와 같은 구조의 시스템에서 활용될 수 있다.

그림 10은 그림 9의 요청 SOAP 메시지에 대한 응답 SOAP 메시지이다.

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
xmlns:env="http://www.w3.org/2001/06/soap-envelope"
xmlns:xsd"http://www.w3.org/2001/XMLSchema"
xmlns:xsi"http://www.w3.org/2001/XMLSchema-instance"
xmlns:enc"http://schema.xmlsoap.org/soap/encoding/"
xmlns:bind="http://bind.ca"
env:encodingStyle="http://www.w3.org/2001/06/soap-encoding">
        <env:Body>
                <bind:get_PathwayIdByInterIdResponse>
                        <result xsi:type="enc:Array"
enc:arrayType="xsd:string[1]">
                                <item>28</item>
                        </result>
                </bind:get_PathwayIdByInterIdResponse>
        </env:Body>
</env:Envelope>
```

그림 10 "get_PathwayIdByInterId()"의 SOAP 응답

5.2 BIND 데이타베이스의 변환

3장에서 기술한대로 BIND는 ASN.1, XML, flatfiles 형식으로만 데이타를 제공하고 있다. 이 연구에서는 프로토타입 시스템을 구축하기 위하여 XML 형식으로 된 데이타를 RDBMS에 맞게 변환하였다. 그림 11은 BIND의 Interaction 정보를 관계형 데이타베이스로 변환한 데이타베이스 스키마이다. Bind_Interaction 테이블은 Interaction을 구성하는 두 유전자의 이름과 Interaction ID를 갖는 중요한 테이블이고, 다른 테이블은 유전자에 대한 세부정보, Interaction에 관련된 실험정보, PubMed 정보 등을 가지고 있는 테이블이다[16,17].

그림 12는 BIND의 Complex 정보를 관계형 데이타베이스로 변환한 데이타베이스 스키마이다. bind_complex 테이블은 complex ID와 Complex에 대한 설명을 갖는 중요한 테이블이고, Complex_Inter_list 테이블은 Complex를 구성하는 Interaction ID를 갖는 테이블이다. Complex_Sub_unit 테이블은 Complex를 구성하는 Objects를 갖는 테이블이고, Complex_Source 테이블은 Complex를 증명해주는 논문의 PubMed ID를 갖는 테

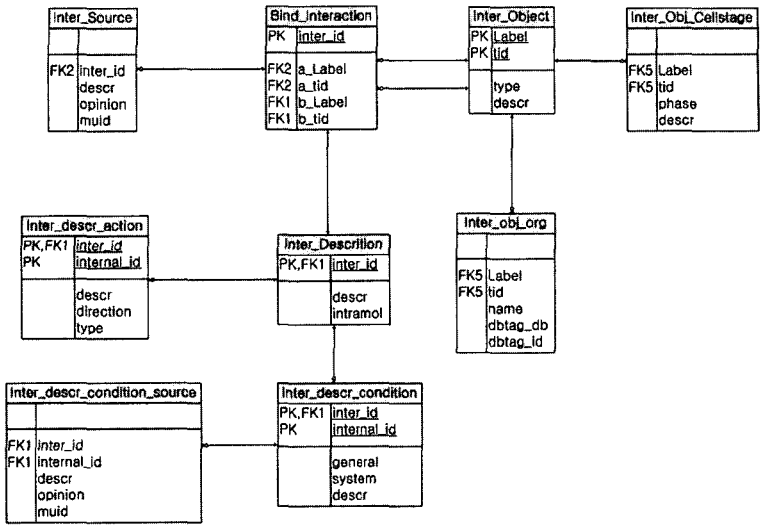

그림 11 BIND의 Interaction 데이타베이스

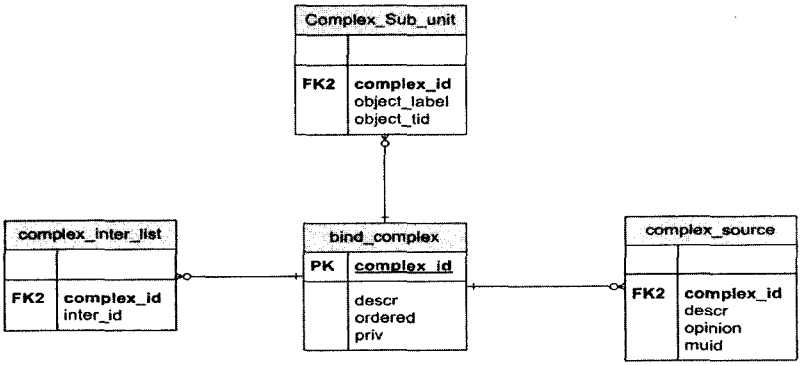

그림 12 BIND의 Complex 데이타베이스

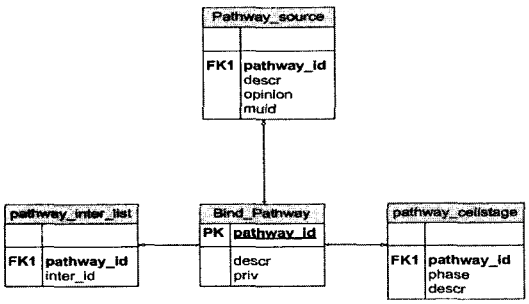

그림 13 BIND의 Pathway 데이타베이스

이블이다.

그림 13은 BIND의 Pathway 정보를 관계형 데이타베이스로 변환한 데이타베이스 스키마이다. bind_Pathway 테이블은 Pathway ID와 Pathway에 대한 설명을 갖는 중심 테이블이고, Pathway_Inter_list 테이블은 Pathway를 구성하는 Interaction ID를 갖는 테이블이다. Pathway_Source 테이블은 Pathway를 증명해주는 논문의 Pubmed ID를 갖는 테이블이다.

5.3 MINT 데이타베이스의 변환

프로토타입 시스템을 개발하기 위하여 MINT에서 제공하는 flatfiles를 'Microsoft Office Excel'을 이용하여 필요한 데이타를 추출하고, 관계형 데이타베이스를 구축하였다. 그림 14는 MINT의 정보를 관계형 데이타베이스로 변환한 데이타베이스 스키마이다. 이 스키마에서 Mint_Interaction 테이블은 Interaction ID(Inter_id)와 Interaction에 참여하는 개체(sp_protein_a, sp_protein_b)정보가 있다. Mint_Domain 테이블은 Interaction에 참여하는 개체의 도메인 정보(name_a, range_start_a, name_b, range_startb, etc.)가 있다. Mint_Experiment

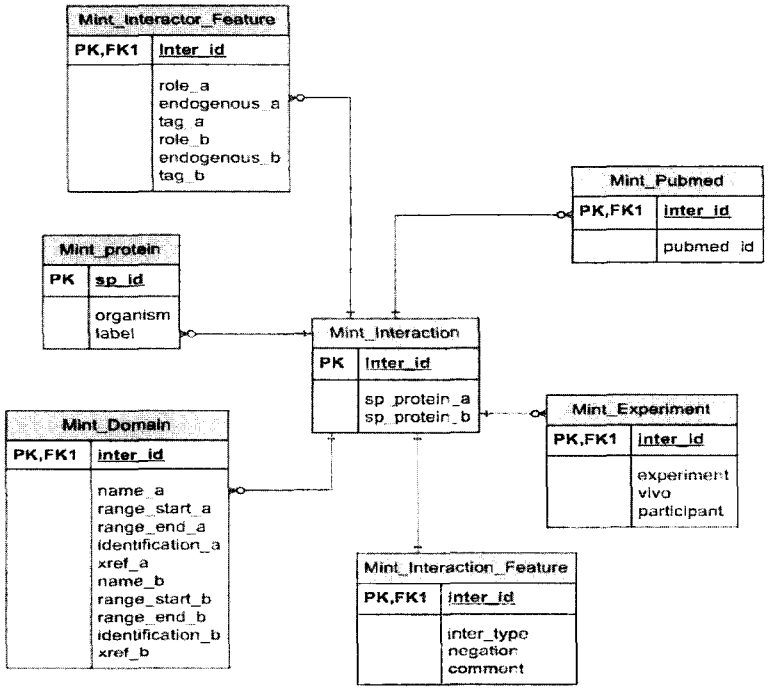

그림 14 MINT 데이타베이스

테이블은 Interaction에 대한 실험 정보(experiment, vivo, participant)가 있다. Mint_PubMed 테이블은 Interaction에 대한 PubMed 정보(pubMed_id)가 있다 [11,12].

5.4 DIP 데이타베이스의 변환

이 연구에서는 DIP에서 제공하는 XML 문서를 파싱하여 관계형 데이타베이스를 구축하였다. 그림 15는 DIP의 정보를 관계형 데이타베이스로 변환한 데이타베이스 스키마이다. 이 스키마에 대하여 대략적으로 설명하면, Edge 테이블은 Interaction ID(e_uid)와 Interaction에 참여하는 protein 정보(e_from, e_to) 등의 단백질 상호작용 정보를 가지고 있다. Node 테이블은 Interaction에 참여하는 개체에 대한 테이블이고, 개체의 이름(n_name), 타입(class, ex : protein), 설명(descr), reference (swiss-prot, pir, ncbi), taxon 등의 정보를 가지고 있다. Edge_Feature 테이블은 Interaction에 대한 PubMed ID(ef_src), 실험정보(ef_val) 등이 있다.

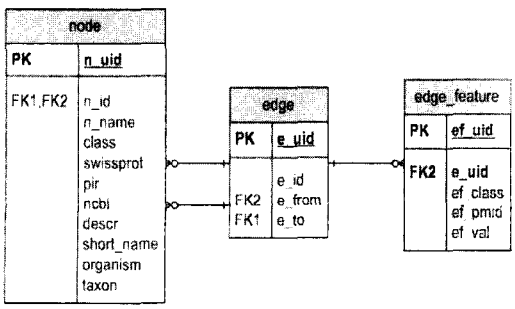

그림 15 DIP 데이타베이스

6. 결론 및 향후 연구

이 논문에서는 유전자 상호작용 데이타베이스를 위한 SOAP 서버 객체를 정의하였고 프로토타입 구현을 통해서 그 효용성에 대해 검토하였다. 이 논문에서 정의한 객체 및 관련 메소드는 과거의 연구와 관련 데이타베이스의 SOAP 서버 객체를 분석하여 도출한 요구사항을 만족한다. 따라서 여기서 정의한 SOAP 서버 객체는 실제 연구자들이 원하는 메소드를 모두 제공함으로써 데이타 제공에 효율적일 뿐만 아니라 연구자는 유전자 상호작용 데이타베이스의 스키마 변동이나 데이타 갱신 시 쉽게 활용할 수 있을 것이다. 향후에는 여기서 정의한 모델 및 메소드의 정당성 및 유용성을 이론적으로 증명하는 연구를 할 필요가 있다. 아울러 새로운 Pathway를 찾는 도구 등을 개발하여 이 객체의 유용성을 증명하는 연구가 진행될 수 있을 것이다.

현재 BIND, MINT, DIP 데이타베이스 간의 참조를 위해 만든 참조 데이타베이스는 자동으로 업데이트가

되지 않는다. 향후 이를 자동으로 구축하는 방안에 대해서도 고민하여, 사용자가 BIND, MINT, DIP 데이타베이스를 연동하여 이용하기 위한 최신 데이타를 제공할 수 있도록 해야 한다.

참 고 문 헌

[1] Lincoln D. Stein, Creating a bioinformatics nation. Nature, 417, 119-120, MAY 2002.

[2] H. Sugawara & S. Miyazaki, Biological SOAP servers and web services provided by the public sequence data bank, Nucleic Acids Research, 31, 3836-3839, APRIL 2003.

[3] Lincoln D. Stein, INTEGRATING BIOLOGICAL DATABASES, NATURE REVIEWS : GENETICS, 337-345, MAY 2003.

[4] http://www.kegg.com

[5] Shuichi Kawashima & Toshiaki Katayama & Yoko Sato & Minoru Kanehisa, A Web Service Using SOAP, WSDL to Access the KEGG System, Genome Informatics, 14, 673-674, 2003.

[6] http://xml.nig.ac.jp

[7] http://www.ncbi.nlm.nih.gov

[8] http://www.ebi.ac.uk

[9] www.bind.ca

[10] http://dip.doe-mbi.ucla.edu

[11] http://cbm.bio.uniroma2.it/mint

[12] Gianni Cesareni&Mario Gimona, MINT : a Molecular INTeraction database, A. Zanzoni et al./FEBS Letters 513, 135-140, December 2001.

[13] http://psidev.sourceforge.net

[14] Min Kyung Kim, Hyun Seok Park and Seong Joon Yoo, "Crosstalk Between Metabolic and Regulatory Pathways," GIW2003, December 2003.

[15] 이호일, 상호작용 데이타베이스의 SOAP 서버를 위한 객체 모델 설계 및 구현, 석사 논문, 2004년 8월, 세종대학교.

[16] Bader, G.D. & Hogue, C.W, BIND-a data specification for storing and describing biomolecular interactions, molecular complexes and pathways. Bioinformatics 16, 465-477, MARCH 2000.

[17] Cheryl Wolting & Cris Alfarano & Brian Bobechko & Cheryl D'Abreo & Vicki Lay & Susan Moore & Brigitte Tuekam, BIND Curation Reference Manual, BIND, 2003.

이 호 일

2002년 세종대학교 전산학 학사. 2004년 세종대학교 컴퓨터공학 석사. 2004년 4월~현재 ㈜마크로젠 연구원

유 성 준

1982년 고려대학교 전자공학 학사. 1982년~2000년 한국전자통신연구원. 1990년 고려대학교 전자공학 석사. 1996년 Syracuse University 전산학 박사. 2002년 3월~현재 세종대학교 컴퓨터공학부 조교수

김 민 경

1993년 이화여자대학교 생물교육학과 이학사. 1998년 서울대학교 의과대학 의학과 의학석사. 2001년 서울대학교 의과대학 의학과 의학박사. 2003년 3월~현재 이화여자대학교 공과대학 컴퓨터학과 연구전임강사