

Biological Database Modeling

For a listing of recent titles in the
Artech House Bioinformatics & Biomedical Imaging Series,
turn to the back of this book.

Biological Database Modeling

Jake Chen
Amandeep S. Sidhu

Editors



**ARTECH
HOUSE**

BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library.

ISBN-13: 978-1-59693-258-6

Cover design by Igor Valdman

© 2008 ARTECH HOUSE, INC.
685 Canton Street
Norwood, MA 02062

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

10 9 8 7 6 5 4 3 2 1

Contents

Preface	<i>xi</i>
Acknowledgments	<i>xv</i>
CHAPTER 1	
Introduction to Data Modeling	1
1.1 Generic Modern Markup Languages	1
1.2 Modeling Complex Data Structures	3
1.3 Data Modeling with General Markup Languages	3
1.4 Ontologies: Enriching Data with Text	4
1.5 Hyperlinks for Semantic Modeling	5
1.6 Evolving Subject Indexes	6
1.7 Languages	6
1.8 Views	7
1.9 Modeling Biological Data	7
References	8
CHAPTER 2	
Public Biological Databases for -Omics Studies in Medicine	9
2.1 Introduction	9
2.2 Public Databases in Medicine	10
2.3 Application of Public Bioinformatics Database in Medicine	11
2.3.1 Application of Genomic Database	11
2.3.2 Application of Proteomic Database	16
2.3.3 Application of the Metabolomics Database	18
2.3.4 Application of Pharmacogenomics Database	19
2.3.5 Application of Systemics Database	21
References	21
CHAPTER 3	
Modeling Biomedical Data	25
3.1 Introduction	25
3.2 Biological Concepts and EER Modeling	27
3.2.1 Sequence Ordering Concept	27
3.2.2 Input/Output Concept	29
3.2.3 Molecular Spatial Relationship Concept	30
3.3 Formal Definitions for EER Extensions	31
3.3.1 Ordered Relationships	31

3.3.2	Process Relationships	33
3.3.3	Molecular Spatial Relationships	34
3.4	Summary of New EER Notation	35
3.5	Semantic Data Models of the Molecular Biological System	35
3.5.1	The DNA/Gene Model	36
3.5.2	The Protein 3D Structure Model	36
3.5.3	The Molecular Interaction and Pathway Model	40
3.6	EER-to-Relational Mapping	41
3.6.1	Ordered Relationship Mapping	41
3.6.2	Process Relationship Mapping	42
3.6.3	Molecular Spatial Relationship Mapping	43
3.7	Introduction to Multilevel Modeling and Data Source Integration	45
3.8	Multilevel Concepts and EER Modeling	46
3.9	Conclusion	48
	References	49
CHAPTER 4		
	Fundamentals of Gene Ontology	51
4.1	Introduction to Gene Ontology	51
4.2	Construction of an Ontology	52
4.3	General Evolution of GO Structures and General Annotation Strategy of Assigning GO Terms to Genes	56
4.3.1	General Evolution of GO Structures	56
4.3.2	General Annotation Strategy of Assigning GO Terms to Genes	57
4.4	Applications of Gene Ontology in Biological and Medical Science	57
4.4.1	Application of Gene Ontology in Biological Science	57
4.4.2	Application of Gene Ontology in Medical Science	58
	References	60
CHAPTER 5		
	Protein Ontology	63
5.1	Introduction	63
5.2	What Is Protein Annotation?	64
5.3	Underlying Issues with Protein Annotation	64
5.3.1	Other Biomedical Ontologies	65
5.3.2	Protein Data Frameworks	66
5.3.3	Critical Analysis of Protein Data Frameworks	68
5.4	Developing Protein Ontology	68
5.5	Protein Ontology Framework	69
5.5.1	The ProteinOntology Concept	70
5.5.2	Generic Concepts in Protein Ontology	70
5.5.3	The ProteinComplex Concept	71
5.5.4	Entry Concept	71
5.5.5	Structure Concept	72
5.5.6	StructuralDomains Concept	72
5.5.7	FunctionalDomains Concept	73
5.5.8	ChemicalBonds Concept	74

5.5.9	Constraints Concept	74
5.5.10	Comparison with Protein Annotation Frameworks	75
5.6	Protein Ontology Instance Store	76
5.7	Strengths and Limitations of Protein Ontology	77
5.8	Summary	78
	References	78

CHAPTER 6

	Information Quality Management Challenges for High-Throughput Data	81
6.1	Motivation	81
6.2	The Experimental Context	84
6.2.1	Transcriptomics	86
6.2.2	Qualitative Proteomics	88
6.3	A Survey of Quality Issues	89
6.3.1	Variability and Experimental Design	89
6.3.2	Analysis of Quality Issues and Techniques	91
6.3.3	Specificity of Techniques and Generality of Dimensions	93
6.3.4	Beyond Data Generation: Annotation and Presentation	94
6.4	Current Approaches to Quality	96
6.4.1	Modeling, Collection, and Use of Provenance Metadata	96
6.4.2	Creating Controlled Vocabularies and Ontologies	97
6.5	Conclusions	98
	Acknowledgments	98
	References	98

CHAPTER 7

	Data Management for Fungal Genomics: An Experience Report	103
7.1	Introduction	103
7.2	Materials Tracking Database	109
7.3	Annotation Database	110
7.4	Microarray Database	111
7.5	Target Curation Database	111
7.6	Discussion	112
7.6.1	Issue of Data and Metadata Capture	113
7.7	Conclusion	116
	Acknowledgments	116
	References	116

CHAPTER 8

	Microarray Data Management: An Enterprise Information Approach	119
8.1	Introduction	119
8.2	Microarray Data Standardization	122
8.2.1	Gene Ontologies	123
8.2.2	Microarray Ontologies	125
8.2.3	Minimum Information About a Microarray Experiment	125
8.3	Database Management Systems	126
8.3.1	Relational Data Model	127

8.3.2	Object-Oriented Data Model	128
8.3.3	Object-Relational Data Model	131
8.4	Microarray Data Storage and Exchange	131
8.4.1	Microarray Repository	133
8.4.2	Microarray Data Warehouses and Datamarts	133
8.4.3	Microarray Data Federations	134
8.4.4	Enterprise Microarray Databases and M-KM	135
8.5	Challenges and Considerations	136
8.6	Conclusions	138
	Acknowledgments	138
	References	139
CHAPTER 9		
	Data Management in Expression-Based Proteomics	143
9.1	Background	143
9.2	Proteomics Data Management Approaches	147
9.3	Data Standards in Mass Spectrometry Based Proteomics Studies	149
9.4	Public Repositories for Mass Spectrometry Data	152
9.5	Proteomics Data Management Tools	154
9.6	Expression Proteomics in the Context of Systems Biology Studies	155
9.7	Protein Annotation Databases	159
9.8	Conclusions	159
	References	160
CHAPTER 10		
	Model-Driven Drug Discovery: Principles and Practices	163
10.1	Introduction	163
10.2	Model Abstraction	165
10.2.1	Evolution of Models	166
10.3	Target Identification	168
10.3.1	Sequence-to-Function Models	170
10.3.2	Sequence Alignments and Phylogenetic Trees	170
10.3.3	Structure-to-Function Models	172
10.3.4	Systems-Based Approaches	173
10.3.5	Target Validation	176
10.4	Lead Identification	177
10.4.1	Target Structure-Based Design	177
10.4.2	Ligand-Based Models	179
10.5	Lead to Drug Phase	182
10.5.1	Predicting Drug-Likeness	182
10.5.2	ADMET Properties	182
10.6	Future Perspectives	183
	Acknowledgments	184
	References	184

CHAPTER 11

Information Management and Interaction in High-Throughput Screening for Drug Discovery	189
11.1 Introduction	189
11.2 Prior Research	191
11.3 Overview of Antimalarial Drug Discovery	192
11.4 Overview of the Proposed Solution and System Architecture	193
11.5 HTS Data Processing	194
11.5.1 Introduction to HTS	194
11.5.2 Example of HTS for Antimalarial Drug Screening	195
11.6 Data Modeling	199
11.6.1 The Database Design	202
11.7 User Interface	204
11.8 Conclusions	206
Acknowledgments	207
References	207
Selected Bibliography	208
About the Authors	209
Index	215

Preface

Database management systems (DBMS) are designed to manage large and complex data sets. In the past several decades, advances in computing hardware and software and the need to handle rapidly accumulating data archived in digital media have led to significant progress in DBMS research and development. DBMS have grown from simple software programs that handled flat files on mainframe computers, which were prohibitively expensive to all but a few prestigious institutions, into today's popular form of specialized software platforms underpinning wide ranges of tasks, which include business transactions, Web searches, inventory management, financial forecasts, multimedia development, mobile networks, pervasive computing, and scientific knowledge discovery. Technologies of DBMS have also become increasingly sophisticated, diverging from generic relational DBMS into object-relational DBMS, object-oriented DBMS, in-memory DBMS, semantic Webs data store, and specialized scientific DBMS. Given the sustained exponential data growth rate brought forth by continued adoption of computing in major industries and new inventions of personal digital devices, one can safely predict that DBMS development will continue to thrive in the next millennium.

In this book, we want to share with our readers some fresh research perspectives of post-genome biology data management, a fast-growing area at the intersection of life sciences and scientific DBMS domains. Efficient experimental techniques, primarily DNA sequencing, microarrays, protein mass spectrometers, and nanotechnology instruments, have been riding the wave of the digital revolution in the recent 20 years, leading to an influx of high-throughput biological data. This information overload in biology has created new post-genome biology studies such as genomics, functional genomics, proteomics, and metabolomics—collectively known as “omics” sciences in biology. While most experimental biologists are still making the transition from one-gene-at-a-time type of studies to the high-throughput data analysis mindset, many leaders of the field have already begun exploring new research and industrial application opportunities. For example, managing and interpreting massive omics data prelude ultimate systems biology studies, in which one may analyze disparate forms of biological data and uncover coordinated functions of the underlying biological systems at the molecular and cellular signalling network level. On the practical side, understanding diverse intricate interplays between environmental stimuli and genetic predisposition through omics evidence can help pharmaceutical scientists design drugs that target human proteins with high therapeutic values and low toxicological profiles. With data management tools to handle terabytes of omics data already released in the public domain, the promise of post-genome biology looms large.

Compared with data from general business application domains, omics data has many unique characteristics that make them challenging to manage. Examples of these data management challenges are:

1. Omics data tends to have more complex and more fast-evolving data structures than business data. Biological data representation often depends on scientific application scenarios. For example, biological sequences such as DNA and proteins can be either represented as simple character strings or connected nodes in three-dimensional spatial vectors. Data representation is an essential first step.
2. Omics data is more likely to come from more heterogeneously distributed locations than business data. To study systems biology, a bioinformatics researcher may routinely download genome data from the Genome Database Center at the University of California, Santa Cruz, collect literature abstracts from the PubMed database at the National Library of Medicine in Maryland, collect proteome information from the Swiss-Prot database in Switzerland, and collect pathway data from the KEGG database in Japan. Data integration has to be carefully planned and executed.
3. Omics data tends to reflect the general features of scientific experimental data: high-volume, noisy, formatted inconsistently, incomplete, and often semantically incompatible with one another. In contrast, data collected from business transactions tends to contain far fewer errors, is often more accurate, and shows more consistencies in data formats/coverage. Meticulous data preprocessing before knowledge discovery are required.
4. Omics data also lags behind business data in standard development. For example, Gene Ontology (GO) as a standard to control vocabularies for genes was not around until a decade ago, whereas standards such as industrial product categories have been around for decades. The ontology standards and naming standards for pathway biology are still under development. This makes it difficult to perform mega collaboration, in which cross-validation of results and knowledge sharing are both essential.

Despite all the challenges, modeling and managing biological data represent significant discovery opportunities in the next several decades. The human genome data bears the ultimate solutions of expanding the several thousand traditional molecular drug targets into tens of thousands genome drug targets; molecular profiling information, based on individuals using either the microarrays or the proteomics platform, promises new types of molecular diagnostics and personalized medicine. As new applications of massive biological data emerge, there will be an increasing need to address data management research issues in biology.

In this compiled volume, we present to our readers a comprehensive view of how to model the structure and semantics of biological data from public literature databases, high-throughput genomics, gene expression profiling, proteomics, and chemical compound screening projects. The idea of compiling this book, which we found to be unique, stems from the editors' past independent work in bioinformatics and biological data management. While topics in this area are diverse and interdisciplinary, we focused on a theme for this book—that is, how to model and manage

omics biological data in databases. By promoting this theme for the past decade among ourselves and the contributing authors of this book, we have contributed to solving complex biological problems and taking biological database management problems to the next level. We hope our readers can extract similar insights by using this book as a reference for future related activities.

There are 11 chapters presented in this book. Individual chapters have been written by selected accomplished research teams active in the research of respective topics. Each chapter covers an important aspect of the fast-growing topic of biological database modeling concepts. Each chapter also addresses its topic with varying degrees of balance between computational data modeling theories and real-world applications.

In Chapters 1 through 5, we introduce basic biological database concepts and general data representation practices essential to post-genome biology. First, biological data management concepts are introduced (Chapter 1) and major public database efforts in omics and systems biology studies are summarized (Chapter 2). Then, biomedical data modeling techniques are introduced (Chapter 3). Next, Gene Ontology as an established basic set of controlled vocabulary in genome database annotations is described (Chapter 4). Finally, the latest research on protein ontology and the use of related semantic webs technologies are presented to enable readers to make the connection between emerging biological data collection and integration trends (Chapter 5).

In Chapters 6 through 9, we examine in detail how to develop data management techniques to process and analyze high-throughput biological data through case studies. First, quality control techniques to reduce variations during experimental data collection steps are described (Chapter 6). Then, biological sequence management experience for a fungi genomics project is discussed (Chapter 7). Next, data management and data integration methods for microarray-based functional genomics studies are investigated (Chapter 8). Finally, data management challenges and opportunities for mass spectrometry based expression proteomics are presented (Chapter 9).

In Chapters 10 and 11, we delve into the practical aspect, demonstrating how to apply biological data management for drug discoveries. First, fundamental drug discovery concepts based on macromolecular structural modeling are introduced (Chapter 10); then, a data management software system that implements high-throughput drug compound screenings is discussed (Chapter 11) to conclude the book.

We hope this book will become a useful resource for bioinformatics graduate students, researchers, and practitioners interested in managing post-genome biological data. By studying the techniques and software applications described in this book, we hope that bioinformatics students will use the book material as a guide to acquire basic concepts and theories of post-genome biological data management, bioinformatics practitioners will find valuable lessons for building future similar biological data management systems, and researchers will find rewarding research data management questions to address in the years to come.

Acknowledgments

We wish to thank all of the authors for sharing their insightful knowledge and making excellent contributions to this book based on their active research in biological data management. This book would not have been completed without the tremendous efforts, held to the highest standards, of all the authors, each of whom spent numerous hours over many drafts in preparing, collating, and revising their writings over the past 2 years. During the publishing process, many colleagues also helped and they deserve our whole-hearted appreciation. They are: David Wong from Indiana University, who provided legal advice for the contract agreement; Dr. Zongmin Ma from Northwestern University of China, who provided assistance in the initial conceptualization and execution of the book publishing process; Susan Lagerstrom-Fife from Springer Science+Business Media, Inc., who guided us from a publisher's perspective while we explored various publication options; Wayne Yuhasz from Artech House Publishers, whose persistence and dedication to timely assistance finally won us over in making our book part of the Artech House Bioinformatics & Biomedical Imaging Series; and Barbara Lovenvirth from Artech House Publishers, who assisted us throughout the final publication process.

We also want to thank the general support from our home institutions: Indiana University School of Informatics at Indianapolis, Indiana; Purdue University School of Science, Department of Computer and Information Sciences at Indianapolis, Indiana; and Curtin University of Technology, Perth, Australia.

Last, but not least, Jake Chen wishes to thank his family, including his wife, Mabel Liu, for her unbounded love and support in assuming additional responsibilities so that the book project could be completed.

*Jake Chen
Indianapolis, Indiana
Amandeep S. Sidhu
Perth, Australia
Editors
October 2007*

CHAPTER 1

Introduction to Data Modeling

Amandeep S. Sidhu and Jake Chen

Scientific data is often scattered among heterogeneous data repositories. Exploring data across multiple data repositories requires the ability to understand and correlate their structures (schemas). Such correlations need to address the diversity of views of the scientific domain represented by different data repositories as well as the diversity of data modeling languages used for expressing these views. In this chapter, we introduce the concepts of *data modeling* and discuss its application to *biological databases*.

1.1 Generic Modern Markup Languages

Modern markup languages, such as Standard Generalized Markup Language (SGML) [1] and eXtensible Markup Language (XML) [2], which were initially conceived for modeling texts, are now receiving increasing attention as formalisms for data and knowledge modeling. XML is currently establishing itself as a successor of HyperText Markup Language (HTML) for a better modeling of texts as well as of other kinds of data. There are several reasons for this evolution. Even though multiple databases may cover the same data, their focus might be different. Modern markup languages such as SGML and XML are generic in that:

- They serve to specify the semantic structure, not the layout, of documents or data items.
- They make it possible to freely specify application-dependent document or data structures.

In the following, the term “data” refers also, but not exclusively, to text data. Thus, a data item may consist of: (1) text only (such data items are also known as human-readable documents); (2) nontext only (such data items are also known as data-oriented documents); or (3) both (such data items are also known as mixed-model documents). In the terminology of generic markup languages, data items are called documents. In the following, the term “data item” is used in lieu of “document” for stressing that not only (structured) texts are meant, but more generally (structured) data of any kind.

Widespread specific markup languages such as PostScript or Rich Text Format (RTF), whose conceptual roots go back to the 1970s, serve to specify the layout of data items. Here, layout is not exclusively meant as the appearance of a data item when printed on paper, but more generally as any kind of presentation of a data item to human perception. Examples of such an extended notion of layout include the formats of data items as they are displayed on a terminal screen, rendered in the script on an output device, or presented by any other means on any device.

The family of generic markup languages started in the late 1980s with the conception of its first specimen, SGML. The purpose of a generic markup language is to specify the semantic—or logical—structure of data items, not their layout. In the following, the term “presentation” is reserved to refer to the layout of a data item in the extended sense above, while the term “representation” refers to how semantics is conveyed through structural elements of the underlying data modeling formalism.

The distinction between layout and structure is important, for a layout format is device or system dependent, whereas a semantic structure should not be. It is desirable that the semantic structure of data items be specified independently of any layout. This ensures both:

- Independence of data modeling from data usage;
- Independence of data modeling from presentation devices.

The first property, data independence from usage, is important because data is rarely used in a single manner only. The second property, data independence from presentation devices, is important for several reasons. To begin with, different kinds of presentation devices require different layouts. For example, a structurally complex data item is likely not to be displayed using identical layouts on standard size screens and on small screens like those of cellular phones. Also, such devices are likely to become technically obsolete sooner than data. Moreover, a presentation format does not necessarily fully convey data semantics. For instance, it is common practice to rely on printed text layout for conveying semantic structure when using text processing systems or the markup language HTML. This practice often leads to semantic losses, especially when files are transferred from one text processing system to another, because the layout of the one system cannot always be faithfully mapped into that of the other system.

In order to specify layouts for classes of documents specified in a generic markup language, so-called style-sheet languages are used in addition. These languages basically allow the definition of layouts for those structural elements specified with the markup language. Such definitions do not have to be unique, thus ensuring the desired independence of the data from their presentations in various contexts.

Generic markup languages (like the XML family of languages) do not impose any predefined structure, nor any predefined names for the structural elements occurring in data items. Structure and names can be freely chosen, hence the denomination of generic markup language. Thus, using generic markup languages it is possible to faithfully model the structure of data items needed in applications and to name the structural elements of a chosen structure in a way that is natural in the application context.

1.2 Modeling Complex Data Structures

Complex structures are essential for they are ubiquitous in data modeling—from records in programming languages to objects in programming languages, artificial intelligence, software engineering, databases, and logics for knowledge representation. The formalisms provided by generic markup languages make it possible to specify complex structures. Therefore, they are much richer than the data model of relational databases. They are also richer than the data models of current object database systems, because they allow optional elements. Optional elements are very appealing in databases, for they make it possible to express exceptions, which often occur in practical database applications.

The term “semistructured data” has been coined for emphasizing the possibility of such exceptions in the framework of structure-conveying data. It is under the denomination of semistructured data that most database research on using markup languages for data modeling is currently pursued.

1.3 Data Modeling with General Markup Languages

Data modeling with generic markup languages is an interdisciplinary area of research at the crossing of four traditionally distinct fields of research:

- Databases;
- Artificial intelligence;
- Information retrieval;
- Document processing.

This convergence is interesting, because each field brings its own focus, methods, and philosophy.

From databases, the research area of data modeling with generic markup languages gains an interest for declarative query languages, of which SQL is the most well-known example. Declarativeness is a loosely defined notion to be understood here as meaning that the users of such query languages do not have to be aware of the computation strategy, of the internal organization of the data in memory, nor—or as little as possible—of termination issues and of efficiency. Indeed, queries expressed in query languages are automatically optimized. This makes query optimization possible. Query optimization guarantees a predictable “average efficiency,” which is one of the appreciated features of database systems. Also from databases, the area of data modeling with generic markup languages inherits its interest for data structures, ensuring an efficient storage, retrieval, and updating of very large data sets. Conversely, database research itself is enriched by the attention to text data, to data accessible from the Web, and to richer data models allowing for exceptions.

From artificial intelligence, the research area gains data and knowledge modeling methods that go far beyond the relational or object database models. Artificial intelligence approaches to knowledge representation have always been driven by natural language applications, where extremely rich and complex semantics are

encountered. The need for software interoperability and data interchange in Web-based applications such as electronic commerce, health care management, and computational biology (also called bioinformatics) nowadays has led researchers, as well as practitioners, to express advanced artificial intelligence knowledge representation formalisms such as description logics by relying upon the generic markup language XML.

From information retrieval, the research area can learn how to automatically “grasp” knowledge from the content of—in general, large—texts. The field of information retrieval itself gains from the consideration of structured texts, which up until recently have almost never been considered by the information retrieval community.

The contributions of the document processing field to the research area of data modeling with generic markup languages are indeed the generic markup languages and hyperlink models. Document processing itself might benefit from the interdisciplinary approach to data modeling with methods for declarative query answering and for an efficient storage.

The focus of the research activities described here is on data modeling as it has emerged from combining techniques and ideas from both databases and artificial intelligence. As a consequence, issues more specific to one of the fields of databases, information retrieval, and document processing are likely to receive less attention in this chapter.

1.4 Ontologies: Enriching Data with Text

Enriching standard data—like the numerical and string data of classical managerial databases—with more informative texts is an old issue in database research and in applications known as data dictionaries. Data dictionaries are basically agreed-upon vocabularies for an application or a class of applications and often taxonomies (i.e., classifications of terms).

Recently, the issue has gained new attention and has been enhanced with artificial intelligence knowledge modeling techniques, leading to so-called ontologies. An ontology provides a vocabulary whose terms are precisely defined by texts such as dictionary entries or encyclopedia entries. Further, ontology also defines semantic relationships between terms using formal modeling techniques, in general taken from logic-based specification formalisms such as description logics. Thus, ontology starts from precisely defined basic concepts, building up complex concepts by relying on relationships that are precisely defined as well. These relationships permit the construction of taxonomies, but also of richer structures.

With the advent of the World Wide Web and data interchange intensive applications, ontologies are becoming a key issue. They are used for ensuring software interoperability and data sharing [3]. In spite of their practical relevance, data dictionaries and ontologies have up until recently not received as much attention from the database research community as database practitioners might have wished. This discrepancy clearly reflects the fact that, up until now, textual data was not a central interest in database research and database system development, nor was the modeling of complex semantic relationships as in artificial intelligence knowledge model-

ing approaches such as description logics. This seems to be changing now. Arguably, textual data and the modeling of complex semantic relationships are gaining in importance within database research.

1.5 Hyperlinks for Semantic Modeling

A hyperlink model defines hyperlink types based on semantic relationships between data items. A browser model defines the behavior of such hyperlinks. The proposed distinction is analogous to the distinction between a generic markup language, whose purpose is to model the logical structure of data items, and a style-sheet language, whose purpose is to specify the layout for a given logical structure.

A hyperlink model might define the following, semantically characterized, hyperlink types: (1) hyperlink to new information; (2) hyperlink to alternative descriptions of the current information; and (3) hyperlink to supplementary information. A browser model could then specify that these three types of hyperlinks behave in the three ways described above. Different browser models may specify the behavior differently.

It is interesting to investigate different browser models for the same hyperlink model. There are two reasons. First, in the future many data items are likely to be browsed using not only standard size screens, but also mini-screens like those of cellular phones or new paper-like electronic output devices like electronic paper (e-paper), with which browsing will most likely take new forms. Second, if hyperlinks are to be used for expressing semantic dependencies, which is taken here as a working assumption, then necessarily browsing along such hyperlinks will not be uniquely definable. Thus, the distinction between hyperlink model and browser model proposed here contributes to both independence of data modeling from data usage and independence of data modeling from presentation devices, which, as pointed out in Section 1.1, are key issues in data modeling.

A wide class of modeling problems investigated in the areas of artificial intelligence and knowledge representation boils down to defining so-called ontologies. An ontology is a set of concepts with a set of relationships between concepts. The concepts represent the basic terms of an application domain, and the meaning of these basic terms must be precisely specified.

The relationships between concepts are domain independent. A very common example is the generalization relationship: it relates a more specific concept with a more general concept, such as rabbit with rodent, rodent with mammal, mammal with animal, car with vehicle, or house with building. Generalization hierarchies, so-called taxonomies, can be found in virtually every application domain. Somewhat less obvious is the fact that there are many other domain-independent relationships, some of them hierarchy-forming. Examples include part-whole relationships such as component-object (finger/hand, hand/arm), member-collection (person/family, family/clan), and agent-result (programmer/program, program/output) relationships, as well as order relationships and similarity relationships. A set of such relationships can represent much of the semantics of a specific domain in a completely domain-independent way.

It is suggestive to map ontologies to hypertext systems, with data items representing concepts and hyperlinks representing relationships between concepts. In this way the hyperlinks are no longer considered as mere contributors to the navigation infrastructure for a browser, but as constructs of formalism for modeling the semantics of data.

1.6 Evolving Subject Indexes

Evolution was first identified in biology, but it is one of the fundamental principles pervading reality. Biology studies a continuously replenished pool of organisms, classified into species according to certain characteristics which some organisms have in common. These characteristics may change over time, requiring the definition of what constitutes a given species to be time dependent and making possible the emergence of subgroups of organisms of the same species, which may start out as races or subspecies and eventually split off to become species of their own.

At this level of generality the issue is hard to grasp from the point of view of knowledge modeling. In most cases it is not even clear which features adequately describe the members of a “pool” and how such features contribute to the classification inside the pool. So let us focus the issue to cases where there is a pool of semi-structured data items that model members of some pool in reality, and some parts of these data items correspond to classification features. For example, such a data item might be a research article with classification features including an author and a date, a list of keywords, and a list of cited articles. It does not matter whether the keywords are provided by the author or automatically extracted from the article using information retrieval methods. Given the classification features, it is possible to establish relationships between data items. In the example domain of research articles, straightforward relationships would be *written by the same author*, *written before*, and *cited by*; and somewhat less straightforward relationships would be *similar subject* and *taken up and advanced by*.

Appropriate numeric measures of the density of such relationships then allow the identification of data items that are *condensation kernels* for classes, and measures of the distance from those condensation kernels might define the boundaries of the classes. Taking into account the relationships along the temporal dimension, one can distinguish condensation kernels at different times and identify metamorphoses of classes [4]. This sketched approach differs from ontologies in several important ways: the condensations kernels and classes cannot be predefined, and membership of data items in classes is fuzzy and time dependent. The research issue would be to integrate numerical and fuzzy notions into the framework of semi-structured data.

1.7 Languages

There is intense activity on query languages for XML-inspired database query languages [5, 6]. Recently, XQuery [7] emerged as the query language of choice for XML. Because XQuery relies upon XPath [8], XQuery is navigational. It is possible to express in XPath/XQuery queries that express redundant (e.g., back and forth)

traversals through an XML document tree. Approaches inspired by or related to logic programming have been proposed, which aim at a more declarative—in the sense of less navigational—query-answering for XML and semi-structured data. One might call such approaches *positional* as opposed to *navigational*. This seems to be a very promising direction of research.

1.8 Views

Views are an essential feature of databases, for they ensure the conceptual independence needed in most applications between the primary data stored in the database and various interpretations, such as partial look-ups, of the stored data. Most applications based on databases rely on views. Views also make sense in texts, especially in texts with complex structures and/or contents.

Having different texts for different purposes would present several drawbacks. First, this would induce redundancies. Second, because of these redundancies, the approach would be error prone. Third, the consistency between the different texts giving complementary views of the same content would be difficult to maintain, which is yet another possible source of errors. For these reasons, it is desirable to model a notion of view while specifying the semantics of the considered documents.

1.9 Modeling Biological Data

Modern biology, particularly genomic research, is data and computation intensive. In biology in general and in genomic research in particular, it is a common practice nowadays to build databases of biological data. Most biological databases are—freely or not—accessible through the Web.

From the viewpoint of data modeling, especially of data modeling with markup languages, biological data and biological databases are interesting for several reasons:

- Biological data is subject to both general building laws, the discovery of which is a primary objective of biology, and exceptions. (The admittance of exceptions distinguishes modern markup languages from traditional data modeling formalisms.)
- Biological databases are based upon a multitude of data schemes. For most types of biological data there are no generally accepted data models or ontologies. (The resulting irregularities in structure are another form of exceptions and thus a case for modern markup languages.)
- Most biological databases contain data items that are enriched with texts. Typically such texts explain assumptions made in building up a data item. (Modern markup languages were designed for text in the first place.)
- Sophisticated querying of biological databases is an essential task in searching for laws governing biological data and processes. The querying has to take into account the irregularities because of exceptions, different data models, and enrichments with text.

- Generic markup languages, especially XML, are increasingly being used for modeling biological data.

Note also that most biological databases are, at least potentially, very large. For this reason, biological databases are also an interesting application from the viewpoint of (conventional) database system research and development.

Unfortunately, biological data modeling is rather difficult to understand for most computer scientists. These databases and the research issues they raise are not widely known outside computational biology. This is unfortunate because it prevents a fruitful cross-fertilization between application-driven computer science research, as mostly practiced by biologists and computational biologists, and method-driven computer science research, as practiced by computer science generalists.

An interesting research issue is to investigate, from the general computer science viewpoint, which are the essential aspects of biological data modeling. In this respect, approaches based on description logics seem especially promising. A further interesting research issue is to investigate whether specific querying methods are needed for biological databases.

References

- [1] van Herwijnen, E., *Practical SGML*, Boston, MA: Kluwer, 1994.
- [2] W3C-XML, "Extensible Markup Language (XML) 1.0," in *W3C Recommendation 16 August 2006, edited in place 29 September 2006*, T. Bray et al., (eds.), 4th ed., World Wide Web Consortium, 2006.
- [3] Smith, B., "Ontology," in *Blackwell Guide to the Philosophy of Computing and Information*, L. Floridi, (ed.), Oxford, U.K.: Blackwell, 2003, pp. 155–166.
- [4] Conrad, S., et al., "Evolving Logical Specification in Information Systems," in *Logics for Databases and Information Systems*, J. Chomicki and G. Saake, (eds.), Boston, MA: Kluwer, 1998.
- [5] Abiteboul, S., P. Buneman, and D. Suciu, *Data on the Web: From Relations to Semistructured Data and XML*, San Mateo, CA: Morgan Kaufmann, 2000.
- [6] Fernandez, M., et al., "A Query Language for a Web-Site Management System," *SIGMOD Record*, Vol. 26, 1997, pp. 4–11.
- [7] W3C-XQuery, "XQuery 1.0: An XML Query Language," in *W3C Proposed Recommendation 21 November 2006*, S. Boag et al., (eds.), World Wide Web Consortium, 2006.
- [8] W3C-XPath, "XML Path Language (XPath) Version 1.0," in *W3C Recommendation 16 November 1999*, J. Clark and S. DeRose, (eds.), World Wide Web Consortium, 1999.