ENHANCED BIOINFORMATICS DATA MODELING CONCEPTS AND THEIR USE IN QUERYING AND INTEGRATION

by

FENG JI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2008

To my father Guoliang Ji.

ACKNOWLEDGEMENTS

I am forever grateful to my advisor Dr. Ramez Elmasri for his expertise, constant guidance, encouragement and support towards the successful completion of my doctoral degree. I also want to thank my committee members, Dr. Nikola Stojanovic, Dr. Jean Gao, Dr. Gautam Das and Mr. David Levine for their time to participate in my committee and providing additional insightful comments into my research.

I thank my fellow labmates at BioLab, Jack Fu, Weimin He, Qing Li, Yiming Zhang, Kyungseo Park for creating an interesting and enjoyable environment to work where I have spent endless hours of my time, and for the research and publications we have worked together. The discussions and team work have played important parts in my studies.

I would like to thank my family with deepest gratitude, especially my parents for being a role model and brought me up to the person who I am today. Their unconditional love and support has helped me through my studies. Especially, I want to thank my father, Prof. Guoliang Ji for his support and encouragement, who has inspired me in the scientific research area. Lastly, and most importantly, I want to thank my wife Renying and son Albert for their sacrifice and patience. I couldn't have done it without their support and understanding.

August 4, 2008

ABSTRACT

ENHANCED BIOINFORMATICS DATA MODELING CONCEPTS
AND THEIR USE IN QUERYING AND INTEGRATION

FENG JI, Ph.D.

The University of Texas at Arlington, 2008

Supervising Professor: Ramez Elmasri

In bioinformatics research, scientists usually face the problems of modeling complex data types and integrating diverse resources. Traditional data models such as EER lack the expressing power to capture many characteristics that are common in bioinformatics data. We first propose extensions to the ER model that allow accurate representation of many of these characteristics. We then utilize these concepts in an integrative system to provide an easy-to-use interface for biologists to construct queries. Our research utilizes the enhanced conceptual modeling concepts to create a prototype mediator for querying multiple data sources. The various relationships between different biological entities are all semantically represented as domain ontologies stored in the mediator for experts to analyze and correlate the integrated query results. The following research has been conducted: (1) We first propose new EER schema notation to represent the common occurring biological concepts: the ordering properties of the DNA sequences, the 3D structure of proteins and the functional processes of metabolic pathways. (2) Then, we utilize these new relationships

in the development of the mediated domain ontology, which helps the interface design and query processor implementation of our mediator system.

Our mediated schema features are based on a hybrid of taxonomy ontologies (core concepts and external classification/annotation concepts) for interpretation of raw data sets (protein and gene sequences) in the context of molecular interactions, biochemical pathways and biological processes. We adopt the RDF data model to implement the mediation data. Our mediator mainly takes a browsing-based approach to integrate different data sources. Extra data can be dynamically retrieved through the web service. By browsing the ontology tree in the query interface, users can select concepts of interest and associated attributes to formulate queries based on their domain knowledge. The query result is a set of various database entry accessions with associated attribute values. Users can click each link of the accessions to see the detailed reports, or cross-compare attributes of these data instances. Query usability and performance experiments are tested for real data sets from UniProt [30], ENZYME [8], CATH [23], and GO [29].

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

A computer system, and a biological system, which one is more complex? Probably, most people would consider the biology system to be more complex. But before drawing a definite a conclusion, we must face the problems of effectively querying and integrating the huge amount of bioinformatics data in the post-genomic era [67]. So, what is this bioinformatics data? The core (raw) bioinformatics data should include these data sources: nucleotide sequences, protein sequences, and 3D chemical structures of these micro-molecules. All these data sets are at the molecular level, that is, their existences can be verified, and their properties can be determined experimentally. There are many levels of data beyond this core data set, such as the structure features of genomic DNA sequences, the various functions of proteins, gene expression patterns, and pathway models. These types of data sets can be roughly considered as describing various functions and processes of the core data set (see Table 1.1).

Data generated from life science domain are inherently complicated, highly heterogenous, and updated very frequently. Their representations are syntactically

Table 1.1. Bioinformatics data sets

Core data	Examples of high level of functions/processes
nucleotide sequence	gene expressions, chromosome, genome organization
protein sequence	protein families and interactions, pathway models
3D structure	domain (tertiary structure), complex (quaternary structure)

different, semantically diverse, and involve deep understanding of domain concepts and relationships [19].

Currently, the publically available databases holding the above data sets amount to more than 1,000 [59]. Not only does the total number increase each year, but also the categories of these databases increases, not mentioning the amount and the complexity of the above data sets stored there. This phenomena definitely brings us the problems of how these data can be accurately modeled, efficiently organized, and effectively queried.

1.1 Motivation

To date, most biological databases adopt the (object-)relational database system. Major data repositories (EMBL [43], GenBank [13], and UniProt [30]) can provide their database entries in many forms from legacy flat-file to more advanced XML/RDF. The XML data exchange format offers a way to mix plenty of "metadata" with the raw data represented in a document tree structure. Clients can apply various processing tools to extract what they want. But all these endeavors can not solve the problem of interoperation permanently under a distributed computing environment. Different systems are installed with different Database Management System (DBMS) to store different types of data. Most importantly, each schema designer has different focus and view, even for the same set of data. For example, to store the nucleotide sequence related to the *gene* concept, some people use entity type Open Reading Frame (ORF), meaning the coding part of DNA sequence, while others use Transcript, meaning the transcribed part of DNA sequence. So, good data modeling, querying, and integration methodologies are desperately needed to make this data-rich research area progress more rapidly.

In this dissertation, we propose solutions to some of the problems related to modeling and integrating bioinformatics data. The next two sections discuss modeling and integration problems.

1.1.1 Bioinformatics Data Modeling Concepts

Many scientists build their biological databases to store their experimental data using relational database systems such as MySQL™or Postgres™. But concerning the complexity of the data, traditional conceptual data modeling techniques lack the intuitive expressing power to accurately represent them. Some work has been done on their conceptual enhancement [16].

In modeling biological data we consider three frequently occurring concepts: sequence ordering, input/output processes, and molecular spatial structure. Sequence data, such as nucleotides in DNA/RNA and amino acids in proteins, have this order property in their physical constructs. Important biological processes such as gene expression, metabolism, cell signaling and bio-chemical pathway regulation all involve ordered events and input/output processes. The biological functionality of these entities are totally determined by their internal molecular structures and various external interactions.

Because of the importance of these relationships, there is a need to model them. Database conceptual models, such as the widely-used Entity-Relationship (ER) model and Enhanced-ER (EER) model do not easily represent these commonly occurring concepts from the bioinformatics domain. This is because traditional database applications do not require these concepts. Although ordering can be incorporated into relationships by adding one or more relationship attributes, this would complicate the schema and would make it difficult to identify the ordered rela-

tionship. It is preferable to have explicit and clear representation of such important and frequently occurring concepts.

A major part of this research is to extend ER and EER modeling to represent bioinformatics data more accurately.

1.1.2 Querying and Integration Problems

Data integration in the domain of life science research means to obtain data from diverse resources to test and validate a researcher's point of view. So, it is a hypothesis-driven process of data analysis.

A typical example is to characterize an unknown protein's biological functions. The starting point is the newly determined protein sequence. The initial result is a list of similar proteins (or homologs) predicted by similarity search tools such as BLAST. These proteins already have determined molecular structures or annotated functions in various biological reactions, pathways or processes. Then, various information must be gathered from multiple online databases such as PubMed [46], GenBank [13], UniPort [30], PDB [68], BIND [42], and KEGG [50]. The researcher has to interpret many types of data from these variety of sources: literature review, sequence alignment results, gene expression profiles, protein interactions and families, 3D domain structure, putative metabolic pathway models, and many others. Unfortunately, each type of data is not easily identified and accessed because of the variety of semantics, interfaces, and data formats used by the underlying data sources. Based on collected relevant data sets, further experimental characterization of this unknown protein can be carried out.

One problem closely associated with the data integration is data querying. Consider this query example [92]:

```
"A corpus of micro-array experiments has revealed certain chemicals
```

to increase the expressivity. We wish to limit those involved in the "apoptosis" pathway. We further restrict to those chemicals which affect proteins either very much or very little. Given that set of chemicals, we look for chemicals with similar side-chains that have a minimum toxicity in mice."

Even though each data source can provide a sophisticated query form in its web interface, no individual data source can answer the above query directly due to limited query capabilities of the data sources [88]. If these data sources can provide interfaces to directly access their data without restriction, it is also hard for the biologist to get familiar with so many different interfaces, and to specify a query from his point of view. Different integrated views can be returned with different queries against the same raw data set. So the integration system should provide an easy-to-use query interface for scientists to design the meaningful queries that fully utilize their domain knowledge.

Another major part of this research is to provide an easy-to-use but powerful querying interface to diverse bioinformatics data sources.

1.2 Contributions

In this thesis, we advocate EER diagrammatic enhancement in modeling bioinformatics core data. We proposed new schema constructs to represent the commonly occurring biological concepts, and utilize these new constructs in the formalization and construction of our mediated domain ontology, and the development of the mediator system. Based on this mediated ontology, a powerful and easy-to-use query interface is developed to access data from multiple diverse sources.

1.2.1 EER Data Model Enhancements

In order to accommodate the special features of DNA sequences, protein structures, and metabolic pathways, we suggest significant yet minimal changes to the EER model by introducing three special types of relationships: the *ordered relationship*, the *process relationship*, and the *molecular spatial* relationship. In addition, many relationships in bioinformatics require duplication of instances in ordered relationships, so we also propose extensions to allow *multisets* (or bags) of relationship instances, where needed. Although the notational changes are minimal, they enhance the modeling power to directly capture these concepts. We also give the formal definitions for these constructs. We show how these EER extensions can be mapped into relations for implementation in relational databases such as ORACLE™or MySQL ™. Also these extensions have been used to facilitate work on the development of ontology and mediator systems as well as data mining and processing tools.

1.2.2 Mediated Domain Ontology

We proposed the Mediated Domain Ontology (MDO) for the purpose of bioinformatics data query and integration. The MDO consists of two sets of concepts: the core, instance-supported domain concepts, and external, instance-associated annotation/classification concepts. The domain concepts in our proposed MDO can be roughly classified into 3 types: entity concepts, attribute concepts, and relationship concepts. Core entity concepts such as *Protein*, *Nucleotide*, *Structure*, *Interaction*, *Reaction*, *Pathway*, *Process*, *BioScource* and *DataScource*, are manually set up by the analysis of various molecular database entries. Attribute concepts include the common attributes of the above entity concept instances, and standard annotation concepts. External annotation concepts such as in Gene Ontology (GO) [29] and Chemical Entities of Biological Interest (ChEBI) [31] can be queried and down-

loaded automatically through the web service provider based on the user needs. The MDO are a hybrid of several taxonomy ontologies. We apply the Resource Description Framework (RDF) data model in the design of mediator schema. Our mediated schema features are based on a hybrid of taxonomy ontologies for integration of protein and gene instance data in the context of interaction, pathway and process.

1.2.3 BioMediator Querying System and Browsing Interface

To test the efficacy of our conceptual modeling extension and explore its usefulness to the construction of mediated ontology for diverse domains, a prototype BioMediator system has been built for biologists to navigate the domain concepts and make queries. The system pre-stores the domain concepts about bioinformatics data sources and their database entries. The users can formulate the queries by browsing the concept tree and selecting the specific concept of interest and its attributes and related concepts. After submitting a query, the system will return a list of accession numbers from the different data sources. Users can click the link to check the detailed report of that data entry. Extra attribute data will be retrieved by the web service.

1.3 Thesis Organization

The remainder of the thesis is organized as follows. In chapter 2, we first discuss the characteristics of bioinformatics data. Then, we provide several examples of biological concepts, and motivate the data modeling needs for these new concepts. We also discuss some issues concerning conceptual data modeling, limited query capacity of web interface, and some related work on integrative systems in bioscience domain. Chapter 3 describes the details of our proposed schema extensions to the EER data model. It includes the formalization, applications on molecular biology

system, and EER-to-relational mappings. In chapter 4, we first introduce some representative bioinformatics ontological concepts, and discuss how they can be used to help querying and integrating the data resources. Then, we give the details of the mediated domain ontology we proposed. It includes the formal definitions, ontology construction, and customized view of domain concepts. Chapter 5 gives the details of BioMediator system architecture. It includes the domain ontology server, user interface for browsing and querying, and query processor. The functionalities of each component are explained. We give an example to show how the user query with specified relationships can be mapped to internal SQL queries, and extra data can be retrieved via external web service provider. Chapter 6 shows how to use ontology browser query interface to formulate queries. We illustrate 2 types of queries with biological examples. Finally, we conclude this thesis with future research directions in chapter 7.

CHAPTER 2

BACKGROUND

This chapter gives the background of bioinformatics data management. Section 2.1 gives an overview of biological data sources, various data types, and their characteristics. Section 2.2 surveys some work related to our research.

2.1 Overview of Biological Data Sources

This section gives the overview of biological data sources, what kinds of data they store, and some basic molecular biology concepts necessary to understand and use these resources. We first classify the bioinformatics data based on biological concepts, then discuss their unique characteristics when modeling them in the application of data management.

2.1.1 Data Types and Databases

Each year, the journal *Nucleic Acids Research* has a special issue, which gives a list of molecular biology databases freely available to the public. The 2008 update [59] includes 1078 databases, which contain data covering various aspects of research projects in life science throughout the world. Even though these data sources are diverse, we still can tell them apart by their stored data types. Below, we give an overview of some these data sources.

Sequences. At the molecular level, there are 2 major types of data sources: polypeptide and nucleic acid. Because they are molecules of biopolymers, the major data type stored in archives are sequences. Protein sequences (protein are larger

or fully functional forms of polypeptides) are strings of 20 amino acid letters, while nucleic acid sequences are strings of 5 nucleotide letters. The major repositories of DNA and RNA sequences are GenBank of NCBI (National Center for Biotechnology Information) [13], Nucleotide Sequence Database of EMBL (European Molecular Biology Laboratory) [43], and DDBJ (DNA Data Bank of Japan) [101]. The major repository of protein sequences is UniProt [30], which is union of protein sequences in Swiss-Prot [5], and TrEMBL (Translated EMBL) [6], and PIR (Protein Information Resource) [24].

Structures. The major database for biological macromolecular structures is the PDB (Protein Data Bank) [68]. It contains 3D structures of proteins, nucleic acids, and a few carbohydrates. The main contents of PDB protein structure data file include the atomic coordinates, the amino acid sequence, the secondary structure feature, experimental conditions, etc. We discuss the details of protein 3D structure model in chapter 3.4.2. The sequence and structure data are so fundamental that they constitute the major part of raw (core) data sets of bioinformatics. These core data are further interpreted into different higher levels of data in the form of annotations.

Interactions. Molecules can interact with each others. These interactions can be the protein-protein or the DNA-protein interactions. There are many other forms of interactions such as binding, docking, protein modification, and chemical crosslinking. These interactions are important for many biological functions. For example, the process of signal transduction involves the binding of extracellular signaling molecules to cell-surface receptors to trigger ordered sequences of biochemical reactions inside the cell [44]. BIND (Biomolecular Interaction Network Database) [42], DIP (Database of Interacting Proteins) [81], and IntAct [56] are major databases

for the interaction data. The main contents include identifiers of two interactors, their types, the interaction type and the detection method.

Reactions. Biomolecules can participate in various enzymatic reactions. The contents of the reaction data includes participating molecules (reactants or substrates, products, catalysts, cofactors), the equation, the reaction type, reaction kinetics, and associating pathways. Their representation are more intuitive in a graphic form. KEGG (Kyoto Encyclopedia of Genes and Genomes) Reaction [50], BRENDA [70] and Reactome [47] contain the various reaction data.

Pathways/Processes. Pathways are networks of molecular reactions. In general, they illustrate the functional relations between molecules. All pathway databases such as KEGG Pathway, MetaCyc (encyclopedia of metabolic pathways) [53], and Reactome [47] also include the reaction data. So, the pathway data content is more comprehensive and complicated than the above data types, and the best way to present it is also in a graphic form.

Genomes. Genomic-scale data is about the complete genomes of various organisms. The genome data includes characterization of repeats, structural assignments of genes (chromosomal position of genes), associated phenotype or diseases. The major data sources are NCBI Genome DataBase and Ensembl [51]. Some tools are developed for the exploration of these data at different levels. Ensembl Human Genome Browser, UCSC Genome Browser [45], and NCBI Genomic maps are good examples.

Expressions. Molecular biology central dogma says: DNA makes RNA, and RNA makes protein. Most bioinformatics data is generated during this biological process. Nucleotide and protein sequences are the input and the output data of this biological "pipeline" respectively, while gene expression data (transcriptomics and proteomics) are the data generated in the intermediate steps. ArrayExpress [49],

GEO (Gene Expression Omnibus) [11], SMD (Stanford Microarray Database) [84], and PRIDE (PRoteomics IDEntifications database) [74] are all good sources of these data. The main contents include genes, proteins, species, cell types, experiment types and protocols.

Annotations. The annotation is defined as semantically rich meta-data applicable to a particular data item. Usually, these data items are protein and nucleotide sequences or genes. Thus, the annotations include the descriptions of various structural features of sequences and functions of genes and proteins in the cells/tissues of the organism. Currently, annotations contribute to the development of different biological ontologies. Representative databases are ChEBI (Chemical Entities of Biological Interest) [31], SO (Sequence Ontology) [34] and GO (Gene Ontology) [29]. They are widely used as public repositories of biological knowledge.

References. Another type of data of great importance is literature references. The contents include titles, authors, journal names, publication dates, and so on. PubMed [46] is such a bibliographic database that offers free abstracts of scientific articles.

Different types of data can be combined into specialized databases such as organism-specific databases and organelle databases, but each one has a different focus and research interest. Actually, major data repositories such as NCBI and EMBL are so comprehensive, containing all types of data. Table 2.1 is a summary of the above mentioned bioinformatics resources. In chapter 3, we will discuss our work on conceptual modeling of the gene sequence, the protein structure, and the biological pathway/process data.

Table 2.1. Bioinformatics resources

Data Type	Data Source
Sequence	GenBank, EMBL Nucleotide, UniProt, DDBJ
Protein	UniProt, SwissProt, NCBI Proein
Structure	PDB, Entrez MMDB [46], EBI MSD [43]
Interaction	BIND, IntAct, DIP
Reaction	KEGG Reaction, BRENDA
Pathway	KEGG, aMAZE [71]
Expression	ArrayExpress, GEO, SMD, PRIDE
Annotation	ChEBI, ENZYME, SO, GO, CATH, SCOP [7]
Reference	PubMed

2.1.2 Data Characteristics

Life science data differs greatly from traditional business data in many dimensions (see Table 2.2). First, the bioinformatics data come from many different domains such as chemistry, biological, biomedical and clinical areas. The structure of data semantics is very complex and fast evolving. The data often bares temporal and spatial properties, which are governed by underlying physical or chemical principles. The data sets produced by various research areas are often incomplete and the values can be imprecise. For example, even though the human genome has been sequenced out, the exact number of genes is still unclear.

Second, the bioinformatics data sets are a mixture of experimental facts and domain concepts. Some data are raw data types, which come directly from the experiments, such as the DNA sequences or the protein structures. Other data are annotation types (domain expert curations), which contains various domain concepts, such as functions of the specific gene or protein.

Third, many-to-many relationships between bio-entities are very common in the genomic and clinical domains. For example, a gene instance can encode several protein instances, and a protein instance can also be encoded by several gene

Table 2.2. Characteristics of bioinformatics data compared to business data

Life science data	Commercial data
Complex data semantic	Easy-to-understand data semantic
Span over diverse domains	
Mixture of instances and concepts	Clear border between data and meta-data
M:N relationships between entities	1:N relationships between entities
Many potential relationships between data instances	Clear relationships among objects
Incomplete and/or imprecise data	Complete and/or precise data

instances. Many of these m:n relationships come from self-referencing of the same entity type. For example, the protein-protein and gene-gene interactions are the most important relations. The cardinality problem of 1:n or m:n depends on the abstraction level of the involved instances. For example, if we model the molecular reactions as 2 entity types: REACTION and MOLECULE. There are 2 types of relationships between them, Reactant and Product. The cardinality of Reactant (or Product) can be m:n or 1:n. If the instances of REACTION and MOLECULE are both individual reactions and molecules, then the cardinality is m:n. If the instances of REACTION are at the level of the reaction class, such as the redox or nuclear transfer reaction, and the instances of MOLECULE are still individual molecules, then the cardinality is 1:n. An individual molecule with its unique property can only participate in one specific class of reactions.

Finally, if we model two protein instances at the same abstraction level, there could be many potential relations of different types between these two protein instances, such as different types of domain binding interactions. In the business data, the relationship between data instances is already captured at the schema level.

The above characteristics should be considered as a critical requirement for conceptual modeling for the data management of life science research. The existing data models do not fully support these requirements. We should enhance the current conceptual data models by making their semantic notation more explicitly powerful.

2.2 Related Works

This section reviews some work related to our research. Section 2.2.1 focuses on the conceptual data modeling of the above mentioned data types. In section 2.2.2, we first discuss the database entry mapping problem of integrating bioinformatics resources. Then, we review some relevant work. Finally, we discuss some querying problems associated with data integration. In section 2.2.3, we overview several integration approaches, and discuss some representative integrative systems that employed these approaches.

2.2.1 Conceptual Data Modeling

Developing a successful biological database needs a clear understanding of the nature of the available data. The questions such as: "What different types of data will be stored? What are the available properties for each data type? What are the obvious and potential relationships between these data types?" must be answered before the real implementation. Conceptual data modeling can provide a scientific way to capture the principal structural properties of data. Of course, the designers of the above mentioned databases already have done extensive works on conceptual modeling of their data. We will give several examples of this research using data models such as Entity-Relationship (ER), Unified Modeling Language (UML), Extensible Markup Language (XML) and Resource Description Framework (RDF) in the application of bioscience data management.

Chen et al. [26] present a *genomic schema element* data model to represent basic biological notion *Sequence* and sequence features. It includes how to model a sequence (a single entity), a gene (a linked list of splicing units), a chromosome (a linked list of contig sequences), and a pairwise sequence comparison (a graph with each sequence in comparison as a node and each similarity hit as an edge). They also present a *genomic schema fragment* data model to represent only one genomic topic area. It includes how to manage sequence similarity search, sequence clustering, etc.

Paton et al. [91] present a collection of conceptual models for genomic sequence data and genome organization. The models are described using the class diagram notation of UML (objected-based model), which can fully model the generalization/specializaton relationships existing in the genome data. For example, *Centromere* and *Telomere* are both subclass of *Chromosome Fragment*.

Macromolecular structure data (include its associated experimental data) has its own standard archive format for data deposition, i.e. mmCIF (macromolecular Crystallographic Information File) [105]. The original PDB format like other sequence database entry format, is a structural flat file with fixed field length. This format cannot be treated as a data model, and creates problems when being queried and updated. So, mmCIF is a true, proprietary data model for structure data. But with the popularity of XML data exchange format, the content of the mmCIF dictionary has been translated into XML schema, and mmCIF data files into XML [17].

Helden et al. [71] present a general model for the physical and functional interactions between genes and proteins as forming a large complex network, and this data model has been implemented in aMAZE database. The model is based on ER, and uses an Object-Oriented (OO) representation. Several main classes: *Biochemical-Entity*, *Interaction*, and *Location* are defined. Subclasses such as *Compound*, *Gene*,

Protein, *Tissue*, *Cell*, *Organ*, *Transformation* (*Reaction*, *Expression*, *Translocation*, etc), and *Control* (*Catalysis*, *Transcriptional-Regulation*, *Activation/Inhibition*, etc) are also defined to describe biological pathway knowledge at all levels.

Keet [77] discusses the characteristics of biological data and its effect on ER, OO and Object Role Modeling (ORM) methodologies. General features of ER, OO and ORM are discussed, emphasizing differences in graphical representation, understandability and inclusiveness of types and attributes in the model.

Scientific domain data models such as Object Protocol Model (OPM) are also proposed for some specific applications [25]. The ONION framework incorporates sequences in RDF methodology [89].

Even though much work has focused on applying the various modeling techniques, little work has been done on conceptual data modeling enhancement. Ram et al. [93] propose a semantic model for 3D protein structures by adding spatial semantics and constructs to represent the contributing forces such as hydrogen bonds and high-level structures such as protein secondary structures. But their enhancements require many additional constructs and notation. The EER model still lacks the expressive power to explicitly represent the biological concepts inherent in the sequence, the structure, and the pathway/process data. The next chapter will present our solutions to this problem.

2.2.2 Querying and Integration Issues

Scientists in life science often need to retrieve different types of data from diverse biological data sources to solve their research problems. As we discussed in the previous sections, most data sources have different goals for selecting the data to store. They vary in the type of the stored data, the archiving data file format, and access methods. Even for the same set of data, different data modelers have different

focus and views. In addition, there is a terminology discrepancy at the data level and at the schema level [72]. Data integration is an old problem in the computer science domain [83, 65]. In bioinformatics domain, there is one unique problem: databases entry redundancy and inconsistency for the same biological entity among multiple databases. This becomes an obstacle to further instance level integration. Many work seek a universal agreement on identification of these biological entities (proteins, genes, mRNAs, etc.). Many efforts have been done to achieve this goal.

The IPI (International Protein Index) has been developed to address the problem of protein redundancy in different databases [78]. IPI effectively maintains a database of cross references between the primary data sources such as Swiss-Prot, TrEMBL and Ensemble. IPI is created by using sequence comparison to identify entries from the source databases that represent the same protein.

Another work is the proposal of Life Science Identifier (LSID) for uniquely naming biologically significant resources including species names, genes or proteins [4]. Similar to a URL, LSID uses a uniform resource name (URN) to locate data. An example is:
`urn:lsid:ncbi.nlm.nih.gov.GenBank:G54036:2`. The last number denotes the version number of that object. So, given the LSID, one can get some meta-data for that gene. This could be useful in the application of semantic web technology.

Much work focuses on the mapping or the cross-referencing between different database entries for the known proteins and genes [32, 22, 20]. Draphici et al. discussed the problems of name space inconsistences between existing annotation databases, and developed a tool to map biological entity IDs from one database to another [33]. To accomplish the mapping, they also defined the authoritative database sources for different types of biological entities. For example, UniProt for proteins, PDB for protein structures, GenBank for nucleotide sequences, UniGene

for nucleotide sequence clusters, Entrez Gene for genes, KEGG for pathways, and OMIM [46] for diseases. Martin et al. presented a work on mapping between UniProt entries and PDB entries at the chain or residue level [85].

The querying problem is closely associated with data integration. Some biological queries are very complex. Consider this query example: "Find all genes in the human genome that expressed in the liver, and have a TTGGACAGGGGAA followed by GCCGCC within 40 symbols in a 4000 symbol stretch upstream of the gene" [100]. The query involves a similarity search, an important operation that is heavily used for both protein and nucleotide databases. Currently, no commercial database system directly supports this kind of query. There are other similar types of searches often used in the bioscience research, such as chemical structure search [106].

Also, this query involves many data types: the sequence, the sequence structure, the gene, the expression, and the genome. There are many relationships between these data types. For example, a gene is part of a DNA sequence, a gene express in a cell, and so on. These all must be clearly defined in the underlying database schema, and presented in a form-based search interface. This query form will become very complicated with all the available data types and their attributes. But this issue will become worse if the biologist faces many different query forms and to specify a query from his point of view. Different integrated data results can be returned with different queries against the same raw data set. If a high-throughput experiment needs a batch of similar queries, the process of data retrieval requires extensive human interactions with multiple data sources, which calls for for automating query and analysis tools [21, 95].

More recently, web service technology has become a new trend for gathering information. Many bioinformatics data resources and analysis tools can be program-

matically accessed through various web services, such as NCBI Entrez Utilities [3] and EBI Web Service [82]. However, it is not easy to integrate and obtain a concise and complete query result among hundreds of overlapping web operations [35]. An integrative system based on web service can offer a framework for automating the process of data analysis [62]. But this requires an easy-to-use query interface and a "global" integrated data schema. Thus, the scientists can design the meaningful queries that fully utilize their domain knowledge.

2.2.3 Integrative Systems

In the past decade, several commercial and non-commercial integration systems for life science resources have been proposed and developed [61, 57]. Generally speaking, the architecture of these systems can be classified into 2 types: the data warehouse and the mediator-wrapper. Semantic integration using ontologies are infiltrating into both structures, which focuses on the sharing of controlled terms or domain concepts. Each architecture has its advantages and disadvantages. The data warehouse approach is good at intensive data analysis in some specific domain while the mediation approach can provide fresh data with little maintenance.

COLUMBA [103] data warehouse is a database of annotated protein structures. It integrates twelve different databases, including PDB, KEGG, Swiss-Prot, CATH, SCOP, GO, and ENZYME. It addressed the problem of integrating protein structures (in PDB) and protein structure annotations (in CATH, SCOP, GO, and ENZYME). The database can be searched using either keyword search or data source-specific web forms. The results of queries are PDB entries with the corresponding GO annotations and CATH architecture.

TAMBIS [61] is a database federation that is based on a mediator-wrapper architecture. It has a domain ontology and a reasoning system over this ontology. The

biological ontology is built using Description Logic (DL), a knowledge representation languages. TAMBIS provides a user interface for browsing the ontology and for constructing queries. The ontology and the associated reasoning component guide the construction of the query, ensuring that only a query that is logically meaningful can be formulated. BACIIS [12] also uses mediator-wrapper approach to support semantic integration of life science databases through their web interfaces.

In recent years, applying semantic web technology is a relatively new way to integrate the bioinformatics data [96]. YeastHub [27] is such a system. It features the construction of a RDF-based data warehouse for integrating a variety of yeast genome data. Multiple data sets need to be registered first before RDF queries can be composed to retrieve related data across different data sets.

NCBI Entrez [86] takes the link-driven approach to integrate its internal sub-databases and various external databases. The data sets distributed in these databases can be browsed on the web from different access points: the genome-centric view, the gene-centric view, or the specie-centric view.

CHAPTER 3

DATA MODEL EXTENSION

This chapter gives the details of our works on data model enhancement to meet the needs of conceptual data modeling in bioinformatics domain. The whole chapter is organized as follows. In section 3.1, we provide several examples of sequence ordering, input/output processes and molecular spatial structure and we motivate the data modeling needs for these new concepts. In section 3.2, we give formal definitions for ordered, process and molecular spatial relationships to enhance the modeling features of the ER/EER models. Section 3.3 summarizes the new EER notations for these relationships. Section 3.4 gives the details of the EER conceptual schema for the molecular biological system that utilize these new constructs. Section 3.5 describes mapping techniques that can be used for the implementations of our new EER constructs using relational databases. Section 3.6 summarizes the chapter.

3.1 Examples of Biological Concepts

In this thesis we focuses on the biomolecular subset of bioinformatics data. Closed related to these are the familiar concepts of sequence ordering, multisets, input/output processes, and molecular spatial structure. In the subsequent sections, we will give 3 motivating examples of modeling biological data at the conceptual level, and illustrate the data modeling need for these new concepts.

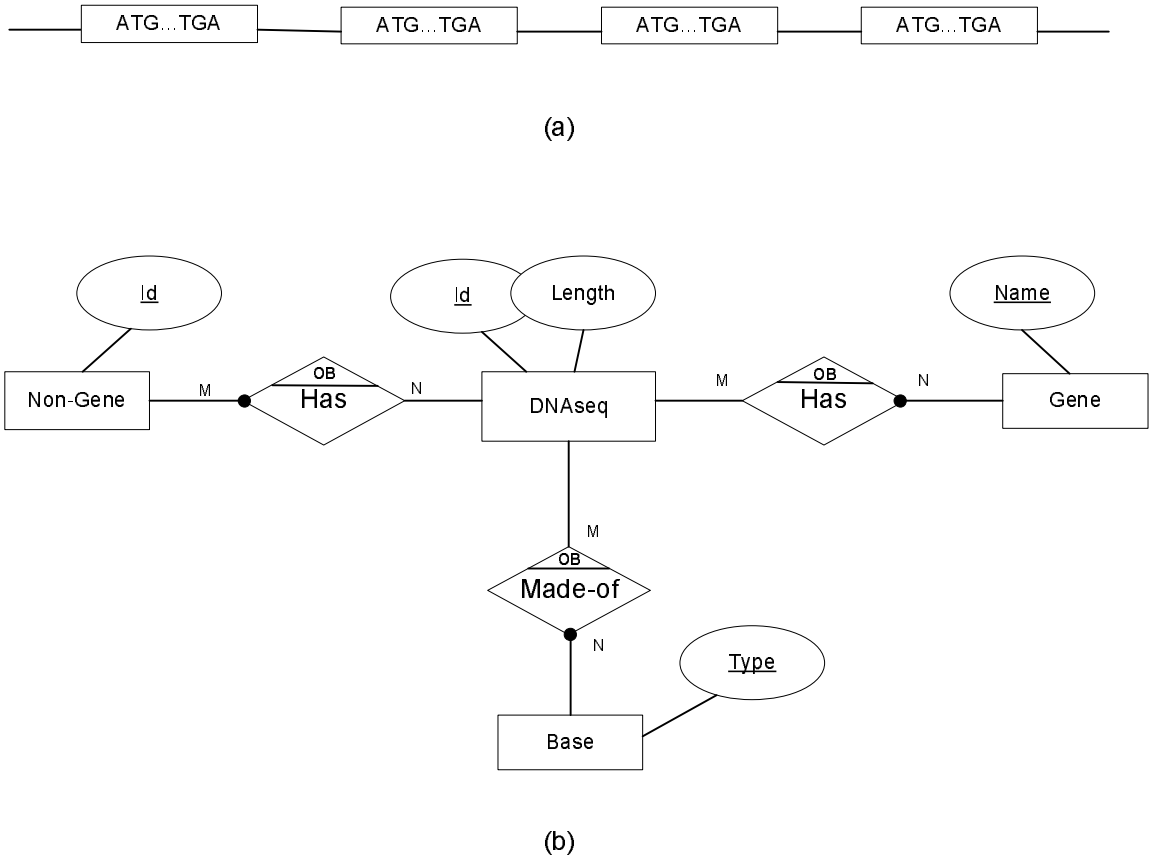


Figure 3.1. (a) A DNA sequence (b) EER model of DNA, gene and base.

3.1.1 Sequence Ordering Concept

Molecular structural data includes linear nucleotide sequences of DNA (genes, intergenic and regulatory regions), and the linear amino acid sequences (proteins) resulting from gene expression. They are internal properties of biological entities (in contrast to external properties such as environment), and although both genetic and protein sequences can change slightly (the basis of evolution), for modeling purposes it is reasonable to treat them as static.

EXAMPLE 1. *Figure 3.1 shows the biological data of DNA sequence, genes and their EER conceptual modeling. A DNA is an ordered sequence of bases A, T,*

C, and G. A gene is one segment of a DNA sequence. Different genes may or may not concatenate to each others. Some genes can be repeated in a DNA sequence; hence, both order and repetition are needed to model this. We model DNA-Base and DNA-Gene as an ordered bag (multiset) relationship.

First, the ordering of elements in these sequences is their most important feature because changes to that order are likely to impact higher levels of structure and therefore also function. In EXAMPLE 1, protein-coding genes are segments in a DNA sequence, as shown in Figure 3.1(a). Boxes and lines denote genes and intergenic regions, respectively. Each triplet of the bases A, T, C and G in these genes, determines one amino acid in the proteins they encode, a single change to one base can dramatically impact protein function, the classic example of this is sickle cell anemia. Obviously, in order to capture the ordering relation between DNA and genes, we need a special relationship for ordering features (symbol O denotes Ordering) in EER models. In addition to the ordering of base pairs, ordering of sequence subsets (genes and intergenic regions) relative to one another is also important to model. Figure 3.1(b) is the EER schema for DNA, gene and intergenic entities in our extended notation. We represent their relationships as binary relationship type.

A second important characteristic of modeling molecular data is that sequences may be a bag (or multi-set) rather than a set of relationship instances, since the same gene (or intergenic sequence) may appear multiple times within the same DNA sequence such as in gene homologs or tandem repeated DNA sequence blocks. We use the letter B in OB to denote that the relationship is an ordered bag that allows repetition. [1] Thirdly, we have to specify the direction of ordering, which applies to all entities of one type related to a single entity (out of many) of another type. The

[1] In the traditional ER model, repetition is not allowed for relationship instances.

solid dot at one end of the relationship in Figure 3.1(b) indicates that related entities on this side are ordered. [2]

3.1.2 Input/output Functional Process Concept

EXAMPLE 2. *Figure 3.2 shows the biological concept of a gene expression process and its EER conceptual modeling. A process such as transcription or translation in Figure 3.2(a) relates the data of genes, mRNA sequence or protein sequence in a directed way. The entities in a process can have three main participating roles. The roles are input (i in Figure 3.2(b)), output (o in Figure 3.2(b)), or catalyst (c in Figure 3.2(b)). We model transcription or translation as a process relationship.*

Molecular interaction is the key to the dynamics of biological processes such as gene expression, protein folding, metabolic pathways and cell signaling. In EXAMPLE 2, a protein is created from its gene through a series of interactions known as transcription and translation, shown in Figure 3.2(a). Some entities act as inputs, some as outputs (products) and others (typically enzymes) as catalysts to steer the process in a certain direction. These three roles in the system of a biochemical interaction are fundamental to molecular biology and important to any modeling scheme. It is also important to reflect hierarchy and subsets in such reactions since a complex process is made up of a sequence of unit processes resembling the workflows of assembly lines.

Pathway data has these kind of attributes. A pathway is a linked set of biochemical reactions. The product of one reaction is a reactant of, or an enzyme that catalyzes, a subsequent reaction. Figure 3.2(b) is the EER schema for the gene expression process. In this modified EER model we can represent the dynamic be-

[2]We note that other data models also have ordered and multi-set relationships, such as the list and bag constructors in the ODMG object model.

havior of different agents. For example, mRNA is the output of the transcription process and an input of the translation process as well. Our extension of the EER model enables us to incorporate this important input/output process concept. In the notation shown in Figure 3.2(b), the p in the relationship indicates a process relationship, the edges marked i represent input entities to the process relationship. Edges marks o and c represent output entities and catalyst entities, respectively. The arrow directions also indicate the type of the role of each entity in the process relationship.

3.1.3 Molecular Spatial Structure Concept

EXAMPLE 3. *Figure 3.3 shows the 3D chemical structure of amino acid alanine and its EER conceptual modeling. Each amino acid (residue) is composed of various types of atoms. Some atoms form chemical bonds in 3D space between each other. We model atoms and residues as molecular spatial relationships.*

The function of a molecule is partly determined by its three dimensional spatial structure, for example the structure of DNA affects which regions can be read to make proteins, and the function of enzymes is often altered by minor influences on their structure due to changes in temperature, or salt concentration. These spatial structures are experimentally determined by X-ray crystallography or NMR [104] which generate topographical measurement data such as bond angles or distances as well as image data. As previously mentioned, a protein is a variable length chain composed from a mix of up to 20 different possible amino acids, or residues. Each residue is itself structurally composed of various types of atoms such as C, H, O, N, shown in Figure 3.3(a). Each atom can be treated as a point and its position is thus represented by x, y, z coordinates in space. How those atoms are positioned can affect their fundamental chemical interactions, because of charge repulsion, and

proximity needed for breaking and forming new chemical bonds between the atoms. This type of information is particularly important to biochemistry research, and is applied to pharmaceutical design since drug interactions are often adjusted at this level of structure and shape modifying function. Figure 3.3(b) is the EER schema for the 3D structure of residues. We use the letter *Sp* to represent the molecular spatial relationship between residues and atoms.

3.2 Formal Definitions for EER Model Extensions

We now give a formal definition for our extensions to the relationship concept in the ER/EER models. The main concepts in these models are entities and entity types, relationships and relationship types, attributes, and class/subclass inheritance [40]. Entities represent objects and relationships represent interactions or associations among objects. Attributes describe properties of entities or relationships.

A relation type R among n entity types $E_1, E_2, \ldots, E_n$ defines a set of associations among the participating entities. Mathematically, R is a set of relationship instances r_i, where each r_i associates n entities $(e_1, e_2, \ldots, e_n)$ and each entity e_j in r_i is a member of entity type $E_j, 1 \leq j \leq n$. A relationship type can be defined as a subset of the Cartesian product $E_1 \times E_2 \times \ldots \times E_n$. A basic binary relationship in the EER model is a set of the relationship instances, where each pair (e_i, e_j) has the properties that $e_i \in E_1, e_j \in E_2$ and $R \subset E_1 \times E_2$. The next three subsections describe formally the proposed new relationships for enhancing the existing EER model to represent biological data.

3.2.1 Ordered Relationships

To model ordering, we must extend the relationship concept in two directions:1) Allow related entities to be ordered, and 2) allow repetitions of the relationship

instances. This means that the relationship set must be extended to allow duplicates. Before giving formal definitions of the extensions to the relationship concept, we show how we propose to extend the revised diagrammatic notation in order to minimize the changes to the ER/EER. We propose 4 types of relationships:

- The original ER model relationship, which is an unordered set of relationship instances.
- An ordered set relationship, where each relationship instance is unique (no duplicate instances are allowed).
- An unordered bag relationship, which allows duplicate relationship instances.
- An ordered bag relationship, which allows duplicates with ordering, and can be used to model the situations discussed earlier.

The notation for these 4 relationships is shown in Figure 3.4. The letters O, B, and OB stand for Ordered, Bag (or multiset), and Ordered Bag, respectively. An edge with the filled circle (or solid dot) indicates that the attached entity type is the one whose elements are ordered by the relationship instances that are related to a specific entity from the other entity type.

We now formalize these four types of relationships. We first define the concepts of *unordered set*, *ordered set*, *unordered bag* and *ordered bag*, and then give the formal definitions of the relationships.

Let E be a set.

Let $\underbrace{E \times E \times \ldots \times E}_{n}$ be the set of all ordered n-tuples $(e_1, e_2, \ldots, e_n)$ where e_1, e_2, $\ldots$, $e_n \in E$.

Let $\underbrace{E \otimes E \otimes \ldots \otimes E}_{n}$ be the set of all unordered n-tuples $[e_1, e_2, \ldots, e_n]$ where e_1, e_2, $\ldots$, $e_n \in E$. For convenience, we denote $\underbrace{E \times E \times \ldots \times E}_{n}$ by E^n and $\underbrace{E \otimes E \otimes \ldots \otimes E}_{n}$ by $\otimes E^n$.

Definition 1. **Unordered set**

We say that F is an unordered set on a set E if [3]

$$F \subseteq \bigcup_{n \in N} F_n$$

where

$$F_n = \{[e_1, e_2, \ldots, e_n] \in \otimes E^n | e_i \neq e_j, \forall i \neq j\}$$

Definition 2. **Ordered set**

We say that F is an ordered set on a set E if

$$F \subseteq \bigcup_{n \in N} F_n$$

where

$$F_n = \{(e_1, e_2, \ldots, e_n) \in E^n | e_i \neq e_j, \forall i \neq j\}$$

Definition 3. **Unordered bag**

We say that F is an unordered bag on a set E if

$$F \subseteq \bigcup_{n \in N} \otimes E^n$$

Definition 4. **Ordered bag**

We say that F is an ordered bag on a set E if

$$F \subseteq \bigcup_{n \in N} E^n$$

Definition 5. **Unordered set relationship**

We say that R is an unordered set relationship between E_1 and E_2 if $R \subseteq E_1 \times E_2$.

[3] N is the set of natural numbers.

Definition 6. **Ordered set relationship**

We say that R is an ordered set relationship between E_1 and E_2 if $R \subseteq E_1 \times E_2$ and for each $e \in E_1$, $\{e_j|(e, e_j) \in R\}$ is an ordered set on E_2.

Definition 7. **Unordered bag relationship**

We say that R is an unordered bag relationship between E_1 and E_2 if R is a multiset of (e_i, e_j) elements, and for each $e \in E_1$, $\{e_j|(e, e_j) \in R\}$ is an unordered bag on E_2.

Definition 8. **Ordered bag relationship**

We say that R is an ordered bag relationship between E_1 and E_2 if R is a multiset of (e_i, e_j) elements, and for each $e \in E_1$, $\{e_j|(e, e_j) \in R\}$ is an ordered bag on E_2.

When ordered relationship of definitions 6 and 8 are represented diagramatically, the dot is on the side of E_2. For example, suppose that a DNA sequence entity with identifier X is:

$$\ldots \underbrace{A \ldots A}_{gene} \underbrace{C \ldots C}_{nongene} \underbrace{A \ldots A}_{gene} \underbrace{G \ldots G}_{nongene} \underbrace{T \ldots T}_{gene} \ldots$$

Suppose that A ... A, A ... A, T ... T are genes in the sequence, whereas C ... C, and G ... G are non-gene sequences. Then, the relationship instances including genes in sequence X will be the ordered list:

$$(\ldots, (X, A \ldots A), (X, A \ldots A), (X, T \ldots T), \ldots)$$

3.2.2 Process Relationships

There are 3 basic roles in a process relationship:

- Input(s): entities consumed by the process, for example by being transformed to some other entities.
- Output(s): entities produced by the process.

- Catalyst(s): entities that are needed for the process to work.

In Figure 3.5(a), E_1 represents the input entity, E_2 represents the output entity, and E_3 represents the catalyst entity. Symbol i stands for input, o stands for output and c stands for catalyst. We use e_1 to represent entities in E_1; e_2 to represent entities in E_2 and e_3 to represent entities in E_3.

Definition 9. **Process relationship (basic type)**

A basic process relationship is defined as a set of relationship instances (e_1, e_2, e_3), where $e_1 \in E_1$ represents the input entity, $e_2 \in E_2$ represents the output entity, and $e_3 \in E_3$ represents the catalyst entity. The relationship instance can also be represented as:

$$\{e_1 \overrightarrow{\{e_3\}} e_2\}$$

where catalyst is optional and the input, output, and catalyst entity types do not have to be distinct.

Definition 10. **Process relationship (general type)**

In general, a process can have multiple inputs, outputs, and catalysts. In Figure 3.5(b), $E_{i1} \ldots E_{ij}$ represent the input entities. $E_{o1} \ldots E_{ol}$ represent the output entities. $E_{c1} \ldots E_{ck}$ represent the catalyst entities. The process relationship is a set of relationship instances:

$$(e_{i1}, \ldots, e_{ij}, e_{o1}, \ldots, e_{ol}, e_{c1}, \ldots, e_{ck})$$

where

$$e_{im} \in E_{im}(1 \leq m \leq j), e_{om} \in E_{om}(1 \leq m \leq l),$$

$$e_{cm} \in E_{cm}(1 \leq m \leq k)$$

3.2.3 Molecular Spatial Relationships

Definition 11. **Molecular spatial relationship**

A molecular spatial relationship is defined to describe the spatial relationships among a set of atoms in 3D space. Let A be a set of atoms, and let M be a set of molecules, as shown in Figure 3.6. The molecular spatial relationship instance is a 3-tuple:

$$\langle(\langle a_1, x_1, y_1, z_1\rangle, \langle a_2, x_2, y_2, z_2\rangle, \ldots,$$

$$\langle a_{n_i}, x_{n_i}, y_{n_i}, z_{n_i}\rangle), formula, m_i\rangle$$

where $(a_1, a_2, \ldots, a_{n_i})$ is a group of associated atoms forming a molecule, $formula$ denotes the formula of the molecule and $m_i \in M$. The $x_{n_i}, y_{n_i}, z_{n_i}$ associated with each atom a_{n_i} describe the 3-dimensional atom location in the spatial structure. The molecular spatial relationship bares some characteristics of aggregation (reverse is **Part-Of**) relationship in which atom entities are part-of a molecule entity. But it has its own property that these atom entities are connected by some forces (bonding) that need explicit modeling.

3.3 Summary of New EER Notations

Table 3.1 summarizes the notations of the proposed new relationships. Notice that we have added considerable modeling and representation power to the basic relationship concept in ER models. However, the notation to display all these new complex concept is not overly complex and hence should be easy to utilize.

3.4 Applications on Molecular Biology System

In this section, we will give the details of the EER conceptual schema of our molecular biological system that utilize the new EER constructs for the new types of

Table 3.1. New EER relationships and their usage

EER relationship	Comments
Unordered set	General relationship, unique instance.
Ordered set	Associates entities with ordering features. The relationship instances are unique.
Unordered bag	Associates entities without ordering features. The relationship instances can be duplicated.
Ordered bag	Associates entities with ordering features. The relationship instances can be duplicated.
Process	Associates different entities by the roles they have in a process. The roles are input, output, and catalyst.
Molecular spatial	Associates atom entities with molecule entities in 3D space.

relationships we defined in above sections. The conceptual schema is roughly divided into several parts: the DNA/gene sequence, the protein structure, the molecular pathway, and the aging process.

3.4.1 The DNA/Gene Model

As we know, a DNA sequence is made up of 4 nucleotide base in a specific order. A gene is one segment of a DNA sequence with a specific function. Usually, DNA sequences come from different sources of organisms, which have well-established phylogenetic classification schema, such as common name, genus and species. Figure 3.7 shows the details of an EER conceptual schema for DNA/gene sequence that utilizes the order and bag (multiset) relationship. Note that we use binary relationship type to represent the order relationship between the DNA, gene, etc. Practically, we can have several options to model this relationship. Figure 3.8(a) shows their relations using binary relationship type. **Gene** and **Non-Gene** each has a m:n binary relation with **DNAseq**. Figure 3.8(b) shows the EER modeling option that DNA sequence, gene, and non-gene forms a ternary relation whenever a relationship instance (dna,

gene, non-gene) exists. Figure 3.8(c) shows a new entity type **Segment** that is created to include all entities of both **Gene** and **Non-Gene** and this **Segment** has an order relationship with **DNAseq**. In Figure 3.8(d) we have a general approach to represent the relationship between biological sequence entities. This model is easy to modify and extend depending on various situations. Because some instances in **DNAseq** are **Gene** type, some are **Non-Gene** type and some are other type, we could create a union of these types and thus make **DNAseq** to be subclass of it. A recursive ordered relationship **Has** exists between DNA sequences themselves. One (long) sequence participates in the super-sequence role and the other (shorter) sequence in the sub-sequence role. In section 3.5.1 we will discuss the different models in the ER-to-Relational mapping process.

3.4.2 The Protein 3D Structure Model

Usually, a structure-determined protein contains one or more chains of residues. These originally spatial-free linear chains are constrained by various physical or chemical forces to form higher levels of 3D structure. They are secondary, tertiary and quaternary structure of the protein.

- Primary structure is a linear polypeptide chain of residues with specific order.
- Secondary structure refers to the general three-dimensional form of local regions (segments of chains) or overall shape of polypeptide chain. Helix, sheet, and turn are characteristic structural components. An alpha-helix is a tight helix formed out of the polypeptide chain. The polypeptide main chain makes up the central structure, and the side chains extend out and away from the helix. The CO group of one amino acid (n) is hydrogen bonded to the NH group of the amino acid four residues away (n +4). In this way every CO and NH group of the backbone is hydrogen bonded. They are formed by hydrogen bonding.

Multiple hydrogen bonds make a segment of amino acid chains fold in specific ways.

- Tertiary structure is the full three-dimensional folded structure of the polypeptide chain. It assembles the different secondary structure elements in a particular arrangement. As helices and sheets are units of secondary structure, so the domain is the unit of tertiary structure. In multi-domain proteins, tertiary structure includes the arrangement of domains relative to each other as well as that of the chain within each domain [63].
- Quaternary structure is a protein complex assembled by multiple-subunit proteins. Examples of proteins with quaternary structure include hemoglobin, DNA polymerase, and ion channels. Quartenary structures are stabilized mainly by non-covalent interactions; all types of non-covalent interactions: hydrogen bonding, van der Walls interactions and ionic bonding, are involved in the interactions between subunits. In rare instances, disulfide bonds between cysteine residues in different polypeptide chains are involved in stabilizing this level of structure.

Figure 3.9 shows the EER conceptual schema of protein 3D structure that utilize the new types of relationships. We go through these entities and relationship from the bottom to the top level.

Atom. This entity type represents the chemical atoms such as C, H, O, N and S in the molecular structure. They can be identified uniquely by their atom serial number and spatial position. Cartesian coordinates (x,y,z) is one of such coordinate models.

SSBond and **HBond**. These are typical examples of molecular spatial relationship types denoting the chemical bonding formed among atoms. It (spatial bond

relationship) can be identified uniquely by its bond type, bond length and atoms that participate in.

Residue. This entity type represents the amino acids connecting to each other in the chains of protein primary sequence. Each residue is a molecule (exists dependently) composed of atoms via **Molecule-Structure** spatial relationship.

Molecule-Structure. This type of molecular spatial relationship is defined to describe the spatial relationship between a set of atoms within a molecule.

Made-of. This is the ordered bag relationship type. It denotes that a sequence of residues (some residues can be duplicated) forms a specific chain of protein primary sequence. The solid dot at one end of the relationship indicates that related entities of **Residue** are ordered with respect to a single entity (out of many) of **Chain**. Thus there exists inherent attributes of **Made-of** ordered relationship, such as the length of chain in terms of residue count and the order number of each residue in this chain.

Chain. This entity type models the one-dimensional structure of protein sequence. It is a line of residues without constraint. A single chain can be partitioned into one or more segments. These segments make up the central structure of secondary components. They can form α-helix, β-pleated sheet, or turn.

Helix. This entity type models one type of the secondary structural component, α-helix. It is formed by every five consecutive residues via the **Helix-Structure** spatial relationship. Its cardinality constrain is 1:5 between entity type **Helix** and **Residue**.

Sheet. This entity type models another type of the secondary structural component, α-pleated sheet. It is two-dimensional structure. It can be modeled as a sequence of 1D structure of **Strand** via the **Made-of** ordered relationship. There are several types of sheets, such as circle, bi-fork, etc. One instance of strand can be

shared by several different sheets. So the cardinality constraint between **Strand** and **Sheet** is m:n. Strand is also one segment of a polypeptide chain, which is formed by consecutive residues via the **Strand-Structure** spatial relationship without cardinality constraints.

Turn. This entity type models one of the secondary structural component, turn. There are three type of turns: 3-turn, 4-turn and 5-turn [75].

Assemble. This molecular spatial relationship denotes that two or more protein components can be assembled into a protein complex, thus forming dimers, trimers, tetramers, and etc. The *Type* attribute of the relationship denotes that the type of assembly whether it is composed of the same type of protein units (homo-multimer) or different types (hetero-multimer).

Motif/Domain. Usually, a motif consists of a small number of secondary elements (helices, sheets and turn), combined in local specific geometric arrangements. These motifs then coalesce to form domains. To simplify modeling, we do not distinguish between motifs and domains. Note that **Motif/Domain-Structure** spatial relationship relates the entity **Motif/Domain** and **SecondaryStructure**. Many proteins can share the same type of domains, so the cardinality ratio between **Protein** and **Motif/Domain** is m:n.

3.4.3 The Molecular Interaction and Pathway Model

Figure 3.10 shows the EER conceptual schema of the molecular interaction and biological pathway. In our conceptual model, the entity **Bioentity** is the high level class of biological objects that are physical entities with attributes pertaining to their internal properties (e.g. the nucleotide base sequence of a gene, the molecular weight of a protein). So it is the union of all types of biological entities, such as genes, proteins, cofactors (metal ions or small organic molecules), etc. Another important

entity is the **Interaction** that relates any pair of biological entities. These interactions include gene-gene and protein-protein interactions. Some complex formed by molecular interactions like DNA-protein binding can also be an instance of interactions. There exists three relationships between **Bioentity** and **Interaction** in our design. The **Input** and **Output** relationships are for any two pairs of interacting entities, and the **Catalyst** relationship is for other helping entities if they exists in the interaction. Here we name these three relationships using "input", "output" and "catalyst" for the purpose of process relationship mapping (discussed in section 3.5.2). Note that this design can also model the reaction concept with reactant (input), product (output) and catalyst roles. By definition, a pathway is a linked set of biochemical interactions (reactions). If we ignore the branch case of the pathway, it can be treated as a sequence of unique interactions. So we use the **Participate** ordered set relationship to denote the relation between **Interaction** and **Pathway**.

3.5 EER-to-Relational Mappings

In this section, we describe the implementation of the above new EER constructs to relational database. We show how to map the ordered relationship, the process relationship and the molecular spatial relationship to relational models. This allows us to implement a conceptual design on a relational database system, such as ORACLE^{TM}or MySQL^{TM}.

3.5.1 Ordered Relationship Mapping

In section 3.2.1 we defined four types of the ordered relationships shown in Figure 3.4. The mapping of **unordered set** relationship is a standard procedure [40]. For the **ordered set** relationship mapping, we create a new relation **R**, including the primary keys of **E1** and **E2** as foreign keys in **R** and rename them as *E1Id*

Table 3.2. Mapping ordered set relationship

R	E1Id	E2Id	OrderNo
	v1	v2	1
	v1	v3	4
	v1	v5	2
	v1	v7	3
	...		

Table 3.3. Mapping unordered bag relationship

R	E1Id	E2Id	BagDiscriminator
	v1	v2	1
	v1	v5	1
	v1	v2	2
	v1	v3	1
	v1	v2	3
	...		

and *E2Id*, respectively. The primary key of this relation is the combination of the attributes *E1Id* and *E2Id*. We also include additional *OrderNo* attribute to indicate the ordering of *E2Ids* related to the same *E1Id* value. The following constraint will hold on the *OrderNo* attribute: for all tuples with the same value for *E1Id*, the values of *OrderNo* will be distinct and numbered 1, 2, 3, ... (see Table 3.2).

For the **unordered bag** relationship mapping, the relation **R** includes the primary key of **E1** as *E1Id*, the primary key of **E2** as *E2Id*, and attribute *BagDiscriminator*. The *BagDiscriminator* is to discriminate the tuples if the value of (*E1Id*,*E2Id*) are the same in the bag relationship, because the elements in the bag can be duplicate. The primary key of this relation is the combination of the foreign key attributes *E1Id*, *E2Id*, and *BagDiscriminator*. The following constraint will hold on the *BagDiscriminator* attribute: for all tuples with the same (*E1Id*,*E2Id*) combination of values,

Table 3.4. Mapping ordered bag relationship

R	E1Id	E2Id	OrderNo
	v1	v2	1
	v1	v2	8
	v1	v2	2
	v1	v3	7
	v1	v3	4
	v1	v4	3
	v1	v4	5
	v1	v4	6
	...		

the values of *BagDiscriminator* will be distinct (they can be ordered 1, 2, 3, ...). Table 3.3 shows one example of the mapping in relation table.

For the **ordered bag** relationship mapping, the relation **R** includes the primary key of **E1**, the primary key of **E2**, and attribute *OrderNo*. Like the attributes of the above relations, the *OrderNo* is to both discriminate and order the tuples with the same *E1Id* value. The same constraint on *OrderNo* for ordered set applies here. The primary key of this relation is (*E1Id*, *E2Id*, *OrderNo*). Table 3.4 shows one example of the mapping in relation table.

3.5.2 Process Relationship Mapping

As defined in section 3.2.2 (Figure 3.5), the entities associated with the **process relationship** have three distinct types: *i* (input), *o* (output) and *c* (catalyst). For each process relationship **R**, we can have a new relation **R** with three attributes (i, o, c) whose values are the primary keys of each participating entities as shown in Table 3.5. Each such tables holds the relationship instances for one of the process relationships. Another relation called **ProcessRelationDesc** is needed to describe the participating entities for all process relationships. Its attributes include *Relation*,

Table 3.5. Mapping process relationship (basic)

R1	Input	Output	Catalyst
	v1	v2	v3
	v4	v8	v6
	...		

R2	Input	Output	Catalyst
	v7	v2	v13
	v6	v9	v5
	...		

Entity and *Role*. *Relation* records the names of the process relationship. *Entity* records the names of the entity type that participate in a process relationship while *Role* specifies their acting roles. Table 3.6 shows an example of the mapping results in relation table.

The above mapping works for process relationship that have one input, one output, and one catalyst only. If we want to map the general case, where there can be multiple inputs, outputs, or catalysts, we can name the input attributes $i1$, $i2$, ..., the outputs $o1$, $o2$, ..., and the catalysts $c1$, $c2$, ... An example is given in Table 3.7.

3.5.3 Molecular Spatial Relationship Mapping

As defined in section 3.2.3 (EER notation shown in Figure 3.6), the molecular spatial relationship **R** associates a group of atoms (component objects) spatially with a molecule entity (a composite object) with specific connectivity among atoms. For the mapping, we can have a new relation called **MolStructure** with attributes (*MoleculeId*, *Atom*, *Discriminator*, *X*, *Y*, *Z*, *AtomOId*). *MoleculeId* refers to the primary key of **Molecule** relation in Table 3.8. As described in the ordered relationship

Table 3.6. Mapping process relationship (process description)

ProcessRelationDesc	Relation	Entity	Role
	R1	E1	i
	R1	E2	o
	R1	E3	c
	R2	E4	i
	R2	E5	o
	R2	E6	c
	R3	E7	i1
	R3	E8	i2
	R3	E9	o1
	R3	E10	o2
	R3	E11	o3
	R3	E12	c1
	R3	E13	c2
	...		

Table 3.7. Mapping process relationship (general)

R3	i1	i2	o1	o2	o3	c1	c2
	v1	v1	v3	v4	v2	v2	v7
	v2	v8	v6	v6	v8	v9	v12
	...						

mapping, the attribute *Discriminator* distinguish the atoms of the same type in a molecule. *X*, *Y* and *Z* attributes are the cartesian coordinates of atoms. *AtomOId* is a system-generated unique object id for each instance in this relation. The primary key of this relation is *AtomOId*. The alternative keys can be (*MoleculeId*, *X*, *Y*, *Z*) or (*MoleculeId*, *Atom*, *Discriminator*). Table 3.9 shows the mapping example of water and alanine molecules. For the simplicity, H atoms are deliberately omitted.

To record the bond information, we should have associated connection relation called **Bond** with attributes (*AtomOId1*, *AtomOId2*). *AtomOId1* and *AtomOId2* refers to the primary key *AtomOId* of relation **MolStructure** in Table 3.9. We can

Table 3.8. Mapping molecular spatial relationship (molecule)

Molecule	MoleculeId	Name	Formula	Isomer
	1	water	H2O	0
	2	alanine	C3H7O2N	21
	...			

Table 3.9. Mapping molecular spatial relationship (structure)

MolStructure	MoleculeId	Atom	Discriminator	X	Y	Z	AtomOId
	1	H	1	-1.4	1.3	0.0	1
	1	H	2	1.4	1.3	0.0	2
	1	O	1	0.0	0.0	0.0	3
	2	N	1	1.6	1.5	0.0	4
	2	C	1	0.0	0.6	0.0	5
	2	C	2	-1.3	1.4	0.0	6
	2	C	3	0.0	-0.6	0.0	7
	2	O	1	1.7	-1.6	0.0	8
	2	O	2	-1.7	-1.6	0.0	9
	...						

enforce a constraint that the value of *AtomOId1* is always less than the value of *AtomOId2* because the connectivity between atoms (nodes) is un-directional.

3.6 Summary

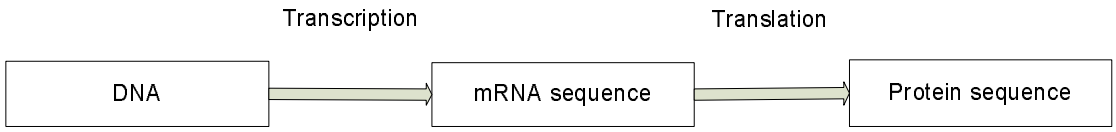
In this chapter, we introduced three new types of relationships into the EER model: ordered relationship, process relationship and molecular spatial relationship. We also extended the relationships to allow bags (or multi-sets) of relationship instances, since many relationships in molecular biology fall into this category. We illustrated the need for these relationships in modeling biological data and we proposed some special diagrammatic notation. By introducing these extensions we anticipate that biological data having these properties will be made explicit to the

Table 3.10. Mapping of molecular spatial relationship (connection)

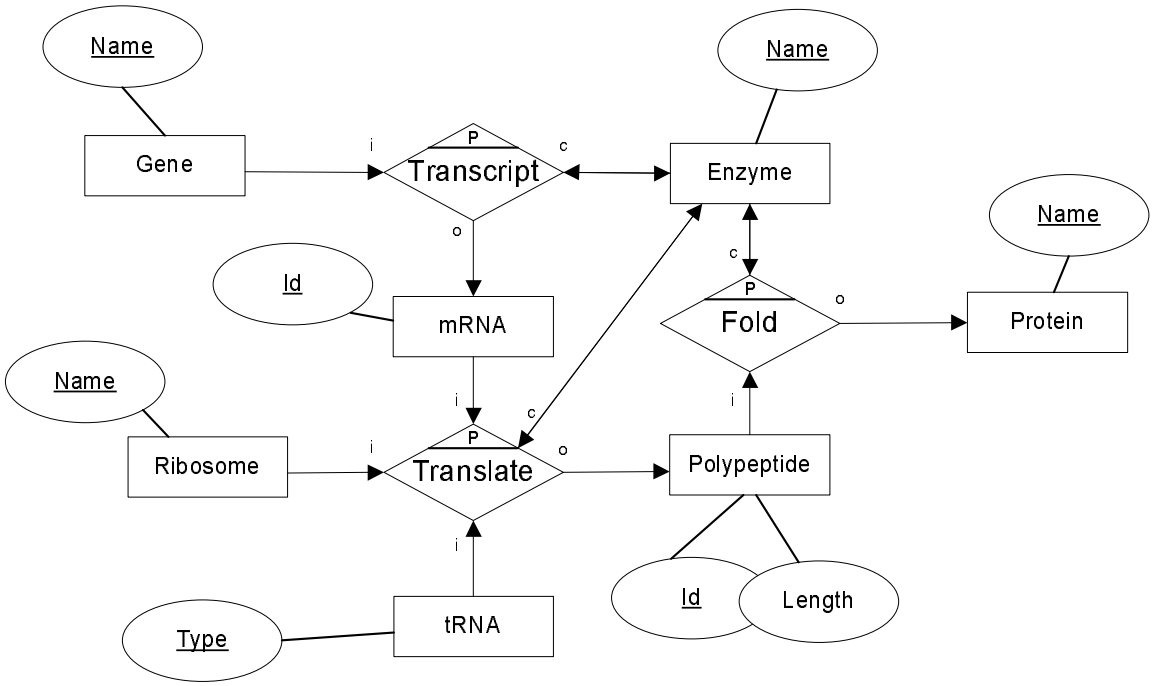
Bond	AtomOId1	AtomOId2
	1	2
	2	3
	4	5
	5	6
	5	7
	7	8
	7	9
	...	

data modeler, which would help direct future biological database implementation. In particular, our unordered bag relationship can be used for various reaction data, and our ordered bag should be useful for sequence and genetic features. Our process relationship would be useful for reaction, interaction and pathway data. These changes do not add much complexity to the existing EER model, thus making them easier for integration. We also give the formal definitions for these new concepts and summarized their notation and usage. We also showed how these additional concepts can be mapped into relations for implementation in relational databases such as ORACLE^{TM}or MySQL^{TM}.

In the next chapter, we will show how these relationships can be used in the construction of domain ontology of our mediator query system.



(a)



(b)

Figure 3.2. (a) Gene expression process (b) EER model of gene expression.

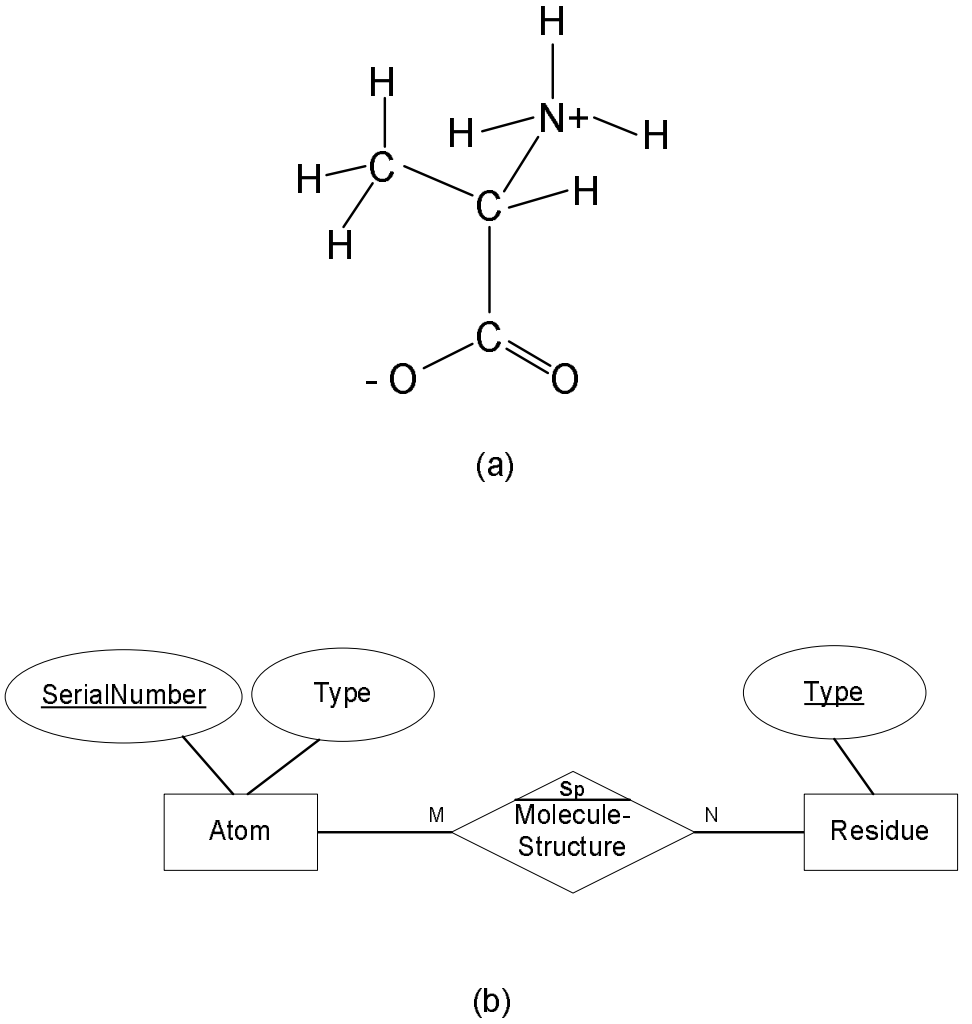


Figure 3.3. (a) 3D structure of alanine (b) EER model for atoms and residues.

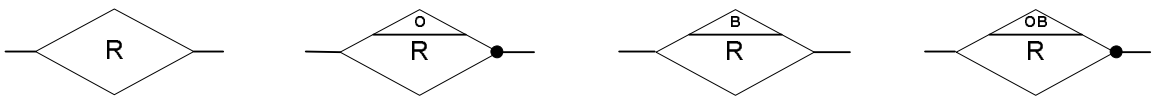
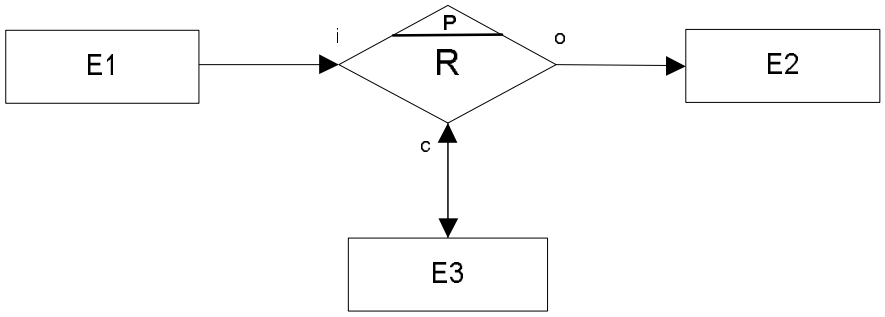
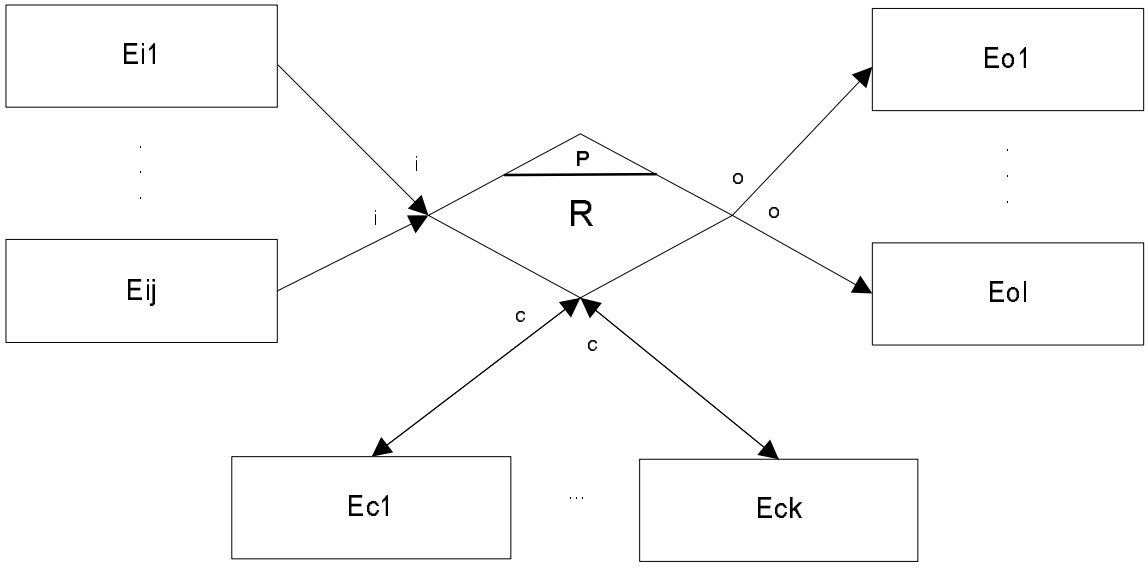


Figure 3.4. EER notation for unordered set, ordered set, unordered bag and ordered bag relationships (from left to right).



(a)



(b)

Figure 3.5. EER notation for process relationships (a) basic (b) general.

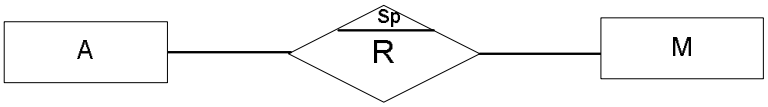


Figure 3.6. EER notation for molecular spatial relationship.

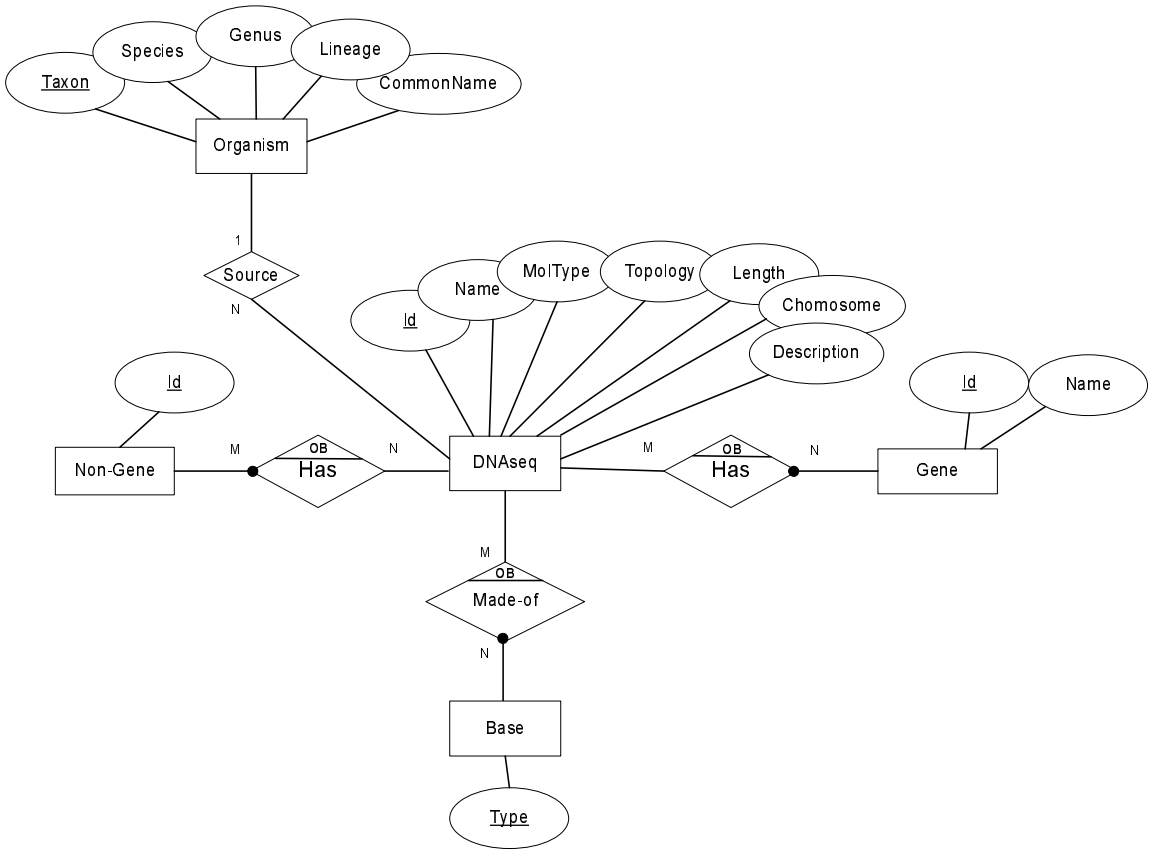


Figure 3.7. EER schema of DNA sequence.

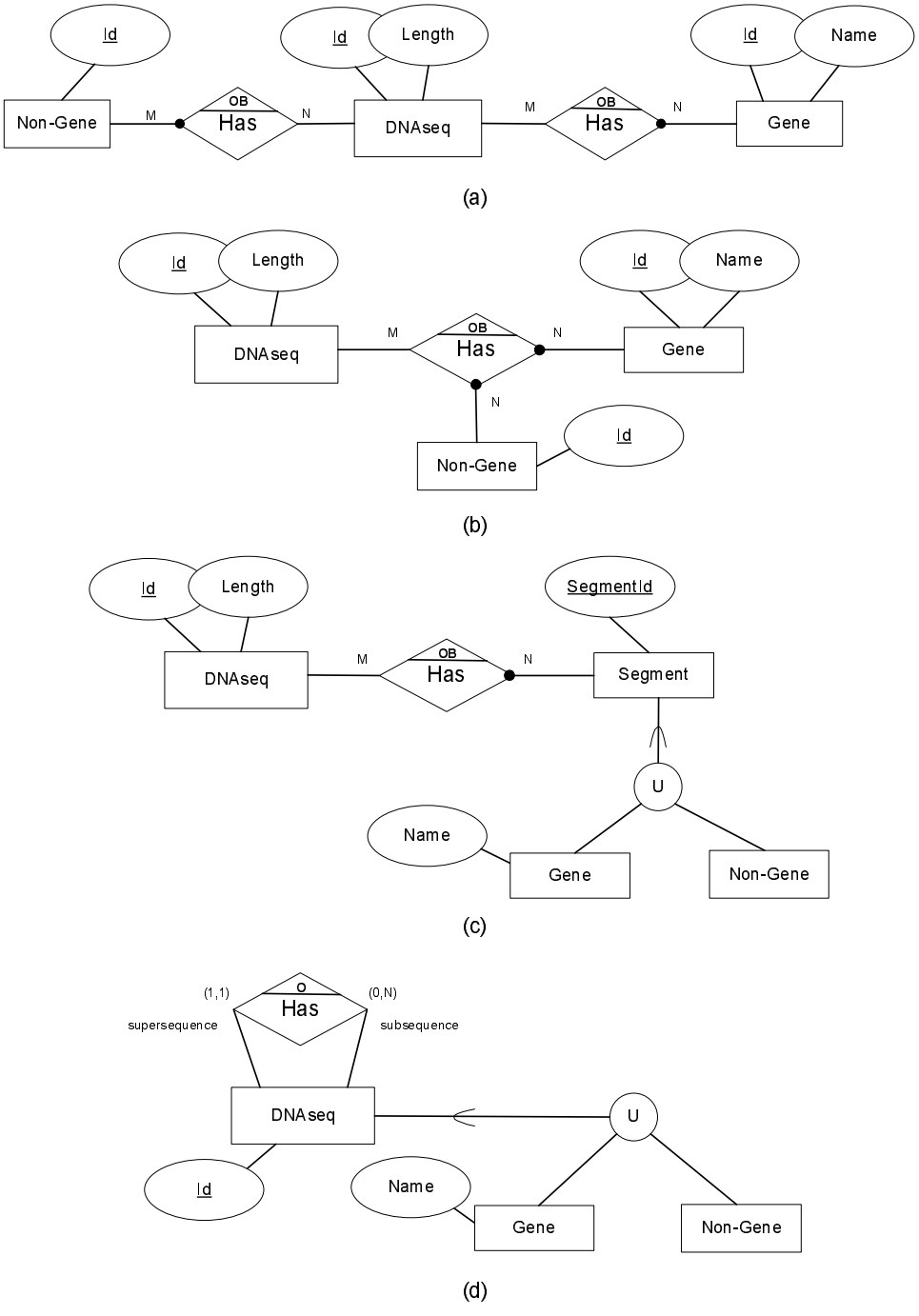


Figure 3.8. EER model options for DNA sequence (a) binary type (b) ternary type (c) union type (d) general type .

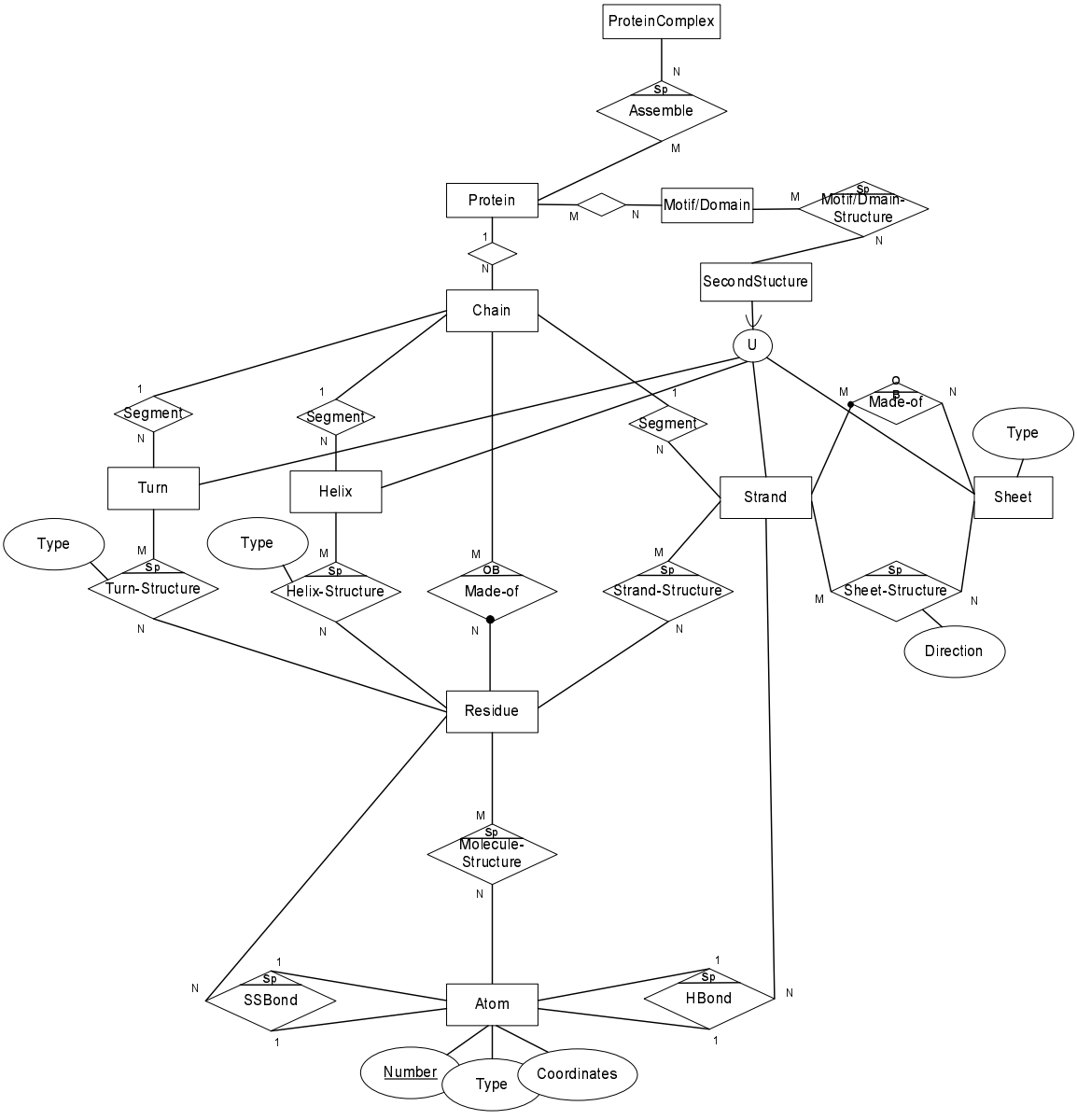


Figure 3.9. EER schema of the protein 3D structure .

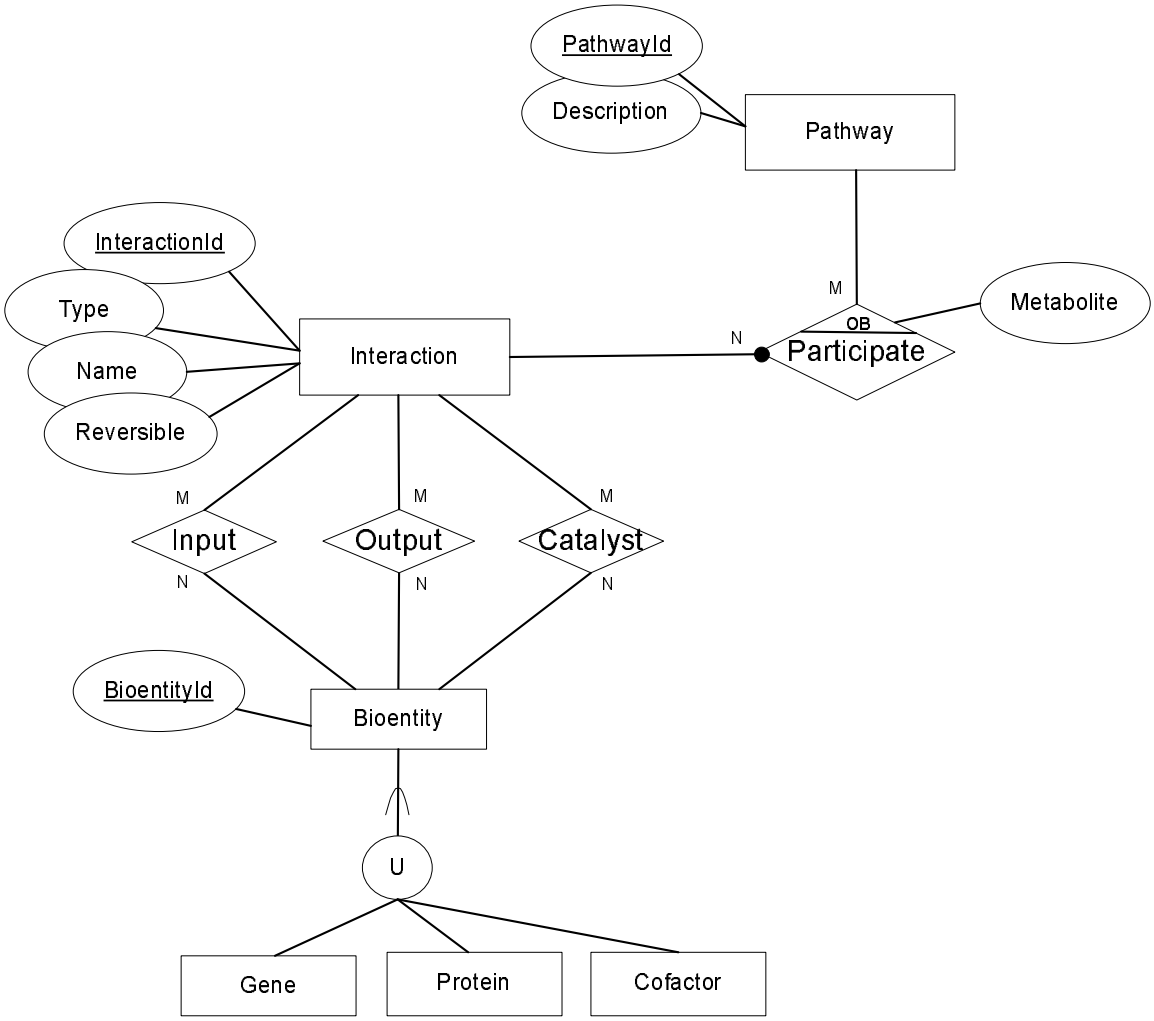


Figure 3.10. EER schema of the molecular interaction and the pathway .

CHAPTER 4

MEDIATED DOMAIN ONTOLOGY

In the previous chapters, we discussed the data modeling problems in bioinformatics data management. We advocated the enhancement of EER model to accommodate the bioinformatics data features. But these data are only raw data (produced by experiments), at the instance level, such as gene sequences or protein structures. Many other data, at the conceptual level, such as protein or gene annotations (produced by curators) are not easily modeled using traditional methods, such as the schema-based approach. They are captured and transmitted as domain knowledge, control vocabularies, and ontological concepts. In this chapter, we discuss how these domain concepts can be organized into a mediated domain ontology, i.e. a "global" data schema. Users can pose queries against this schema, thus support integrated access to multiple bioinformatics data sources. Section 4.1 introduces different types of concepts. Section 4.2 discusses the structure and characteristics of our mediated ontology. Section 4.3 gives its formal definitions. Section 4.4 shows how our mediated ontology can be used in querying. Section 4.5 summarizes this chapter.

4.1 Ontology Concepts

In this section, we discuss different types of concepts in mediated domain ontology, the data sources of these concepts, and how they are used for querying and integration purposes.

4.1.1 Entity Concepts and Instances

Generally speaking, there are 2 types of concepts. One type of concepts such as *Protein*, has many instances, uniprot:P07327 is one of them. The other is the annotation type, such as "alcohol dehydrogenase activity" (GO:0004024), which annotates this protein instance functionality. So, the ontological concepts (or vocabulary terms) can be roughly classified into 2 types. One is the basic concept, and the other is the annotation concept, or strictly speaking, the description of other concepts. The instances of the first type are stored in various databases as database entries. Basic concepts are shared among different ontologies.

All concepts in the ontology derive from the root concept. For the purpose of mapping between the ontology graph data model and Entity-Relationship (ER) data model [40], we classify concepts into three top classes: the entity-like, the relationship-like, and the attribute-like concepts.

An instance of the entity-like concept is an individual object defined by a set of attribute-like concepts. The instance of the attribute-like concept is a value. For example, *Molecule* is an entity concept defined by attribute-like concepts: *Name*, *CAS number* (Chemical Abstracts Service registry number), *Formula*, *Molecular weight*, etc. So one instance example is a 4-tuple: (water, cas:7732-18-5, H2O, 18). Every concept has at least one attribute: *Name*, which is a label to the concept (abstract concept entity may have synonyms). The entity-like concept includes cellular components, tissues, organs, etc. But these entity concepts do not have many instances. Currently, they are not part of our supported data set.

A relationship-like concept represents a binary relationship between any two concepts. It includes the interaction type concepts such as atom bonding, molecular binding and the process type concepts such as transcription and translation. Several

more specific classes can be derived from these 3 classes via *isa* relationships. They are: *Structure*, *Interaction*, *Role*, *Reaction*, *Pathway*, *Process*, *State*, and *Source*.

The structure-like concept includes various types of molecular structure concepts that are already existing or experiment verified such as 1D structure (protein and DNA sequence), 2D structure (turn, helix, sheet), 3D structure (protein domain, 2-layer lipid membrane structure, ribosome structure), etc. Please see chapter 3 for examples.

The interaction-like concepts includes various kinds of molecular interactions, such as the protein-protein, the DNA-protein, the gene-gene, etc. The *Role* concept includes these kinds of concepts: *Input*, *Output*, *Catalyst*, *Donor*, *Receptor*, *Inhibitor*, *Enhancer*, etc. It denotes different roles of the object involved in various kinds of interactions. The *Reaction* concept includes various kinds of biochemical reactions. The *Pathway* concept includes various kinds of biological pathways. The *Process* concept includes various kinds of biological processes. Because many instance data existed for *Structure*, *Interaction*, *Reaction* and *Pathway*, they are practically treated as entity-like concepts.

The *State* concept includes these kinds of concepts that describe the status or state of molecules in the interaction/reaction/process. These concepts include ionic/molecular, oxi/red, active/inactive states, etc. The state-like concept can also be extended to describe the various kinds of diseases.

The concept *Source* can describe either the biological sources of the protein sequence or the stored locations. We further classify it into *BioSource* and *DataSource*. *BioSource* includes organisms, organs, tissues, cells, clinic samples, etc. *DataSource* includes GenBank, UniProt, PDB, etc.

According to the classification of ontology concepts, the data instances from these sources can also be classified into different types: protein, enzyme, gene, struc-

ture, interaction, reaction, pathway, process, and disease (currently supported data set). Each type represents a different level of complexity of the instance object. So an instance object's potential properties include all the attributes of its class concept and the directly related concepts. Like the Objected-Oriented (OO) model, the child concept can inherit all the properties of its parent concept, and add extra properties by the specialization of the abstract concept.

Table 4.1 shows entity concepts and instance examples (database entries).

Table 4.1. Entity concepts and instances

Entity Concept	Representative Database	Entry accession
compound	NCBI PubChem [46]	CID:164010
protein	UniProt	P07327
gene	NCBI UniGene [46]	Hs.654433
mrna	EMBL Nucleotide	M12963
dna	GenBank	AY948115
structure	PDB	1HSO
reaction	Reactome	REACT-896.4
pathway	KEGG Pathway	ko00010
organism	NCBI Taxonomy	9606
disease	OMIM	103700
...		

4.1.2 Relationship Concepts

This section discusses various types of relationships proposed in biological and biomedical domain. These domain relationships can help promote the interoperability of ontologies and support the automation of text mining in biomedical literature and query formulation in mediator systems. Currently, our mediated ontology contains these kinds of ontological relationships: ChEBI relationships [31] and OBO relationships [98].

OBO relationships are proposed by Open Biomedical Ontologies consortium. The proposed 10 relationships are: *is-a*, *part-of*, *located-in*, *contained-in*, *adjacent-to*, *transformation-of*, *derives-from*, *preceded-by*, *has-participant*, and *has-agent*. In chapter 3, we proposed 3 extended EER relationships: *sequence-ordered*, *functional-process* and *molecular-spatial*. In the following, we give the comparisons between two data models.

The two models overlap in certain areas and differ in others. The *is-a* is a basic relationship and is already handled by class/subclass inheritance in EER modeling. The *part-of* is also handled by aggregation relationship, although not explicitly. OBO spatial relationships (connecting one entity to another in terms of relations between the spatial regions they occupy): *located-in* and *contained-in* can be derived from *molecular-spatial* relationships.

The *located-in* relates two continuants $c1$ and $c2$ such that the spatial regions they occupy $r1$ and $r2$ are related by *part-of* relation all the time. The *contained-in* relates two continuants such that the spatial regions they occupy are overlapping all the time. By definition, *molecular-spatial* relationship relates a set of atoms (components) to a molecule (composite object). Atoms are located in x, y, z coordinates within (contained in) a molecule. The molecular spatial relationship bares some characteristics of aggregation (reverse is *part-of*) relationship that these atom entities are connected by some forces (bonding) that need explicit modeling. So, these two OBO spatial relationships can be explicitly modeled by the molecular spatial relationship. But *adjacent-to* relation, which relates two disjoint continuants proximate to each other in space all the time, cannot be modeled by our model explicitly. An example is: intron *adjacent-to* exon. OBO temporal relations (connecting entities existing at different times): *transformation-of*, *derives-from*, and *precede-by* are also similar to *process* relationships. The *process* relationship relates various kinds of entities,

designated by 3 distinct roles: input, output and catalyst that they participate in during a biological process. The *transformation-of* relates continuant class $C1$ and $C2$ wherever an instance of the class $C1$ is to have existed at some earlier time as an instance of the distinct class $C2$. That means the same instance belongs to different classes at different time. For example, normal colon changes into carcinomatous colon. In ER modeling, the same entity (instance) cannot belong to different entity type (class). This is unlike *transformation-of* and *derives-from* relationships, which relate different entities in the time axis in both instance level and class level. As *derives-from* relates classes of continuants by time, *preceded-by* relates classes of processes by time. An example is: translation *preceded-by* transcription. This relationship can be modeled by *process* relationship if we treat processes (defined in OBO) as entities in ER, so $P1$ preceded-by $P2$ can be modeled such that $P1$ is the input entity of a *process* relationship while $P2$ is the output and process name is "preceded-by". The *has-participant* relates a process, a continuant, and a time where the continuant participates as a bearer in the process. Examples are: translation *has-participant* amino acid and cell division *has-participant* chromosome. The *has-agent* relates a process, a continuant, and a time where the continuant participates as a direct cause for the process. An example is: transcription *has-agent* RNA polymerase. This responsible continuant is thus an entity that acts as a catalyst in a *process* relationship.

From the above definitions of OBO relations, we can see that they all incorporate a temporal argument, while our model does not provide it explicitly. But *process* relationship already indicates this implicitly, that input entities occur before output entities in a process. Concerning relationships in extended ER model that cannot be modeled by the OBO relationships, *sequence-ordered* relationship cannot

be modeled. It relates two entity types $E1$ and $E2$, where instances of $E1$ are ordered and related to one instance of $E2$. Also this relationship set can have duplicates.

Chemical entities are often bound to proteins when the protein 3D structure is measured, and can participate in enzymic reactions. The protein interaction and the pathway databases all involve small molecules. The ChEBI (Chemical Entities of Biological Interest) ontology proposed 7 relationships: *is-conjugate-base-of*, *is-conjugate-acid-of*, *is-tautomer-of*, *is-enantiomer-of*, *has-functional-parent*, *has-parent-hydride*, and *is-substituent-group-from* [31]. These relationships can only be applied on the *Molecule* concept level, but not all types of molecules.

The above different domain relationship sets are non-overlap. These sematic relationships are very important domain knowledge. In addition to our defined relationships, they can be used by scientists to query sequence, interaction, structure, reaction, and pathway/process data sets. For example, *has-functional-parent* denotes the relationship between two molecular entities, one of which possesses one or more functional groups from which the other can be derived by functional modification. So, this can be used in querying protein modification data.

4.1.3 Attribute Concepts and Annotations

In order to accurately describe the properties of the raw instance data, such as the function of a gene or the detailed domain structure of a specific protein, the external classification/annotation concepts need to be integrated into our mediated ontology.

The annotation is descriptions of the raw bioinformatics data (sequences, structures, etc). In a typical sequence database entry, it includes biological functions, sequence features, or literature references. Annotations are in free text form, usually captured from literatures by domain experts. They are easy for human to read

and understand, but are difficult for automated computer processing. If we want to integrate different data instances (proteins or genes), annotations must be parsed to some basic forms, which can be used by scientists to make queries on the source data instances. Because different annotators may use different terms for the same concept, domain ontologies have been created to address this standardization problem. This also helps the integration of bioinformatics data sources. The Gene Ontology (GO) is such an ontology. It has being successfully used to annotate genes and gene products. It has 3 sub-ontologies: cellular components, molecular functions, and biological processes. Its concepts are organized into a DAG (Directed Acyclic Graph) structure. In this graph data model, a node represents an individual concept, and an edge denotes a relationship between the two concepts (nodes). GO only has 2 relationships: *isa* and *partof*. The ontology graph has partial-order, and multiple inheritance properties.

While the annotation describes the instance data, the classification has a different purpose. It categorizes the instance data into different groups. For example, NCBI Taxonomy classifies the organisms by their evolutional lineages. So, its structure is strictly a tree, which is a hierarchy of *isa* relationships (from the leaf to the root). Figure 4.1 shows the *Homo Sapiens* taxonomy [1, 31]. Annotations and classifications are both treated as our mediated ontology concepts. Their relationships with the instance data can be denoted by: NCBI-taxonomy *classify* organism, and GO-annotation *annotate* protein, for instance.

There exists several standard instance classification systems. For the protein structure data, CATH and SCOP (Structure Classification of Protein) [7] are both the standard classification systems. CATH is a hierarchical classification of protein domain structure, which cluster proteins at four major levels: Class(C), Architecture(A), Topology(T) and Homologous superfamily(H). Class describes the

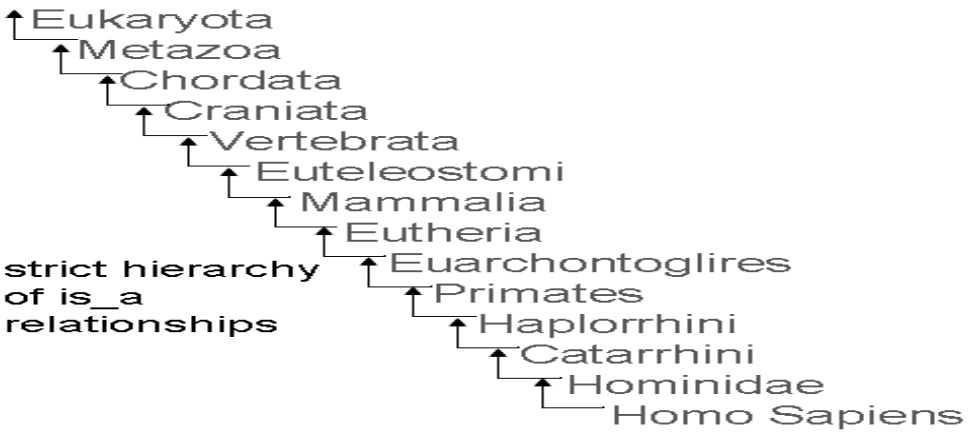


Figure 4.1. NCBI taxonomy, from [1] .

secondary structure. Architecture describes the gross orientation of secondary structure. Topology cluster structures into fold groups according to their topological connections and numbers of secondary structures. The homologous superfamilies cluster proteins with highly similar structures and functions [23]. Examples of CATH values are "mainly alpha", "orthogonal bundle", "helicase, ruva protein; domain 3", and so on. Currently, we only choose CATH data. Later, we will incorporate SCOP data.

The enzyme is a major class of protein data. It acts as a catalyst role in the biochemical reactions. The only authentic classification of enzymatic reactions is ENZYME [41]. Enzyme Commission (EC) number is assigned to each type of characterized enzyme. Examples of EC values are "oxidoreductases", "receptor protein-tyrosine kinase", "hydrolases", and so on.

For nucleotide sequence data, we choose SO (Sequence Ontology) as a standard. The SO provides a set of terms and relationships to describe the biological sequence features. It includes both raw features, such as nucleotide similarity hits, and interpretations such as gene models [34].

For the pathway data, we choose BioPAX [9] and KEGG database [50] ontology. KEGG has its own classification system called KO (KEGG Orthology). It organizes

the pathways in 4 levels. Examples of KO values are "metabolism", "carbohydrate metabolism", "glycolysis/gluconeogenesis", and so on.

Our low level (classification) concepts are directly imported from several authentic annotation sources: ENZYME, CATH, PDB and GO. These data sets only include protein/gene/pathway database entry ids and their annotation/classification system ids and values. For example, the Swiss-Prot protein id:P08833 (Insulin-like growth factor-binding protein 1), its PDB structure id:1ZT3 (C-terminal domain of Insulin-like Growth Factor Binding Protein-1 isolated from human amniotic fluid), and id:1ZT5 (C-terminal domain of Insulin-like Growth Factor Binding Protein-1 isolated from human amniotic fluid complexed with Iron(II)), its GO annotation id:0005615 (cellular component, extracellular space), etc. Our system does not store other information such as protein sequences and 3D structure features. These extra data will dynamically be fetched from the original data sources through our web service retriever based on the query specification of the user.

4.2 Ontology Structure

The Mediated Domain Ontology (MDO) is a conceptual data model for our mediator system. The term "mediated" means that it incorporates both the stable concepts (entity-like and relation-like concepts), but also a dynamically evolving control vocabularies of diverse classification/annotation systems on sub-domains. The high level (core) concepts are independent of external data sources. The low level concepts (annotations and classifications) will reflect the updated protein or gene annotations. MDO has a DAG structure. The core concepts are backbones; the low level concepts are extended from core concepts through *annotate* and *classify* relationships.

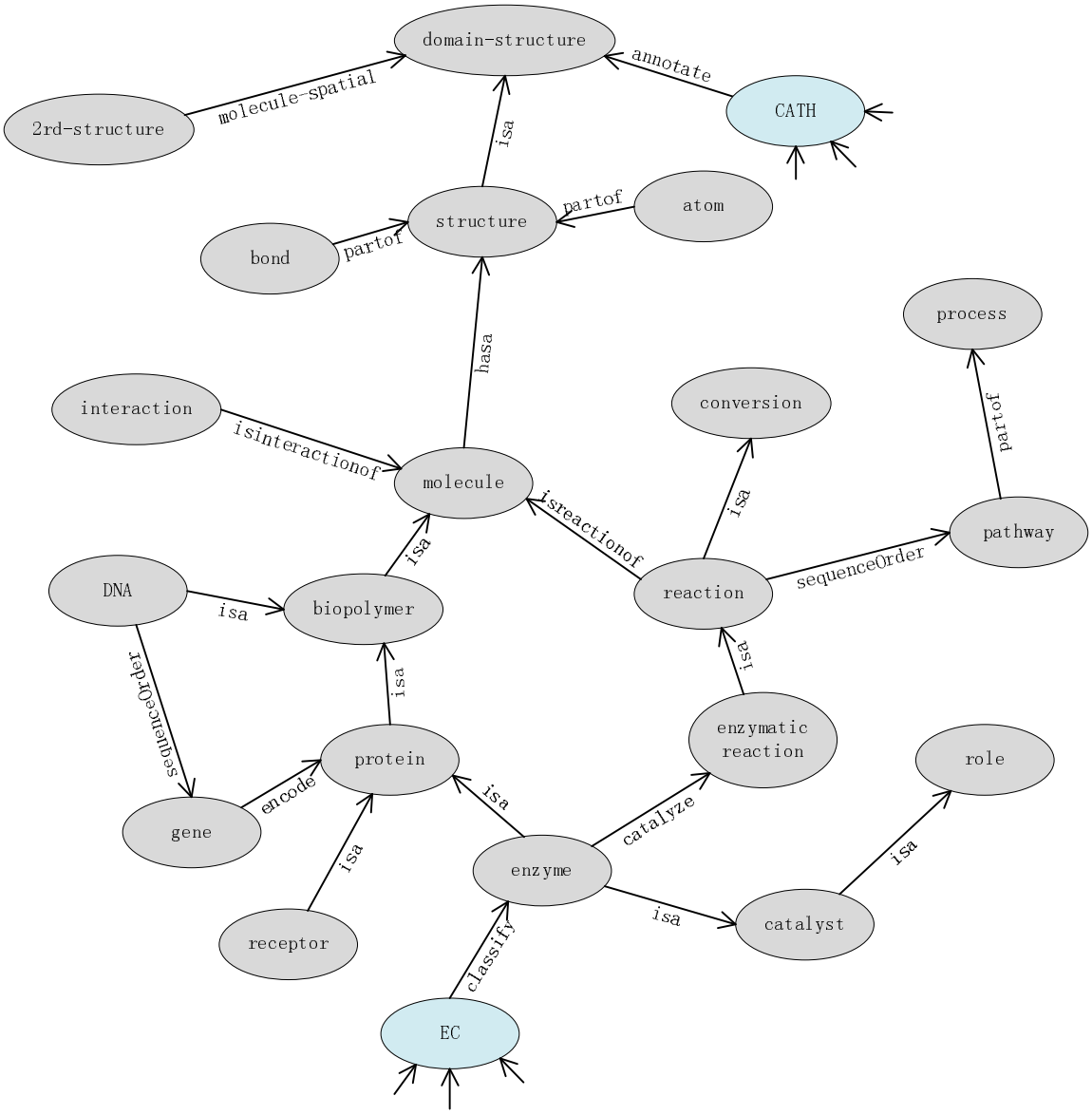


Figure 4.2. High level concepts in mediated ontology.

Figure 4.2 shows the structure of the core concepts in MDO. The *Entity*, *Relationship*, and *Attribute* concepts are not shown out. *Molecule* is the core of core concept sets. It *hasa Structure* (Molecular structure), and can produce other concepts such as *Reaction* and *Pathway* through molecular interactions. *Molecule* can

be specialized into *DNA* and *Protein*. *Protein* can be specialized into *Receptor* and *Enzyme*. *Enzyme* has one more relationship: *catalyze* with *Enzymatic reaction* compared to its parent concept *Protein*.

4.3 Formal Definitions

We now give a formal definition for our mediated domain ontology. Let C be the set of domain concepts. The concepts can be classified into 3 types:

- Entity-like concept EC, whose instances are tuples, and are available in the data sources.
- Attribute-like concept AC, whose instances are values.
- Relation-like concept RC, which denotes the binary relationships between 2 concepts.

Definition 1. A attribute-like concept AC is used to describe the internal or static characteristics of an entity. Its instance is a value.
EXAMPLE: *Name*, *Length*, and *Type* are typical attributes.

Definition 2. A entity-like concept EC which abstractly denotes a class of entities, can be designated by a concept name (or label) and defined by a set of attribute-like concepts AC. Thus, a entity-like concept EC is defined as $EC = \langle Name, otherACs \rangle$.
EXAMPLE: *Molecule* is defined as: $\langle Name, CAS, Formula, MolecularWeight \rangle$.

Definition 3. A relation-like concept RC is used to describe the external or dynamic characteristics of an entity. It denotes a binary relationship between 2 concepts. Its relationship instance is a triplet.
EXAMPLE: *Encode* is a relationship concept. *Gene encode Protein* is one of the relationship instances.

Definition 4. The structure of domain model is an ontology $O = \langle C, R \rangle$, where $C = \{c_1, c_2, \ldots, c_n\}$ is a set of concepts, and $R = \{r_1, r_2, \ldots, r_m\}$ is a set of relationship concepts RC between any pair of concepts c_i and c_j.

Definition 5. A concept can be interpreted (supported) by the existence of one or more data instances d. The interpretation of a concept c is defined as $I(c)$. Depending on the concept type, this interpretation function I will return different sets of data instances (the set of all instances of the supporting sources). For the EC, it will be evaluated to a set of data instances $D = \{d_1, d_2, \ldots, d_k\}$. For the RC, it will be evaluated to a set of data instance pairs. Some highly abstract concepts in the ontology cannot directly create instances.

There are two types of RCs, named intra-relationship and inter-relationship. Intra-relationship is between the same concepts; inter-relationship is between 2 different concepts. There are 3 basic RCs: *isa*, *partof*, and *attributeof*, which denote the relationship between part and whole, the relationship of concept type hierarchy, and the relationship of attributes. The others are: *definedby*, *attributeof*, *typeof*, and *isabout*. We also define 3 basic domain RCs: *order*, *process*, and *molecule-spatial*, which denote 3 common concepts in life science [39]. ACs include many types of standard annotations, such as *GO-annotation*, *SO-annotation*, *CATH-annotation*, etc.

4.4 Customized View of Domain Concepts

Since each domain ontology has its own focus, the same concept can be expressed in different degrees of abstraction. For example, the concept *Structure* is the leading role in the CATH ontology, but in EC ontology it becomes a supporting

role. The concept *Structure* can be specified by different criteria such as topology in CATH, bond type in EC, and so on, thus creating different specification paths in the mediated ontology. The same principle can be applied in the reverse (abstraction) path. For example, one specific structure concept of the protein can be "zoom out" into the disease concept or the pathway concept.

The above different paths of the concept navigation provide the basis for the customized view of domain concepts for querying and integration. In the rest of this chapter, we will use biological studies of aging as an example to illustrate the application of our proposed mediated taxonomy/classification approach to integrate the related data sets.

Aging is a process that is complex and multi-disciplinary. The data that describes the aging process span over different domains and abstraction levels from genomic and proteomic experimental results reported in the aging research literature to various online database resources. They include aging-related knowledge discovery in research literatures such as Telemakus [58], organelle databases such as MitoMap [79] that contain mitochondrial proteins of humans and many other general databases that contain genes/proteins with a role in aging. Usually scientists researching on proteins are often interested in sets of proteins, sharing certain properties such as folding structure, or enzyme reaction and doing comparative studies on these data instances in other prospects such as source organism, protein-protein interaction or pathways. Some typical questions arise. Examples in mitochondria related aging process are:

1. Retrieve all variants (isozymes) of Insulin-like Growth Factor (IGF)?
2. Do all organism have similar or same isozymes?
3. Do these isozymes interact with same receptor?
4. What differences (structure or function) are there among these IGF receptor?

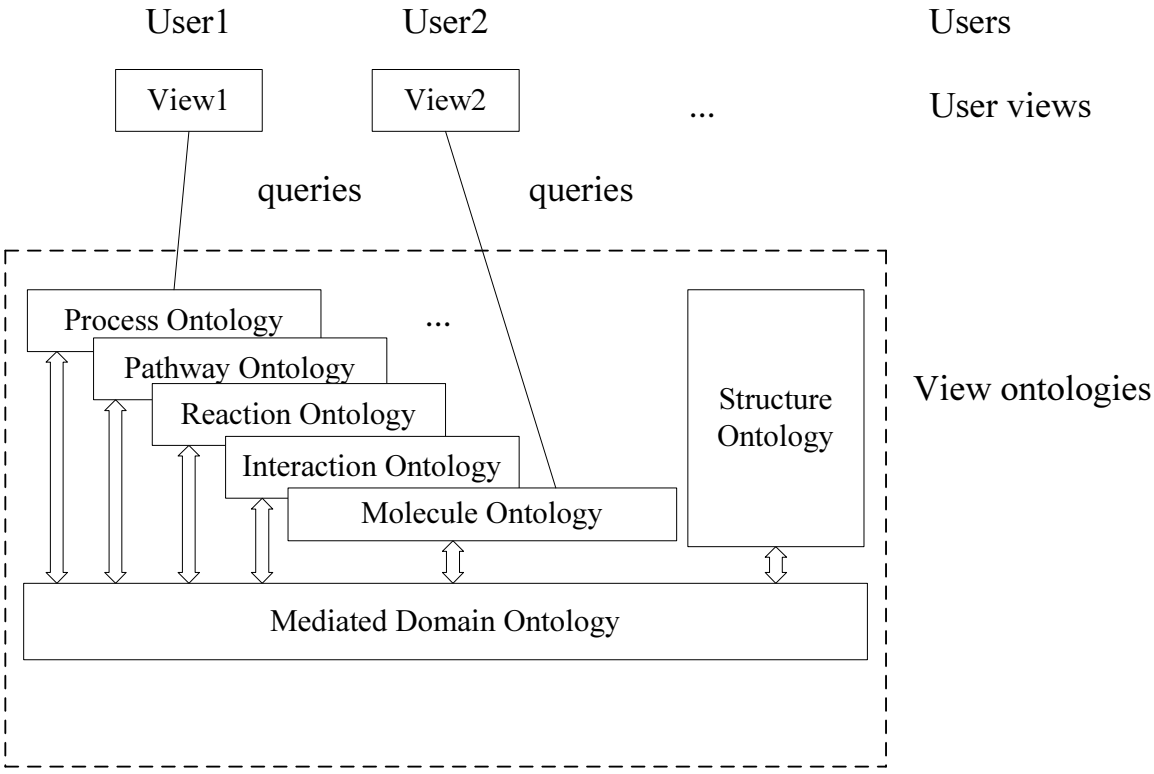


Figure 4.3. Mediator logical architecture.

5. Retrieve all structures of IGF binding proteins in the mitogen-activated protein kinase (MAPK) pathways.

The above questions can be formulated using SQL queries with concept relationship constraints. For example, question no.1 can be translated into the following SQL query:

```
SELECT  m2.name (or accession) as isozymes
FROM    molecule m1, molecule m2
WHERE   m1.name = 'IGF'
AND     (m1.accession, m2.accession) IN isozyme_view
```

Figure 4.3 show the mediator logical system architecture. Each type of query represents one specific view of the user. For example, view1 may focus on the struc-

ture view of data constrained by other concepts (attribute values and domain relationships)

4.5 Summary

In this chapter, we described how we created our mediated ontology from the many ontology databases that exist. The mediated ontology is the heart of our integrated querying system, which guides the user through query formulation by selecting ontology terms of interest. The user selections are converted into an SQL query on the mediated ontology, which retrieves various accession numbers from diverse data sources that match the user query. These are described in the following chapters.

CHAPTER 5

BIOMEDIATOR SYSTEM

This chapter gives the overview of BioMediator system architecture and the functions of each component. The mediator system has been implemented in C sharp language, running on .NET Framework 2.0 in the windows system. The mediated domain ontology described in the previous chapter is stored on the MySQL server. The user interface consists of several ASP.NET web pages, hosted by IIS web server. Figure 5.1 shows the overall architecture. Section 5.1 presents the Domain Ontology Server that stores the MDO and the instance data. Section 5.2 presents the User Query Interface that the user can use to formulate queries by browsing the domain ontology concepts. Section 5.3 presents the Query Processor that executes the user queries and returns the results back to the user. Section 5.4 presents the Service Data Retriever that retrieves the data requested by the user. Section 5.5 presents the external web services provider: BioServiceBroker [28].

5.1 Domain Ontology Server

In the previous chapter, we discussed the mediated domain ontology. The MDO can be implemented into a relational database server using the Resource Description Framework (RDF) data model. This domain ontology server stores all the mediation data, and manages the domain concepts, their properties, and relationships, as well as the information about the mapping to the data sources. The next section will briefly describe the basic concepts of the RDF data model, and discuss why we adopt it.

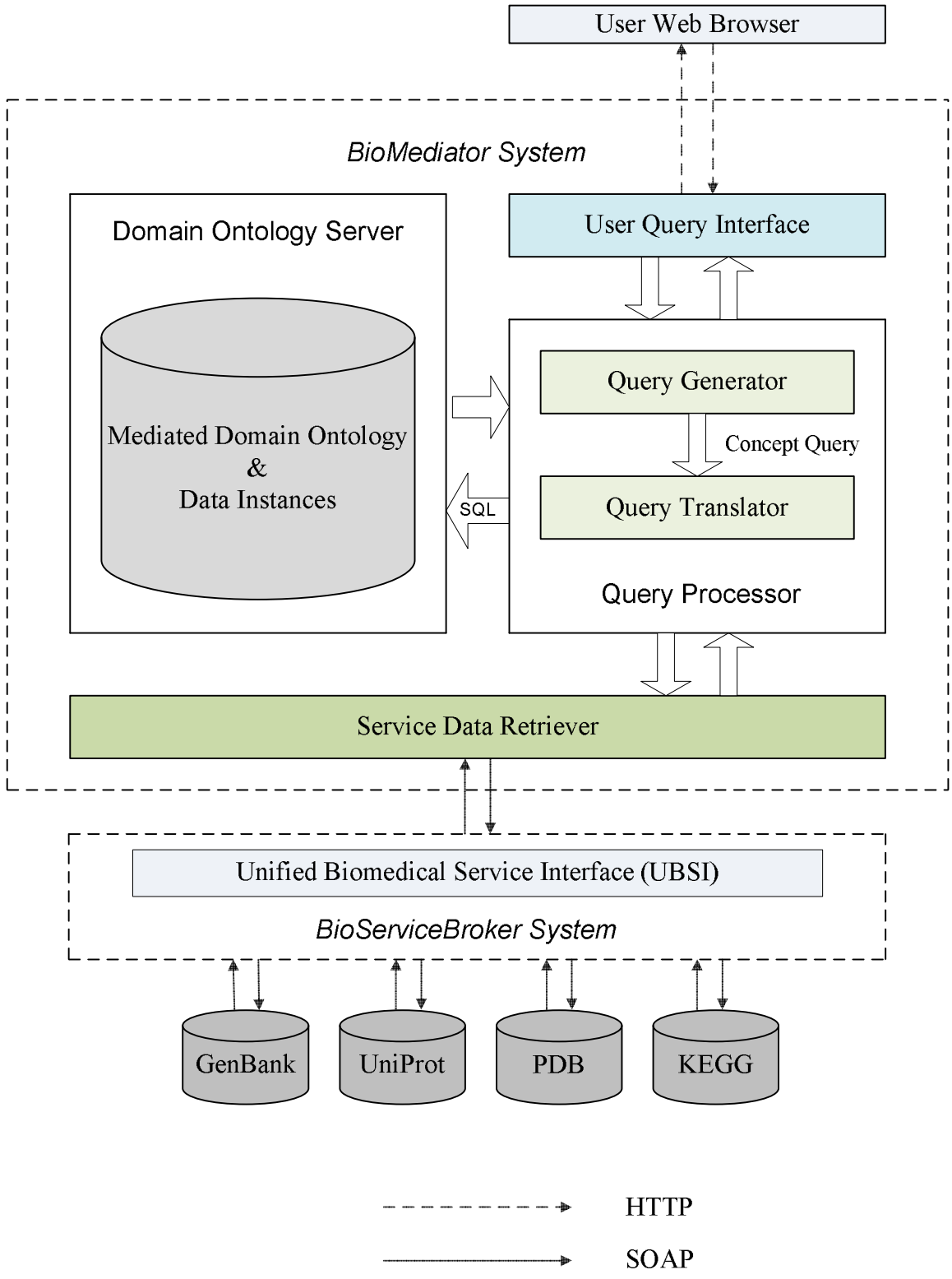


Figure 5.1. Mediator system architecture.

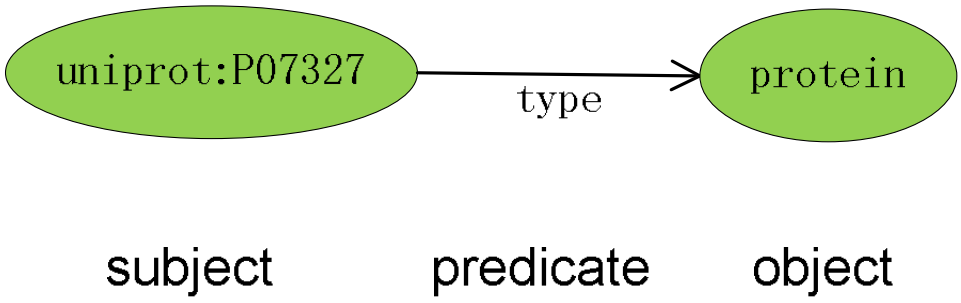


Figure 5.2. RDF graph data model in triple.

5.1.1 RDF Data Model

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It is a data model used by the semantic web technology. It is simple graph data model, shown in Figure 5.2. A triple has three parts: a subject, an object and a predicate that denotes the relationship between the subject and the object. The arrow is always pointing toward the object. So, this graph describes:
an instance **uniprot:P07327**'s **type** is a **protein**. Note that in an RDF graph, a node may be a URI (Uniform Resource Identifier) with optional fragment identifier (URI reference, or URIref), a literal, or blank.

As we discussed in chapter 2, bioinformatics data is not like traditional commercial data. The entity types are easy to determine, but its attributes are relatively difficult to identify. Relationships at the data instance level, such as one protein instance interacts with other protein instances, are the primary concerns of the scientists. These various relationships cannot be easily captured or modeled at the schema level. Figure 5.3 shows an RDF graph that describes various information about the protein instance: uniprot:P07327. The rectangle nodes denote literal values. The content can be expressed by a collection of RDF statements in Table 5.1. Each statement is a triple of the subject-predicate-object form. It asserts the follow-

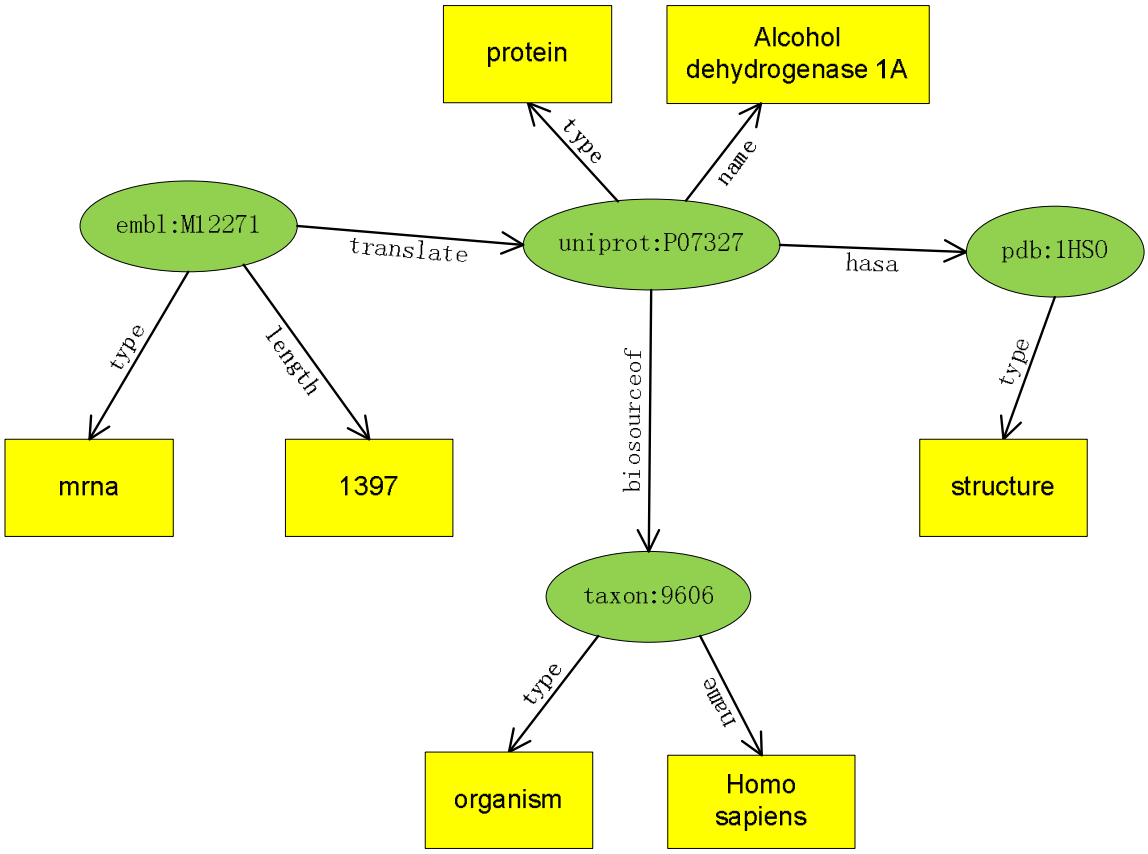


Figure 5.3. RDF graph about a protein instance.

ing facts: uniprot:P07327 is a protein; its name is "Alcohol dehydrogenase 1A"; it has a pdb:1ZT3 3D molecular structure; and so on.

Since the bioinformatics data is a mixture of raw data instances and annotations, we use this simple RDF model to store various relationships between data instances. We do not use ordinary tables to store bio-entities (data warehouse approach). The data warehouse approach uses a pre-defined and unified data schema that specify the entities and relationships at the schema level. Our RDF approach stores these relationships at the instance level. Different bio-entity types (entity

Table 5.1. RDF statements about a protein instance

(uniprot:P07327) (type) (protein)
(uniprot:P07327) (name) (Alcohol dehydrogenase 1A)
(uniprot:P07327) (biosourceof) (taxon:9606)
(uniprot:P07327) (hasa) (pdb:1HSO)
(taxon:9606) (type) (organism)
(taxon:9606) (name) (human)
(embl:M12271) (type) (mrna)
(embl:M12271) (length) (1397)
(embl:M12271) (translate) (uniprot:P07327)
(pdb:1HSO) (type) (structure)

concepts in MDO) and their associated attributes can be implemented by database views.

5.1.2 Mediator Data Schema

Figure 5.4 shows the conceptual ER model of the mediator schema. Entity CONCEPT stores all the core concepts. The attribute *Level* denotes the abstraction level of the concept. The attributes *CType*, *CType2* and *CType3* denote the different types of the same concept instance. For example, the concept *enzyme CType* value is *protein*, *CType2* value is *catalyst*, and *CType3* value is *null*. Note that many of these values could be NULLs. This design gives us the possibility of implementation for multiple inheritance concept hierarchies (*typeof* or *isa* relationship). It also provides a way of increasing selectivity of the target data instance for the queries. Table 5.2 shows some examples of domain concepts.

Relationship *Relation* stores all possible binary relationships between two concepts except for *isa* relationship. Table 5.4 shows some examples of relationships between two concepts. The rest of the concepts that come from external sources (classification or annotation terms) and attribute concepts such as *Name*, *Symbol*,

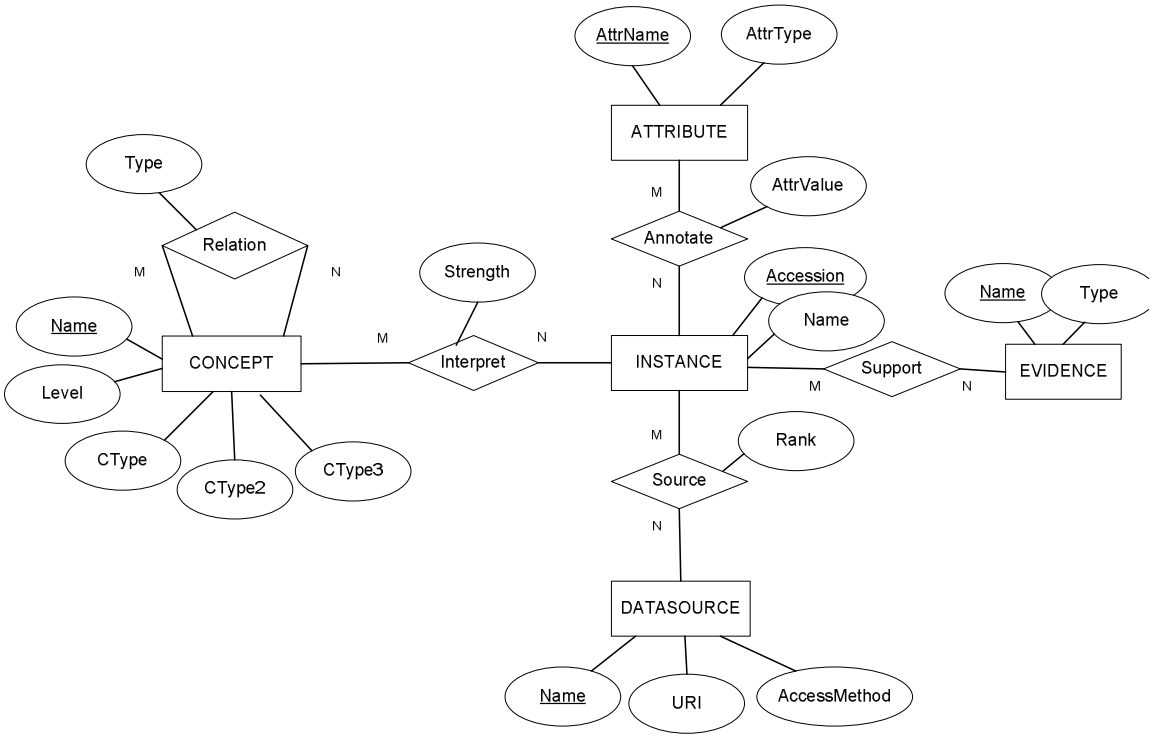


Figure 5.4. ER model of the mediator schema.

Length, and so on are stored in the entity ATTRIBUTE. Their value can be used to annotate the instance data stored in the entity INSTANCE with the key attribute: *Accession* of external data sources. Data instances can be either directly supported by various evidence records (stored in the entity EVIDENCE) such as experimental results written in the research literatures, or conceptually interpreted by the core entity concepts.

Note that we put the attribute names as values in the ATTRIBUTE table, as shown in Table 5.3 for the joined results. This design is beneficial for the mediation system since there is no need to store all the values of attributes, and some attributes of the instance will change in the future or are unknown at present but will be added later on. This allows the flexibility to add new attributes in the future as needed.

Table 5.3 shows the examples of records in the join table INSTANCE-ATTRIBUTE. Note that *AttrVal* usually is a coding number, such as EC number 1.1.1.1, which is long text. Table 5.5 shows some examples of instance-level relationships.

5.2 User Query Interface

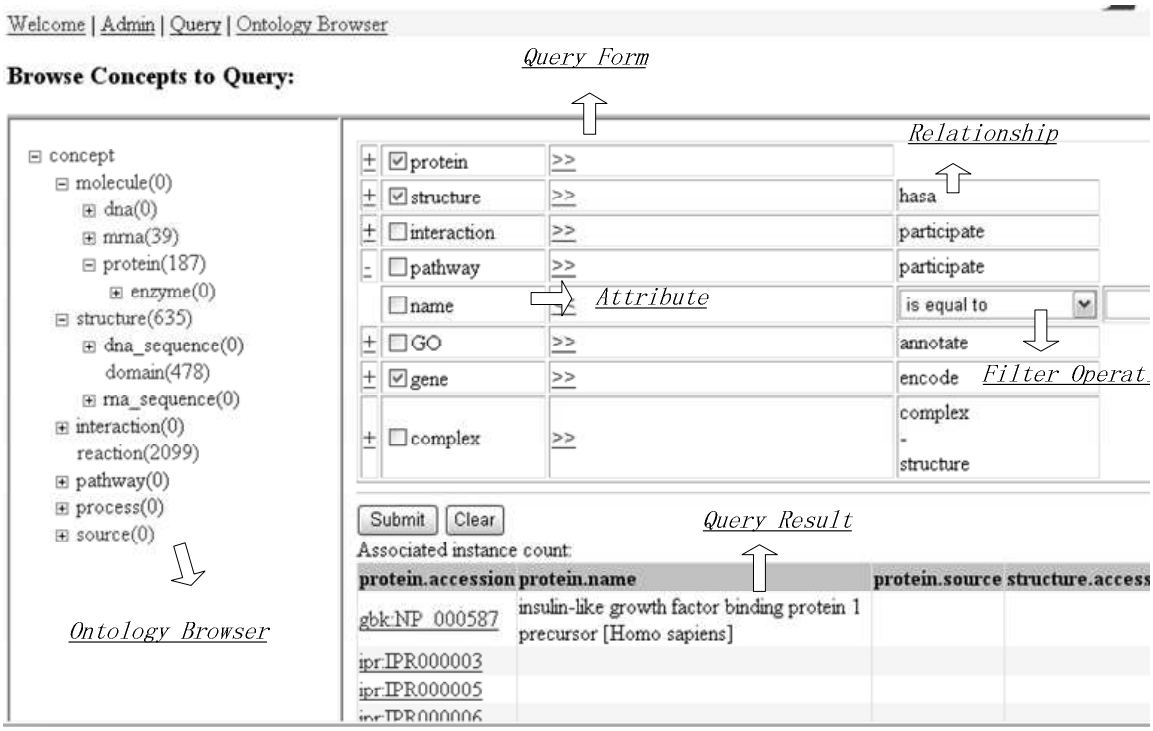


Figure 5.5. A web interface for ontology browsing and querying.

The main user interface is a query interface for browsing concepts and formulating queries, shown in Figure 5.5. It consists two main parts: Ontology Browser and Query Form.

Ontology Browser (left-hand frame) gives users a tree view of the entity concepts. Users can browse it to find specific concepts through several high level ac-

cess points: *molecule*, *structure*, *interaction*, *reaction*, *pathway*, *process*, *disease*, and *source*. These concepts are at the first level, the rest of the more specific concepts are built up by *isa* relationships under the core concepts. Note that the displayed concepts in the tree view are just one subset of the total concepts.

Once a node is clicked, the concept becomes the main focus when users want to make queries about it. We call it the center concept. This concept, a list of related concepts, and their corresponding relationships will be shown in Query Form (right-hand frame). A list of attributes of each concept can be vertically expanded/collapsed by clicking the "+/-" sign beside it. Also a list of filter operations corresponding to each type of attributes can be horizontally expanded/collapsed by clicking the "+/-" sign beside it. For examples, the operations: "equal to" and "in" will be applied to identifier type attribute such as *accession* (for a database entry); the operations: "equal to" and "like" will be applied to string type attribute such as *name*; the operations: "equal to", "less than", "greater than" and "between" will be applied to integer type attribute such as *length* and *weight*. The users can select concepts and attributes of interest for the query.

The users can customize the domain concepts according to their own view of data. The users can select any members (typeof) of these concepts to form the first level concepts. For example, if a user wants to query the information about one class of proteins, their 3D structures, genes, and associated diseases, he or she can do the following: In the molecule domain ontology, select *protein*. This is the center concept. The other selected concepts are the context (associated or constraint concepts). In the structure domain ontology, select *protein 3D structure*. In the disease domain ontology, select *disease*. This set of entity concepts: *protein*, *protein 3D structure*, and *disease* constitutes the first level concepts, which will be displayed in the Ontology Browser. After the user sets up the initial concepts set, the system

will automatically calculate the first level instance data set: protein instances, protein 3D structure instances, and disease instances, and relationship data set: protein-structure instances; protein-disease instances with all known available data sources.

5.3 Query Processor

Currently, the Query Processor module has two components: the concept Query Generator and the Query Translator. Next, we will discuss their individual functions.

5.3.1 Concept Query Generator

The concept Query Generator module captures all the information that users specified in the browsing query interface, and transforms it into a concept query. Then, the Query Translator translates this concept query into a internal SQL query that can be efficiently executed against the underlying relational database.

The syntax of the concept query is similar to the syntax of SQL with a little extension. We add RDF like triples into WHERE clause to specify the relationships among entity concepts. The following examples show the above user query and the transformed concept query.

```
user query:
Find name, RNA transcript, 3D structure of the
     protein with UniProt accession: P07327.


concept query:
SELECT protein.accession, protein.name,
       rna.accession, structure.accession
FROM   protein, rna, structure
```

```
WHERE  protein.accession = 'uniprot:P07327'
AND    ( rna <translate> protein
         OR protein <hasa> structure )
```

So why do we need the concept query? Some types of queries will be repeated just by changing the input parameters. The user can save his/her selections from the query form to a concept query script. This script can be execute directly in concept query interface by typing in or loading from a disk. If users get familiar with the concept syntax and names of the entity, attribute and relationship concepts, they can write the query directly.

5.3.2 Query Translator

The query translator module translates the concept query into the internal SQL queries. The following examples show how the above concept query is translated into an SQL query.

```
concept query:
SELECT protein.accession, protein.name,
       rna.accession, structure.accession
FROM   protein, rna, structure
WHERE  protein.accession = 'uniprot:P07327'
AND    ( rna <translate> protein
         OR protein <hasa> structure )

SQL query:
SELECT protein.accession, protein.name,
       rna.accession, structure.accession
FROM   protein
```

```
       LEFT JOIN irelation ON ( protein.instance_id =
       irelation.instance_id1 ) LEFT JOIN structure ON
       ( irelation.instance_id2 = structure.instance_id ) ...
WHERE  protein.accession = 'uniprot:P07327'
AND    ( irelation.relationship = 'translate' OR
         irelation.relationship = 'hasa' )
```

Note that in the above SQL query, entity protein, rna, structure, etc are not physical tables, they are database views, generated by the user who makes a query when these entities appeared in the concept query for the first time. These views can be automatically updated by the local database engine when new data are added in. Here is an SQL example to generate the protein view:

```
CREATE VIEW protein AS
SELECT i.instance_id, i.accession,
  (CASE WHEN i_a.attr_id = 1 THEN i_a.attr_value ELSE null END)
  AS Name,
  (CASE WHEN i_a.attr_id = 5 THEN i_a.attr_value ELSE null END)
  AS Length,
  ...
  (CASE WHEN i_a.attr_id = 31 THEN i_a.attr_value ELSE null END)
  AS GoAnnotation
FROM  instance i, instance_attribute i_a
WHERE i.instance_id = i_a.instance_id
AND   i.concept_name = 'protein'
```

5.4 Service Data Retriever

The Service Data Retriever component consumes the result of query processing that contains the remote data source information, such as APIs, web services, and retrieve data in XML format through HTTP protocol.

Figure 5.6. Analysis of a UniProt XML file.

Currently, the supported web service operation is the UniProt fetch service. The accepted parameter is the protein accession, like "uniprot:P07327". The return

data set is one or more database entries in XML format. A parser is automatically called to parse this protein entry to extract *EC* instances, *RC* instances, and *AC* instances and populate these data sets into the mediator system. Figure 5.6 shows the analysis of a typical UniProt protein entry in XML format. The following is the output data sets. Note that the symbol | is a separator.

```
instance records:
uniprot:P07327|ADH1A_HUMAN|protein
taxonomy:9606|Homo sapiens|organism
embl:M12271|NULL|mrna
...
instance-attribute records:
uniprot:P07327|name|alcohol dehydrogenase subunit alpha
...
instance-relation records:
uniprot:P07327|encode|unigene:hs.654433
uniprot:P07327|biosourceof|taxonomy:9606
uniprot:P07327|translate|embl:M12271
uniprot:P07327|molecule-structure|pdb:1HSO
uniprot:P07327|molecule-structure|pdb:1U3T
uniprot:P07327|protein-disease|OMIM:103700
uniprot:P07327|protein-pathway|KEGG:hasa:124
uniprot:P07327|protein-drug|PharmGKB:PA24570
...
```

5.5 BioService Provider

This section briefly describes the server side of BioService system: BioService-Broker [28], its functionality and services. The BioMediator's external data access is through the web services provided by BioServiceBroker. BioServiceBroker is a multi-level ontology-enabled service broker for dynamically integrating web services in the bioinformatics domain. It has a UBSI (Unified Biomedical Service Interface) for clients to invoke many published service operations. Our Service Data Retriever dynamically retrieves data by calling UBSI's operations. For example, to retrieve the sequence of a specific protein, Our Service Data Retriever calls *getProteinSequence* of UBSI; the input parameter is the protein database accession number.

Table 5.2. CONCEPT relation

Label	Name	CType	CType2	CType3
entity	entity			
attribute	attribute			
relationship	relationship			
source	source			
role	role			
data source	dataSource	source		
biological source	bioSource	source		
...				
molecule	molecule	physicalEntity		
structure	structure	entity		
interaction	interaction	relationship	entity	
reaction	reaction	entity		
pathway	pathway	entity		
process	process	entity		
...				
catalyst	catalyst	role	molecule	
...				
protein	protein	molecule		
enzyme	enzyme	protein	catalyst	
DNA	dna	molecule		
RNA	rna	molecule		
...				
chemical structure	chemicalStructure			
...				
organelle	organelle	bioSource		
cell	cell	bioSource		
tissue	tissue	bioSource		
organ	organ	bioSource		
organism	organism	bioSource		
...				

Table 5.3. INSTANCE-ATTRIBUTE relation

Accession	AttriuteName	AttributeValue
spt:P07327	EC-annotation	1.1.1.1
gbk:gene3484	symbol	IGFBP1
gbk:NM-000596	length	1660 bp
spt:P14139	EC-annotation	6.6.1.2
spt:P07327	CATH-annotation	1.10.8.10
omim:609342	MESH-annotation	C06.405.748.398
...		

Table 5.4. CONCEPT-CONCEPT relation

Concept	Concept	Relationship
bond	molecular structure	partof
atom	molecule	molecular structure
gene	DNA	ordered-bag
gene	protein	encode
gene	mRNA	transcript
...		

Table 5.5. INSTANCE-INSTANCE relation

Instance	Instance	Relationship
gbk:gene3484	spt:P07327	encode
gbk:gene3484	gbk:NM-000596	transcript
gbk:NM-000596	spt:P07327	translate
spt:P07327	spt:P08833	isozyme
...		

CHAPTER 6

APPLICATIONS

This chapter describes how to use BioMedaitor to query for bioinformatics data sources. We use some examples to illustrate the process of query formulation.

6.1 Customizable Query Formulation

The web site of major bioinformatics data sources usually provides an ontology interface to navigate their data. Each concept is associated with a collection of database entries. High level concepts associate more data instances, such as in GO. The problem is when you browse the concept tree, you must go down many levels of hierarchy to find the specific term of the interest. When you click the term, the system will return a web page full of links to the external databases. The data sources to which they linked span over various domains on different levels of abstraction. For example, in NCBI UniGene web site, one gene can sometimes link to more than 50 nucleotide sequences, but there is no indication of any differences among them. Most of the time, users must check these links one by one to see if there are any interest in the data. Actually, these browsing results can be conceptually modeled in a framework of a mediated ontology. The same instance level of data can be differentiated from each other in the context of this framework for multiple comparisons.

Our query interface provides a "browse-and-query" model for users to locate the data of their interest. The browsing interface provides several levels of access points, from individual molecules to integrated pathways and processes. Once you choose

the main concept of interest, the other related concepts will become a context in a query. This context can be classified into 2 types: the association and the constraint. The next two sections will illustrate these concepts.

6.1.1 Association Queries

The Association Query means retrieving all relevant information about one type of data instances. It includes: proteins, protein structures, enzymatic reactions, protein annotations, etc. Usually, the query includes a center concept and other associated concepts, and the return tuple set is only limited by the center concept.

EXAMPLE 1. *Find all information about genes and structures of "IGFBP" proteins.*

We can first browse the concept tree to locate the *protein* concept. Once found, we expand the concept node to see if there are any classification called "IGFBP". Once found, click the IGFBP concept node to check if there are any attributes or related concepts of interest. Suppose that the *gene* concept related by *encode* is the user interest. Then the user can select these attributes to query the system. In the query, the "IGFBP" protein family limits the returned set of protein instances and any matching gene instances and structure instances will be returned whenever a "IGFBP" protein instance is retrieved. The system will return a list of protein accession numbers and the values of the selected attribute. Some values of genes and structures may be empty. The formulated query in concept query format is as follows:

```
SELECT protein.accession, protein.name,
       gene.accession, structure.accession
FROM   protein, rna, structure
WHERE  protein.type = 'IGFBP'
```

```
AND    ( gene <encode> protein
         OR protein <hasa> structure )
```

6.1.2 Constraint Queries

The Constraint Query means the context concepts restrict the return set of main concept instances. The conditions specified in other related concepts also affect the main concept.

EXAMPLE 2. *Find proteins with some characteristic structure specified by CATH 3D domain classification.*

In this query, the user can first browse from *structure* concept, then locate and select *domain structure*. The *protein* concept will appear as a related concept to *domain structure*. The user can check the attribute *CATH-annotation* of *domain structure*, and specify the conditions. The following is the formulated query.

```
SELECT protein.accession, protein.name,
FROM   protein, structure
WHERE  structure.CATH-annotation = 'CATH:T556'
AND    protein <hasa> structure
```

6.2 Browsing-based Data Integration

Our mediator only stores the identifers of database entries from various data sources, and associated annotations or classifications. These identifers belong to different types of the data, and are integrated under a mediated ontology. Extra data such as sequences of proteins or nucleotides are not stored locally. We adopt the navigational or linked-based integration. After the users submits the query, the system will return a set of database accessions in a tabular format. The detailed reports of database entries can be browsed by the links. If the return set is beyond

the user comprehension, the result can be further filtered out by applying the various filter operations on the concepts's attributes.

Currently, our mediator integrates the bioinformatics data from the following sources: UniPort, PDB, ENZYME, CATH, and GO. Annotation data sets including the database accessions are reloaded often to keep the data updated with new release of each source. Because bulk downloads and parsing for accessions and annotations are one time operations, the mediator does not incur much maintenance problems.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis, first we studied the characteristics of bioinformatics data, various data exchange formats, and data models used for sequences, structures, interactions, pathways and processes. Conceptual data models such as EER lack the representation power for capturing these biological concepts. We proposed the enhanced EER model and diagrams to address this problem. Then, we studied the problems with querying and integrating the above resources in various integrative systems. Most systems lack the flexible query interface for the biologist to make queries from his or her view of data relationships, and thus can not fully integrate different types of raw data in the various contexts. We found that ontology-based and web-service-assisted mediator architecture is more suitable for integrating often updated data sources. For this reason, we focused on the development of the mediated domain ontology that uses our new proposed relationships. Our mediator system is implemented based on this ontology, which helps the query interface design and use.

7.1 Summary of Contributions

- EER Data Model Enhancements. We introduced three new types of relationships into the EER model: ordered relationship, process relationship and molecular spatial relationship. We also extended the relationships to allow bags (or multi-sets) of relationship instances, since many relationships in molecular biology fall into this category. We illustrated the need for these relationships in modeling biological data and we proposed some special diagrammatic notation.

By introducing these extensions we anticipate that biological data having these properties will be made explicit to the data modeler, which would help direct future biological database implementation. In particular, our unordered bag relationship can be used for various reaction data, and our ordered bag should be useful for sequences and genetic features. Our process relationship would be useful for the reaction, interaction and pathway data. These changes do not add much complexity to the existing EER model, thus making them easier for integration. We also give the formal definitions for these new concepts and summarized their notation and usage. We also showed how these additional concepts can be mapped into relations for implementation in relational databases.

- We proposed the mediated domain ontology for the purpose of bioinformatics data query and integration. It consists of two sets of concepts: the instance-supported domain concepts (core), and instance-associated annotation/classification concepts (external). The ontological concepts can be roughly classified into 3 types: entity concepts, attribute concepts, and relationship concepts. Core entity concepts such as *Protein*, *Nucleotide*, *Structure*, *Interaction*, *Reaction*, *Pathway*, *Process*, *BioScource* and *DataScource*, are manually set up by the analysis of various molecular database entries. Attribute concepts include the common attributes of the above entity concept instances, and standard annotation concepts. External annotation concepts such as in GO and in ChEBI can be queried and downloaded automatically through the web service provider based on the user needs. We apply RDF data model in the design of the mediator schema. It is based on hybrid taxonomy ontologies for integration of the protein and gene instance data in the context of interactions, pathways and processes.

- BioMediator Querying and Browsing Interface. A prototype BioMediator system has been built for biologists to navigate the domain concepts and construct queries. The system pre-stores the domain concepts about bioinformatics data sources and their database entries. The users can formulate the queries by browsing the concept tree and selecting the specific concepts of interest and its attributes and related concepts. After submitting a query, the system will return a list of accession numbers from the different data sources. Users can click the links to check the detailed reports of that data entry. Extra attribute data will be retrieved via external BioService provider [28].

7.2 Future Research Direction

Currently, we only experimented with a subset of data from each source. It is possible that with more and more integrated data, the query performance will degrade. Even though the RDF model can provide a way for making flexible query feasible, efficient indexing and querying these millions of instances will become a problem. So, subsequent work should focus on the indexing of this pseudo RDF store built on the relational databases.

REFERENCES

[1] Chebi tutorial. http://www.ebi.ac.uk/chebi/tutorialForward.do.

[2] Medical subject headings (mesh). http://www.nlm.nih.gov/mesh/.

[3] Ncbi entrez web service. http://eutils.ncbi.nlm.nih.gov/.

[4] Object management group life sciences identifiers specification. http://www.omg.org/cgi-bin/apps/doc?dtc/04-05-01.pdf.

[5] Swiss-prot protein knowledgebase, http://ca.expasy.org/sprot/.

[6] Trembl computer-annotated supplement to swiss-prot, http://ca.expasy.org/sprot/.

[7] Andreeva A., Howorth D., Chandonia J.-M., Brenner S.E., Hubbard T.J.P., Chothia C., and Murzin A.G. Data growth and its impact on the scop database: new developments. *Nucleic Acids Research*, 36, Database issue:D419–D425, 2008.

[8] Bairoch A. The enzyme database in 2000. *Nucleic Acids Research*, 28:304–305, 2000.

[9] Gary D. Bader, Erik Brauner, Michael P. Cary, Robert Goldberg, Chris Hogue, Peter Karp, Joanne Luciano, Debbie Marks, Natalia Maltsev, Eric Neumann, Suzanne Paley, John Pick, Aviv Regev, Andrey Rzhetsky, Chris Sander, Vincent Schachter, Imran Shah, Mustafa Syed, and Jeremy Zucker. Biopax : Biological pathway exchange.

[10] M. Baitaluk, M. Sedova, A. Ray, and A. Gupta. Biologicalnetworks: visualization and analysis tool for systems biology. *Nucleic Acids Res.*, 34:W466–W471, 2006.

[11] Tanya Barrett, Dennis B. Troup, Stephen E. Wilhite, Pierre Ledoux, Dmitry Rudnev, Carlos Evangelista, Irene F. Kim, Alexandra Soboleva, Maxim Tomashevsky, and Ron Edgar. Ncbi geo: mining tens of millions of expression profiles database and tools update. *Nucleic Acids Research*, 36, Database issue:D760–D765, 2007.

[12] Zina Ben-Miled, Nianhua Li, Yang Liu, Yue He, Eric Lynch, and Omran A. Bukhres. On the integration of a large number of life science web databases. In *Data Integration in the Life Sciences*, pages 172–186, 2004.

[13] Dennis A. Benson, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and David L. Wheeler. Genbank. *Nucleic Acids Research*, 34, Database issue:D16–D20, 2006.

[14] Deepavali Bhagwat, Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vijayvargiya. An annotation management system for relational databases. In *Proceedings of the 30th VLDB Conference*, pages 900–911, 2004.

[15] Ewan Birney and Michele Clamp. Biological database design and implementation. *Briefings in Bioinformatics*, 5(1):31–38, 2004.

[16] Erich Bornberg-Bauer and Norman W. Paton. Conceptual data modelling for bioinformatics. *Briefings in Bioinformatics*, 3(2):166–180, 2002.

[17] Philip E. Bourne, John Westbook, and Helen M. Berman. The protein data bank and lessons in data management. *Briefings in Bioinformatics*, 5(1):23–30, 2004.

[18] James M. Brundege and Christopher Dubay. Bioquery: an object framework for building queries to biological databases. *Bioinformatics*, 19(7):901–902, 2003.

[19] Francois Bry and Peer Kroger. A computational biology database digest: Data, data analysis, and data management. *Distributed and parallel databases*, 13:7–42, 2003.

[20] K.J. Bussey, D. Kane, M. Sunshine, S. Narasimhan, S. Nishizuka, W.C. Reinhold, B. Zeeberg Ajay, and J.N. Weinstein. Matchminer: a tool for batch navigation among gene and gene product identifiers. *Genome Biology*, 4(4):R27, 2003.

[21] David Buttler, Matthew Coleman, Terence Critchlow, Renato Fileto, Wei Han, Calton Pu, Daniel Rocco, and Li Xiong. Querying multiple bioinformatics information sources: can semantic web research help? *SIGMOD Rec.*, 31(4):59–64, 2002.

[22] C.I. Castillo-Davis and D.L. Hartl. Genemerge - post-genomic analysis, data mining, and hypothesis testing. *Bioinformatics*, 19(7):891–892, 2002.

[23] CATH. Cath classification. http://www.cathdb.info/latest/index.html/.

[24] Wu CH, Yeh LS, Huang H, Arminski L, Castro-Alvear J, Chen Y, Hu Z, Kourtesis P, Ledley RS, Suzek BE, Garrett L, Vinayaka CR, Zhang J, and Barker WC. The protein information resource. *Nucleic Acids Research*, 31, Database issue(1):345–347, 2003.

[25] I-Min A. Chen and Victor M. Markowitz. An overview of the object-protocol model (opm) and opm data management tools. *Inf. Syst.*, 20(5):393–418, 1995.

[26] Jake Yue Chen and John V. Carlis. Genomic data modeling. *Information Systems*, 28(4):287–310, 2003.

[27] Kei-Hoi Cheung, Kevin Y. Yip, Andrew Smith, Remko deKnikker, Andy Masiar, and Mark Gerstein. Yeasthub: a semantic web use case for integrating data in the life sciences domain. *Bioinformatics*, (21)i85-i96, 2005.

[28] Sheng chieh Jack Fu. Uta phd dissertation: A multi-level biomedical ontology-enabled broker: dynamic service-based data source integration, 2008.

[29] The Gene Ontology Consortium. The gene ontology project in 2008. *Nucleic Acids Research*, 36, Database issue:D440–D444, 2008.

[30] The UniProt Consortium. The universal protein resource (uniprot). *Nucleic Acids Research*, 35(Database issue):D193–D197, 2007.

[31] Kirill Degtyarenko1, Paula de Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcntara, Michael Darsow, Mickal Guedj, and Michael Ashburner. Chebi: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, 36, Database issue:D344–D350, 2008.

[32] M. Diehn, G. Sherlock, G. Binkley, H. Jin, J.C. Matese, T. Hernandez-Boussard, C.A. Rees, J.M. Cherry, D. Botstein, P.O. Brown, and A.A. Alizadeh. Source: a unified genomic resource of functional annotations, ontologies, and gene expression data. *Nucleic Acids Research*, 31(1):219–223, 2003.

[33] Sorin Draghici, Sivakumar Sellamuthu, and Purvesh Khatri. Babel's tower revisited: a univeral resource for cross-referencing across annotation databases. *Bioinformatcs*, 22(23):2934–2939, 2006.

[34] Karen Eilbeck, Suzanna E Lewis, Christopher J Mungall, Mark Yandell, Lincoln Stein, Richard Durbin, and Michael Ashburner. The sequence ontology: a tool for the unification of genome annotations. *Genome Biology*, 6:R44, 2005.

[35] Ramez Elmasri, Jack Fu, and Feng Ji. Multi-level biomedical ontology-enabled service broker for web-based interoperation. In *23rd Annual ACM Symposium on Applied Computing (SAC 2008)*, pages 2341–2345, Ceara, Brazil, 2008.

[36] Ramez Elmasri, Jack Fu, and Feng Ji. Multi-level conceptual modeling for biomedical data and ontologies integration. In *The 20th IEEE International*

Symposium on Computer-based Medical Systems (CBMS 2007), pages 589–594, Maribor, Slovenia, June 20-22, 2007.

[37] Ramez Elmasri, Jack Fu, Feng Ji, and Qing Li. Bioso: Bioinformatic service ontology for dynamic biomedical web services integration. In *Biotechnology and Bioinformatics Symposium (BIOT 2007)*, Colorado Springs, Colorado, 2007.

[38] Ramez Elmasri, Jack Fu, Feng Ji, Yiming Zhang, and Zoe Raja. Extending eer modeling concepts for biological data. In *The 19th IEEE International Symposium on Computer-based Medical Systems (CBMS 2006)*, pages 599–604, Salt Lake City, Utah, 2006.

[39] Ramez Elmasri, Feng Ji, Jack Fu, Yiming Zhang, and Zoe Raja. Modelling concepts and database implementation techniques for complex biological data. *Int. J. Bioinformatics Research and Application*, 3(2):366–388, 2007.

[40] Ramez Elmasri and Navathe S.B. *Fundamentals of Database Systems*. Addison-Wesley Publishing Company, Boston, MA, 2006.

[41] ENZYME. Swissprot enzyme database. http://www.expasy.ch/enzyme/.

[42] Alfarano C. et al. The biomolecular interaction network database and related tools 2005 update. *Nucleic Acids Res.*, 33(Database issue):D418–24, 2005.

[43] Carola Kanz et al. The embl nucleotide sequence database. *Nucleic Acids Research*, 33, Database issue:D29–D33, 2005.

[44] Chong Su et al. Bacteriome.organ integrated protein interaction database for e. coli. *Nucleic Acids Research*, 36(Database issue):D632–D636, 2008.

[45] D. Karolchik et al. The ucsc genome browser database: 2008 update. *Nucleic Acids Research*, 36, Database issue:D773–D779, 2008.

[46] David L. Wheeler et al. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 35, Database issue:D5–D12, 2007.

[47] G. Joshi-Tope et al. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database Issue):D428–D432, 2005.

[48] H. Stuckenschmidt et al. Index structures and alogorithms for querying distributed rdf repositories. In *Proceedings of WWW2004*, pages 631–639, 2004.

[49] Ivliev AE et al. Microarray retriever: a web-based tool for searching and large scale retrieval of public microarray data. *Bioinformatcs*, 36(Web Server issue):W327–31, 2008.

[50] Minoru Kanehisa et al. Kegg for linking genomes to life and the environment. *Nucleic Acids Research*, 36, Database issue:D480–D484, 2008.

[51] P. Flicek et al. Ensembl 2008. *Nucleic Acids Research*, 36, Database issue:D707–D714, 2008.

[52] P. Mork et al. The multiple roles of ontologies in the biomediator data integration system. In *Proceedings of the Data Integration in the Life Sciences Workshop*, 2005.

[53] Peter Karp et al. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, 36, Database issue:D623–631, 2008.

[54] R. Stevens et al. Semantic webs for life sciences. In *Pacific Symposium on Biocomputing 2006*, pages 112–115. World Scientific Publishing Company, 2006.

[55] Robert D. Finn et al. Pfam: clans, web tools and services. *Nucleic Acids Research*, 34, Database issue:D247–D251, 2006.

[56] S. Kerrien et al. Intact-an open source molecular interaction database. *Nucleic Acids Research*, 35, Database issue:D561–D565, 2007.

[57] Zdobnov et al. The ebi srs server recent developments. *Bioinformatics*, 18:368–373, 2002.

[58] S.S. Fuller, D. Revere, P.F. Bugni, and G.M. Martin. A knowledgebase system to enhance scientific discovery: Telemakus. *Biomedical Digital Libraries*, 1(1):1–15, 2004.

[59] Michael Y. Galperin. The molecular biology database collection: 2008 update. *Nucleic Acids Research*, 36, Database issue:D2–D4, 2008.

[60] Floris Geerts, Anastasios Kementsietsidis, and Diego Milano. Mondrian: Annotating and querying databases through colors and blocks. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, pages 82–82.

[61] C.A. Goble, R. Stevens, G. Ng, S. Bechhofer, N.W. Paton, P.G. Baker, M. Peim, and A. Brass. Transparent access to multiple bioinformatics information sources. *IBM System Journal*, 40(2):532–552, 2001.

[62] Paul M.K. Gordon, Quang Trinh, and Christoph W. Sensen. Semantic web service provision: a realistic framework for bioinformatics programmers. *Bioinformatics*, 23(9):1178–1180, 2007.

[63] N. Guex, A. Diemand, and M.C Peitsch. Protein modelling for all. *Trends in Biochemical Sciences*, 24:364–367, 1999.

[64] Alon Halevy. Answer queries using views: A survey. *The VLDB Journal*, 10:270–294, 2001.

[65] Alon Halevy, Anand Rajaraman, and Joann Ordille. Data integration: The teenage years. *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, 10:9–16, 2006.

[66] Joachim Hammer and Markus Schneider. Going back to our database roots for managing genomic data. *OMICS: A Journal of Integrative Biology*, 7(1), 2003.

[67] Thomas Hernandez and Subbarao Kambhampati. Integration of biological sources: Current systems and challenges. *ACM SIGMOD Record*, 33(3):151–60, 2004.

[68] Berman HM, Henrick K, and Nakamura H. Announcing the worldwide protein data bank. *Nature Structural Biology*, 10(12):980, 2003.

[69] Pengyu Hong, Sheng Zhong, and Wing Hung Wong. Ubic2 -towards ubiquitous bioinformation computing: Data protocols, middleware, and web services for heterogeneous biological information integration and retrieval. *International Journal of Software Engineering and Knowledge Engineering*, 15(3):475–486, 2005.

[70] Barthelmes J., Ebeling C., Chang A., Schomburg I., and Schomburg D. "brenda, amenda and frenda: the enzyme information system in 2007. *Nucleic Acids Research*, 35, Database issue:D511–D514, 2007.

[71] Van Helden J., Naim A., Mancuso R., Eldridge M., Wernisch L., Gilbert D., and Wodak SJ. Representing and analysing molecular and cellular function using the computer. *Biol Chem.*, 381(9-10):921–35, 2000.

[72] Vaida Jakoniene and Patrick Lambrix. Ontology-based integration for bioinformatics. In *Proceeding of the 31th VLDB Conference*, Trondheim, Norway, 2005.

[73] Feng Ji, Ramez Elmasri, Yiming Zhang, B. Ritesh, and Zoe Raja. Incorporating concepts for bioingormatics data modeling into eer models. In *The 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2005)*, pages 189–192, Cairo, Egypt, January 3-6, 2005.

[74] Philip Jones, Richard G. Cote, Sang Yun Cho, Sebastian Klie, Lennart Martens, Antony F. Quinn, David Thorneycroft, and Henning Hermjakob.

Pride: new developments and new datasets. *Nucleic Acids Research*, 36, Database issue:D878–D883, 2008.

[75] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2673, 1998.

[76] P.D. Karp. Pathway databases: A case study in computational symbolic theories. *Science*, 293:2040–2044, 2001.

[77] C.M. Keet. Biological data and conceptual modelling methods. *Journal of Conceptual Modeling*, 29, 2003.

[78] Paul J. Kersey, Jorge Duarte, Allyson Williams, Youla Karavidopoulou, Ewan Birney, and Rolf Apweiler. The international protein index: An integrated database for proteomics experiments. *Proteomics*, 4(7):1985–1988, 2004.

[79] A.M. Kogelnik, M.T. Lott, M.D. Brown, S.B. Navathe, and D.C. Wallace. Mitomap: a human mitochondrial genome database. *Nucleic Acids Res.*, 24(1):177–179, 1996.

[80] Larkshmi Krishnamurthy, Joseph H. Nadeau, Gultekin Özsoyoglu, Z. Meral Özsoyoglu, Greg Schaeffer, Murat Tasan, and Wanhong Xu. Pathways database system: An integrated system for biological pathways. *Bioinformatics*, 19(8):930–937, 2003.

[81] Salwinski L, Miller CS, Smith AJ, Pettit FK, Bowie JU, and Eisenberg D. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32(Database issue):D449–51, 2004.

[82] Alberto Labarga, Franck Valentin, Mikael Anderson, and Rodrigo Lopez. Mapping pdb chains to uniprotkb entries. *Bioinformatcs*, Web Server issue:1–6, 2007.

[83] Maurizio Lenzerini. Data integration: A theoretical perspective. *PODS*, pages 233–246, 2002.

[84] Gardiner-Garden M. and Littlejohn TG. A comparison of microarray databases. *Briefings in Bioinformaitcs*, 2(1):143–58, 2001.

[85] Andrew C.R. Martin. Mapping pdb chains to uniprotkb entries. *Bioinformatcs*, 21(23):4297–4301, 2005.

[86] J. McEntyre. Linking up with entrez. *Trends Genet.*, 14(1):39–40, 1998.

[87] George Mihaila, Felix Naumann, Louiqa Raschid, and Maria Esther Vidal. A data model and query language to explore enhanced links and paths in life science sources. In *Eighth International Workshop on the Web and Databases (WebDB 2005)*, Baltimore, Maryland, June 2005.

[88] Ben Miled. Complex life science multidatabase queries. *Proceedings of the IEEE*, 90(11), 2002.

[89] P. Mitra, G. Wiederhold, and S. Decker. A scalable framework for the interoperation of information sources. In *The first Semantic Web Working Symposium, (SWWS 2001)*, pages 317–329, 2001.

[90] Peter Mork, Ron Shaker, and Peter Tarczy-Hornoch. The multiple roles of ontologies in the biomediator data integration system. In *DILS*, pages 96–104, 2005.

[91] Norman W. Paton, Shakeel A. Khan, Andrew Hayes, Fouzia Moussouni, Andy Brass, Karen Elibeck, Carole A. Goble, Simon J. Hubbard, and Stephen G. Oliver. Conceptual modelling of genomic information. *Bioinformatics*, 16(6):548–557, 2000.

[92] Eric Prudhommeaux. Case study: Federate for drug research, 2004. http://www.w3.org/2004/10/04-pharmaFederate/.

[93] S. Ram and W. Wei. Modeling the semantic of 3d protein structures. In *The 23rd International Conference on Conceptual Modeling, (ER), LNCS 3288*, pages 696–708, 2004.

[94] C. Riemer, A. ElSherbini, N. Stojanovic, S. Schwartz, P.B. Kwitkin, W. Miller, and R. Hardison. A database of experimental results on globin gene expression. *Genomics*, 53:325–337, 1998.

[95] Paolo Romano, Domenico Marra, and Luciano Milanesi. Web services and workflow management for biological resources. *BMC Bioinformatcs*, 6(Suppl 4):S24:1–9, 2005.

[96] Alan Ruttenberg, Tim Clark, William Bug, Matthias Samwald, Olivier Bodenreider, Helen Chen, Donald Doherty, Kerstin Forsberg, Yong Gao, Vipul Kashyap, June Kinoshita, Joanne Luciano, , M Scott Marshall, Chimezie Ogbuji, Jonathan Rees1, Susie Stephens, Gwendolyn T Wong, Elizabeth Wu, Davide Zaccagnini, Tonya Hongsermeier, Eric Neumann, and Ivan Herman andKei Hoi Cheung. Advancing translational research with the semantic web. *BMC Bioinformatcs*, 8(Suppl 3):S2:1–16, 2007.

[97] Kai-Uwe Sattler, Ingolf Geist, and Eike Schallehn. Concept-based querying in mediator systems. *The VLDB Journal*, 14(1):97–111, 2005.

[98] Barry Smith, Werner Ceusters, Bert Klagges, Jacob Koler, Anand Kumar, Jane Lomax, Chris Mungall, Fabian Neuhaus, Alan L Rector, and Cornelius Rosse. Relations in biomedical ontologies. 6:R46(2):166–180, 2005.

[99] Lincoln Stein. Creating a bioinformatics nation. *Nature*, 417:119–120, 2002.

[100] Sandeep Tata, Jignesh M. Patel, James S. Friedman, and Anand Swaroop. Declarative querying for biological sequences. In *ICDE*, page 87, 2006.

[101] Y. Tateno, N. Saitou, K. Okubo, H. Sugawara, and T. Gojobori. "ddbj in collaboration with mass-sequencing teams on annotation. *Nucleic Acids Research*, 33, Database issue:D25–D28, 2005.

[102] Thodoros Topaloglou, Susan B. Davidson, H. V. Jagadish, Victor M. Markowitz, Evan W. Steeg, and Mike Tyers. Biological data management: Research, practice and opportunities. In *VLDB*, pages 1233–1236, 2004.

[103] Silke Trissl, Kristian Rother, Heiko Muller, Thomas Steinke, Ina Koch, Robert Preissner, Cornelius Frommel, and Ulf Leser. Columnba: an integrated database of proteins, structures, and annotatons. *BMC Bioinformatics*, 6(81):1–11, 2005.

[104] J. Westbrook, Z. Feng, L. Chen, H. Yang, and H. Berman. The protein data bank and structural genomics. *Nucleic Acids Research*, 31(1):489–491, 2005.

[105] John D. Westbrook and Philip E. Bourne. Star/mmcif: An extensive ontology for macromolecular structure and beyond. *Bioinformatics*, 16(2):159–168, 2000.

[106] Xifeng Yan, Philip S. Yu, and Jiawei Han. Substructure similarity search in graph databases. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 766–777, New York, NY, USA, 2005. ACM.

[107] Xuan Zhang and Gagan Agrawal. A tool for supporting integration across multiple flatfile datasets. In *BIBE*, pages 141–148, 2006.

BIOGRAPHICAL STATEMENT

Feng Ji received his M.S. degree in Physical Chemistry from Nanjing University (P.R. China) in 1997, and his M.S. degree in Computer Science from University of Texas at Arlington in 2002. He is currently a Ph.D. student in the Department of Computer Science and Engineering at UT-Arlington. His current research areas focus on bioinformatics data modeling and integration using ontologies in mediator systems.