THE UNIVERSITY OF KANSAS
PALEONTOLOGICAL CONTRIBUTIONS

August 1996 Number 8

PALEOBANK, A RELATIONAL DATABASE FOR INVERTEBRATE PALEONTOLOGY: THE DATA MODEL

Jill W. Krebs,[1] Roger L. Kaesler,[1,2,3] Elizabeth A. Brosius,[1,4] Douglas L. Miller,[5] and Yi-Maw Chang[1]

[1]Paleontological Institute, The University of Kansas, 121 Lindley Hall, Lawrence, Kansas 66045-2911; e-mail: Kaesler@KUHUB.CC.UKANS.EDU; [2]Department of Geology, The University of Kansas; [3]Division of Invertebrate Paleontology, Natural History Museum, The University of Kansas; [4]Kansas Geological Survey, The University of Kansas; [5]Information Technology Services, The University of Kansas Computer Center

Abstract.—As the science of paleontology enters a new phase in data management, interest in electronic databases burgeons. The proper design and management of a complex database necessitates the preparation of a logical data model. PaleoBank, a database that is an extension of the long-standing *Treatise on Invertebrate Paleontology*, is also intended as a research tool for paleontologists who are not involved in the *Treatise* project. The logical data model for PaleoBank, which is of the entity-relationship type, is presented in detail in electronic format elsewhere. Several universal resource locators (URLs) are given below that refer to relevant electronic documents. The data model comprises four entity-relationship diagrams, entity documentation in both complete and abbreviated forms, a complete description of the relationships in the data model, and details of the relational database design.

INTRODUCTION

PaleoBank is an electronic, relational database that is being developed as an extension of the *Treatise on Invertebrate Paleontology*, partly as a means of moviñg the *Treatise* project into the new century and partly as a means of helping paleontologists manage the vast amounts of information they have accumulated about the fossil record. In publishing this brief description of the data model that underpins the PaleoBank database and in presenting electronically a fully detailed version of the data model, we share with other biologists, geologists, and especially paleontologists both the data model and the perspective we have gained in building it.

Data management is a hot topic. The geological and biological literature is replete with references to databases. These references range from vague mention of the fact that data exist in some form to detailed descriptions of formal, electronically accessible relational databases. Electronic databases themselves range in scope from multiuser systems of the sort that facilitate the work of museums to highly specialized databases that an individual scientist might use in research.

All this points to the fact that our science has entered a phase in which the management of information is taking on new importance. As a result of billions of dollars invested in research since the end of World War II, we now know a great deal about the earth and how it works. Unfortunately, much of this information is not readily accessible, and paleontologists must find new ways to make the data available to others and to use it more effectively—ways to sort, examine, manipulate, and interpret it. Collecting information into electronic databases provides the potential to do all these things more flexibly and with less expenditure of time than was formerly possible. Data in electronic databases can be shared easily and quickly. They can be updated often, even daily or hourly. They can be collected in quantity yet remain accessible. Finally, standards for the data can be developed centrally and the quality of data can be controlled centrally, thereby largely eliminating both redundancy and inconsistency.

It is in this context that a data model becomes especially important. As we developed PaleoBank and delved into the intricacies of the interactions between the data and the programming, we soon realized the folly of attempting to develop a comprehensive paleontological database without first devoting the time and effort necessary to develop an equally complex and thorough data model. To be of value to a variety of users, a paleontological database must contain a great many different kinds of information. Such databases tend to be large and complex, often encompassing mountains of data that interact in many, sometimes unexpected ways with still other mountains of data. Moreover, databases of this kind must be designed so as to be long-lived and independent of the electronic medium in which they first appear. A detailed data model can be translated into any medium, platform, or software, and it lays the essential groundwork for effective programming. If the data model is well designed and carefully thought out, it becomes both a firm guide and a flexible plan that can be changed easily to meet new challenges and evolving contingencies.

For a given system of data, an entity-relationship data model identifies and defines entities—real-world concepts and objects; and it describes relationships—all associations, both implicit and explicit, among entities. The model diagrams and describes the conceptual framework of the data system, and it restricts and controls the input and output of data. Perhaps of greater importance, the process of developing the data model reveals illogical, woolly thinking and misrepresentations of the data. In short, the data model refines all aspects of the data and our understanding of them.

PaleoBank has the potential to house the world's largest databases on invertebrate fossils. The data model is intended to ensure that as PaleoBank grows it does so in an orderly fashion. Beyond that, we want to make PaleoBank a research tool that any paleontologist can use to capture new information as well as a means of disseminating that information, when it is appropriate to do so, to the scientific community.

PaleoBank, the Paleontological Institute, and the Treatise.—Most paleontological databases that are available either commercially or as freeware consist of compilations of data that are ready to be sorted and searched to answer specific questions or to prepare reports for special purposes. PaleoBank is closely linked to the longstanding *Treatise on Invertebrate Paleontology*, but it would fail utterly if it were to present only an electronic retrofitting of the sometimes outmoded information from published volumes of the *Treatise*. Instead, the first phase of PaleoBank focuses on the means of capturing new or revised information from specialists. Only when up-to-date information has been captured will the database be distributed to the community of users.

PaleoBank incorporates information about a number of important aspects of paleontology. These include taxonomy, morphology, stratigraphy, paleoecology, biogeography, plate tectonics, and bibliography. All of these categories of information, except plate tectonics, have been traditionally incorporated in the *Treatise on Invertebrate Paleontology*, although paleoecology has typically played only a minor role in most volumes. Incorporating such a wide variety of information into our entity-relationship data model has made us increasingly aware of the complexity of biological and paleontological systematics and taxonomy. Issues we have addressed include but are not limited to the following questions. What are the principal entities to be incorporated into the data model? What are the attributes of each entity? How are the real-world objects and the concepts that are represented in the data model interrelated in the real world? How can those relationships best be represented in detail?

The concepts involved in animal taxonomy and the interrelationships that exist among the various taxonomic concepts encountered in invertebrate paleontology are among the most complexly interwoven ideas in all of human scholarship. These include the nomenclatorial subtleties that are involved in synonymies; the concepts of *nomen novum, nomen translatum,* and *nomen nudum;* and movement of a species from one genus to another, to mention only a few. All of these complex ideas, of course, are governed by the *International Code of Zoological Nomenclature,* a set of quasilegal rules and suggestions that changes from time to time, sometimes with unsettling effects (Ride *et al.,* 1985).

To design a data model that accurately represents the complex interrelationships inherent in taxonomy has been a challenging task but an essential one, since we expect PaleoBank to facilitate the preparation of future volumes of the *Treatise on Invertebrate Paleontology*. It will do this by enabling the coordinating authors to draw upon information that is stored in electronic form and that can be sorted and queried in any number of ways because of the relational structure of the database. Moreover, PaleoBank will prevent many inadvertent taxonomic errors because of the fine sieve we've created in the data modeling of the taxonomic concepts. The report-writing capability of PaleoBank will allow preparation of *Treatise* manuscript, of manuscript for other kinds of systematics and taxonomic publications, and of extensive lists of references.

WHAT IS A DATA MODEL?

Relational-database theory.—To understand data models one must understand something about relational databases. Throughout his detailed and thorough introduction to database systems, C. J. Date acknowledged the pioneering work of E. F. Codd in developing relational-database theory in abstract, mathematical form (Date, 1995, see especially p. 56–57 and 100–101). As Date pointed out, both set theory and predicate logic underlie the relational model (Date, 1995, p. 56). Since Codd's work in the field, which took form with his 1969 and 1970 papers, relational databases have been based on solid principles and have been characterized by the application of a rigorous, logically consistent methodology (Codd, 1969, 1970).

Relational databases consist of records that are assembled from separate elements of data distributed throughout a system of related files. For example, the PaleoBank record for a single taxon is assembled from data elements stored in files called AUTHOR, PUBLICATION, and BASIC TAXON. Until the individual taxon record is called up for review or report writing, it is stored in these separate, related files that are linked by computer code. These features give relational databases a great advantage over the older, storage-intensive flat-file databases. When a relational file is updated—for instance, an AUTHOR's name is changed from John D. Smith to J. D. Smith—all of its related files will automatically reflect the changed data. In a flat-file database each record must be revised individually.

What is a logical data model?—A logical data model is, in effect, the blueprint from which a relational database is built. Because of its rigorous logical foundation, a logical data model is extremely stable yet flexible. It is nonredundant, streamlined, and efficient; and a properly prepared and documented data model can have an elegance all its own. It is independent of the inherent capabilities and limitations of any specific software or database-management system, and it takes advantage of well-developed, relational-database theory. As a result, as database-management systems evolve to fulfill more of the potentialities of database theory, a data model can continue to serve as the plan for a database when programmers work with new database application software (e.g., such commercial products as Visual FoxPro©, with which PaleoBank is programmed).

WHY PREPARE A DATA MODEL?

Data are the heart of any database. For data to be managed coherently and efficiently in any medium, electronic or otherwise, the data and their inherent structure must be understood comprehensively. The data-modeling process elicits such understanding (perhaps for the first time) and documents it verbally and symbolically in the completed data model.

A data model focuses attention first on the most important matter, the intended functions of a database. It then addresses how the different elements in the data are to be represented and defined. Finally, it shows how the elements of the data are interrelated. Creating the best model of these components requires careful thought and might be achieved only after a number of false starts, especially for such a complex system of data as is involved in biological and paleontological systematics.

In the process of designing the data model, the designers systematically determine the requirements and limitations of the database. The modeling process enables (and sometimes forces) members of the design team to develop a broad understanding of the overall project. A less tangible byproduct of the modeling process is the cooperation and sense of unity that develops among team members after struggling to model a particularly difficult concept. (The concept of junior homonymy was an especially formidable and persistent bugbear for our modeling team.)

A data model, though often specialized and technical, is diagrammed and written in English rather than in some sort of arcane code. Conventions for producing diagrams and written documentation are powerful and clear. They are also flexible and allow data modelers quite a bit of creative freedom in the design process. Rigorous definitions of terms emerge from the process, providing another building block in the foundation of the enterprise. The model is, therefore, the ideal means of communicating about the database, both during the design stage and subsequently during programming. It becomes the document of communication for designers, programmers, and users.

Because data modeling forces designers to sort out complex relationships, a comprehensive data model can save untold amounts of time in coding and debugging. Errors in understanding and conflicting interpretations of the structure of the data can be resolved in data modeling sessions before programming begins. Once the data model has been prepared and translated into the relational database design, a significant portion of the programmer's work is defined.

The PaleoBank data model provides an example of the flexibility we have stressed. The master data model can easily be modified to meet the individual requirements of a database for a specific group of fossils. For example, a database on a rather small group of primarily Paleozoic, colonial organisms, such as the stromatoporoids, is likely to have requirements that are appreciably different from a database that attempts to incorporate both biological and paleontological information on such a long-ranging and diverse group as the gastropods. Since all elements of the data and their relationships have been meticulously defined in the data model, however, modification of the software to accommodate different groups of organisms should be straightforward. In addition, PaleoBank will be able to incorporate data from other databases.

The data model allows the collection of data to be standardized and centrally controlled. Because data entry will be easy for contributors, we expect data to pour into the database. Ultimately reports from queries of the database at the Paleontological Institute will be available to contributors to facilitate their work. With respect to the *Treatise on Invertebrate Paleontology*, logjams of manuscripts waiting for editing will largely be eliminated, and manuscripts will be published with unprecedented timeliness.

Our experience has made us come to regard the preparation of the data model as being of utmost importance, just as the theory insists. Preparing a data model, however, does present some problems, many of which stem not so much from the modeling itself as from the approach one takes to it. Members of the design team need to use the Zen concept of beginner's mind and also good brainstorming protocol in order to forestall conflicts during the process. Because data modeling is a difficult and time-consuming task, problems may arise if the design team takes the task too lightly or does not budget enough time for it in advance of programming deadlines. A related problem is the temptation to cut corners in data modeling; where we have tried to do this we have paid dearly. The team must decide to model and to model properly. The final problem we encountered was knowing when to stop. A data model has a way of taking on a life of its own. It is essential that all members of the team realize that the model of a complex system of data will never be perfect and that it will never be finished. Data modeling must eventually wind down, and programming and data entry must eventually begin.

COMPONENTS

Ideas.—Entities, attributes of entities, and relationships among entities are the building blocks of data models. Nevertheless, the most important parts of any data model are not these cognitive components but the ideas that go into it. A logical data model should combine the knowledge of experts in one or more fields, and any modeling exercise seems likely to be most successful when the data-modeling team includes programmers as well as those who work with the data in the real world. Our team, for example, included taxonomists, paleoecologists, programmers, and editors, which allowed us to incorporate different kinds of knowledge into the data model.

Entities.—Entities are the representations in the data model of real-world concepts and objects. Some entities in the PaleoBank data model, for example, are BASIC TAXON, AUTHOR, PUBLICATION, ILLUSTRATION, SPECIMEN, and PLATE TECTONICS. By convention, the names of entities are written in capital letters. This is useful because it enables one to distinguish readily between PUBLICATIONs as formally defined in the data model and publications in general. A single example of an entity is referred to as an instance. The entire list of taxa, for example, comprises the entity BASIC TAXON, an instance of which is a specific taxon.

Attributes.—Each entity in the data model has characteristics that are represented by single-valued attributes, and the values of these attributes are the actual elements of data in the database. For example, the attributes of the PaleoBank entity SPECIMEN include the name of the species, the repository where it is stored, and the geographic location of the site from which the specimen was collected. For each of these attributes the data model contains a definition, a description of the domain of possible values for the attribute, and a set of business rules. The domain of the geographic location attribute, for example, might be all pairs of latitudes and longitudes, which would thus specifically rule out the use of any other geographic location systems. Of course, some other geographic location system could be used, but mixing two systems in the database would be unwise as it could cause reasonable queries of the database to lead to meaningless answers. The business rules, which are of special importance for the programmer, guard the consistency and integrity of the data, eliminate redundancies, and prevent nonsensical domain values from being entered.

Relationships.—The data model represents the real-world relationships among entities in several ways. Figure 1 shows the symbols used in the entity-relationship diagrams and illustrates the different kinds of relationships. The relationship between two entities is defined as **unconditional** if it is mandatory—that is, if there is at least one instance of entity B for every instance of entity A. For example, for every instance of the entity BOOK, there is at least one instance of the entity PUBLISHER. A **conditional** relationship, on the other hand, is not mandatory—there may be no instance of entity B for a given instance of entity A. The relationship between the entities PUBLICATION and KEY WORD is conditional because not all publications are assigned key words.

Two-directional relationships between entities are also defined according to their **cardinality.** Relationships are either one-to-one, one-to-many, or many-to-many (see Fig. 1). In a one-to-one relationship, each instance of entity A is related to a maximum of one instance of entity B and vice versa: for each GENUS, there is a maximum of one TYPE SPECIES, and for each TYPE SPECIES, there is a maximum of one GENUS. A one-to-many relationship exists when there may be a maximum of many instances of entity B for every instance of entity A, but a maximum of only one A for each B: there may be many SPECIMENs in each REPOSITORY, but only one REPOSITORY for each SPECIMEN. A many-to-many relationship reflects the fact that each instance of one entity may be related to many instances of the other: for each PUBLICATION there may be many AUTHORs, and each AUTHOR might write many publications.

DOCUMENTATION

The complete representation of the data model requires several components: entity-relationship diagrams, documentation of the entities and their attributes, and descriptions of the relationships among entities. We have made all this information available on the Paleontological Institute's World Wide Web page. Specific information on PaleoBank is on the PaleoBank main page. See Table 1 for URLs for these pages and for all pertinent documents.

Because of the elaborate formatting of some parts of the data model, we have elected to present the documents as Portable Document Format (.pdf) files. These files can

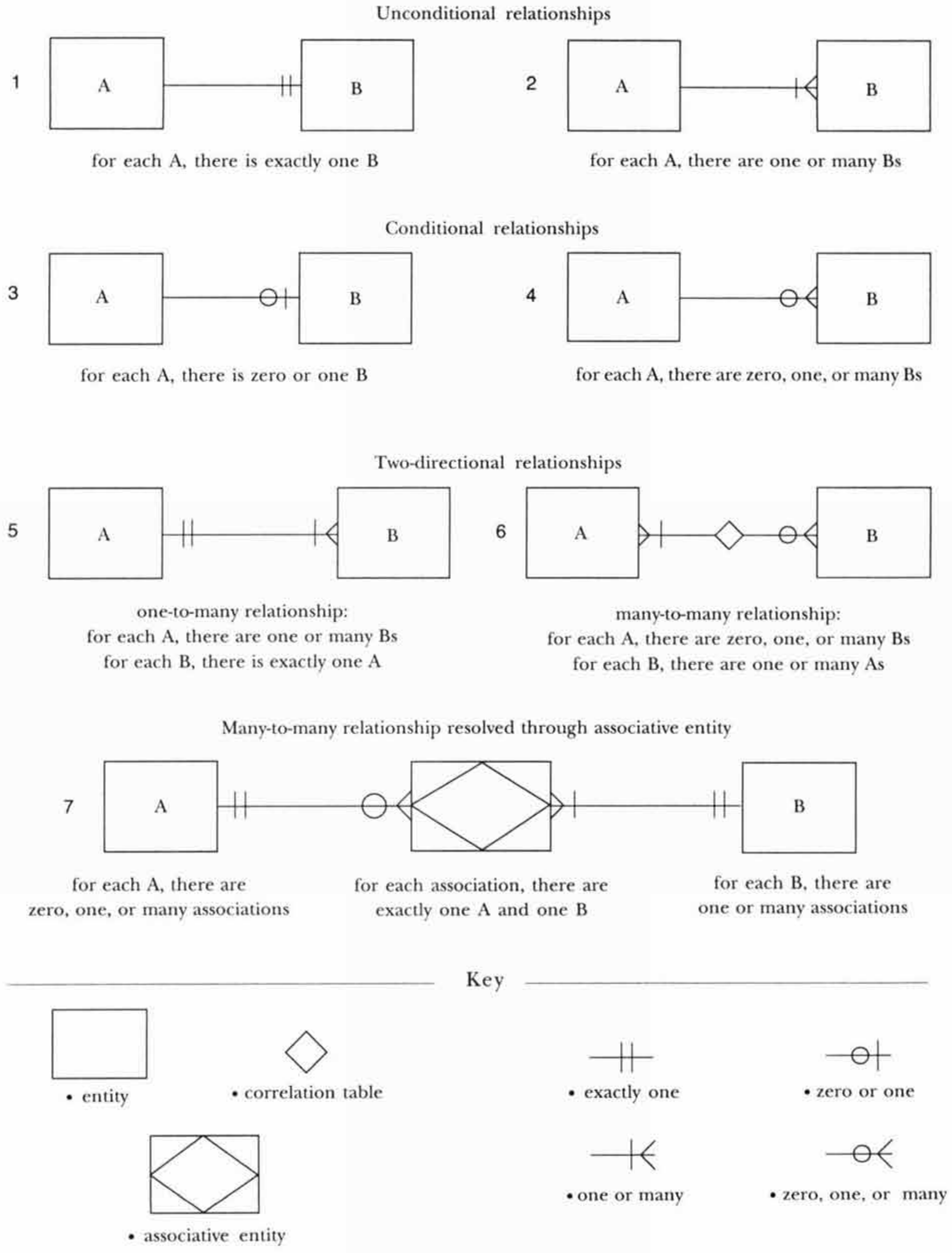
Unconditional relationships
1
A
B
for each A, there is exactly one B
2
A
B
for each A, there are one or many Bs
Conditional relationships
3
A
B
for each A, there is zero or one B
4
A
B
for each A, there are zero, one, or many Bs
Two-directional relationships
5
A
B
one-to-many relationship:
for each A, there are one or many Bs
for each B, there is exactly one A
6
A
B
many-to-many relationship:
for each A, there are zero, one, or many Bs
for each B, there are one or many As
Many-to-many relationship resolved through associative entity
7
A
B
for each A, there are zero, one, or many associations
for each association, there are exactly one A and one B
for each B, there are one or many associations
Key
• entity
• correlation table
• exactly one
• zero or one
• one or many
• zero, one, or many
• associative entity

Figure 1. Explanation of symbols used in entity-relationship diagrams. *1–4*, Unconditional and conditional relationships; *1–2*, the two kinds of unconditional relationships; *3–4*, the two kinds of conditional relationships. *5–6*, These diagrams illustrate that in the PaleoBank data model relationships between entities work in both directions. *7*, Example of many-to-many relationship. Many-to-many relationships are resolved either with correlation tables (see Fig. 1.6) or with associative entities.

be read with Adobe Acrobat Reader©. Full instructions for obtaining a copy of Adobe Acrobat Reader© free of charge are given with the documentation of the data model at the second of the two URLs given in Table 1.

Entity-relationship diagrams.—Once the conventions used in the entity-relationship diagram are understood, the diagram becomes a powerful tool in the modeling process. As the data model grows, the entity-relationship diagram progresses from an extremely simple drawing used during brainstorming sessions to the formal, graphical illustration of the entire data model showing all entities and their relationships. It is best to focus on specific parts of the diagram rather than trying to assimilate the entire diagram at once. If approached in this manner, the diagram makes it possible to apprehend quickly the scope of the data model. In conjunction with the written documentation, the diagrams are an invaluable guide to the workings of the data model.

Entity-relationship diagrams used in data models make use of several symbolic conventions. Unfortunately, these vary somewhat among practitioners because many modelers do as we did and develop conventions that are peculiar to their own modeling needs. The following description of the conventions used in the data model of PaleoBank is a collage of symbols and meanings taken from various systems, in particular the Knowledgeware symbols shown on the inside back cover of the book by Teorey (1990). See Figure 1.

Entities appear in our entity-relationship diagrams as rectangles, each containing in upper-case letters the name of the entity.

Relationships have several characteristics and components, and their representation is more complex than that of entities. Two entities are connected by a line if they have a relationship. The symbols on the lines show the cardinality of the relationship—i.e., whether the relationship is one-to-one, one-to-many, or many-to-many—and its conditionality (see section on relationships above, p. 4).

The documentation of PaleoBank includes four entity-relationship diagrams, which are published in full on the World Wide Web. Entity-relationship diagram 1 shows the TAXONOMIC CONCEPTS layer of the data model, diagram 2 shows the PUBLICATION-ILLUSTRATION layer, diagram 3 shows the MORPHOLOGY and the STRATIGRAPHY-GEOGRAPHY layers, and diagram 4 shows the ECOLOGY and the PLATE TECTONICS layers of the data model (see Table 1 for URLs).

Entities and attributes.—The entities and attributes document is a stripped-down version of the entity documentation. It lists only the names, definitions, and subtypes of the entities; relationships in addition to foreign keys; and the types and names of the attributes of the entities. In simpler data models, some of these are shown in the entity-relationship diagrams. Because of the complexity of our diagrams, however, we have presented this information in a separate document (see Table 1 for URL).

Entity documentation.—The entity documentation defines each entity, details its relationships, and describes all its attributes. These include an unequivocal identifier (the primary key), any foreign keys that refer to other entities, and all other attributes. For each attribute, a definition, a domain, and a set of business rules are given. This document is the largest and most complex of the data model (see Table 1 for URL).

Descriptions of relationships.—We have discussed how the real-world relationships among entities are diagrammed in the data model. These relationships are implemented through the use of correlation tables, foreign keys, or associative entities, which are a special kind of entity. Each relationship is described in the data model according to its function (PUBLISHER publishes BOOK), its cardinality (each PUBLISHER publishes many BOOKs), and its conditionality (see Table 1 for URL).

Relational database design.—With a data model in place, development of software is both predictable and straightforward. The design of the relational database flows directly from the data model, and with such fourth-genera-

Table 1. URLs giving access to electronic documents that comprise the data model of PaleoBank. Note that all but the first two of these documents are .pdf files and require the use of Adobe Acrobat Reader© for access. Information for obtaining this software free of charge from the publisher is available through a link from the PaleoBank main page listed in the table.

Subject	URL
Paleontological Institute home page	http://www.ukans.edu/~paleo
PaleoBank main page	http://www.ukans.edu/~paleo/paleobank.html
PaleoBank data model	http://www.ukans.edu/~paleo/cover.pdf
Entity-relationship diagrams	http://www.ukans.edu/~paleo/diag1.pdf
	http://www.ukans.edu/~paleo/diag2.pdf
	http://www.ukans.edu/~paleo/diag3.pdf
	http://www.ukans.edu/~paleo/diag4.pdf
Listing of entities and their attributes	http://www.ukans.edu/~paleo/entatt.pdf
Complete documentation of entities	http://www.ukans.edu/~paleo/entity.pdf
Description of relationships among entities	http://www.ukans.edu/~paleo/descr.pdf
Relational database design	http://www.ukans.edu/~paleo/rdd.pdf

Table 2. Members of the two boards who met to advise on *Treatise* policy and on the development of PaleoBank.

Board of Editorial Advisors	Board of Technical Advisors
Warren D. Allmon, Paleontological Research Institute	Arly Allen, Sheridan Electronic Systems
Simon Conway Morris, University of Cambridge	Stanley D. Blum, University of California, Berkeley
Jere H. Lipps, University of California, Berkeley	Douglas H. Erwin, Smithsonian Institution
Richard A. Robison, The University of Kansas	David K. Goodman, ARCO Alaska
James Sprinkle, University of Texas at Austin	David R. Lindberg, University of California, Berkeley
Barry Webby, University of Sydney	Robert Magill, Missouri Botanical Garden
Robin C. Whatley, University College of Wales	Paul Markwick, The University of Chicago
	Jerry Niebaum, The University of Kansas
	Robert Nunley, The University of Kansas
	F. James Rohlf, State University of New York, Stony Brook
	J. John Sepkoski, Jr., The University of Chicago
	E. O. Wiley, The University of Kansas

tion programming languages as Microsoft's Visual FoxPro©, programming the relational database is greatly simplified. The document that presents the relational database design for PaleoBank may be referred to on the World Wide Web (see Table 1 for URL).

AFTERWORD

Data modeling is a fundamental step in laying the foundation for a successful database. We hope that our explication of the data model for PaleoBank will provide potential users with insight into its inner workings, making the database more useful. We would like to receive comments from readers, especially regarding the data model for PaleoBank and our use of this hybrid publication incorporating both print and electronic media. We are also interested in hearing about other data models and databases that are in preparation, in particular those with a paleontological bent.

Acknowledgments.—We are grateful to the members of our two advisory boards listed in Table 2, the Board of Editorial Advisors and the Board of Technical Advisors, without whom the PaleoBank project is unlikely to have materialized. Among these we owe special thanks to Warren D. Allmon, Douglas H. Erwin, and David R. Lindberg. Above all we are indebted to Stanley D. Blum, who was instrumental in leading us (sometimes by the nose) into the fascinating arena of data modeling. Jerry Niebaum, Michael Grobe, and Grace Schroer of The University of Kansas Computer Center provided ongoing technical and moral support. At the Paleontological Institute, Jill Hardesty edited this paper and Karen Renteria prepared Figure 1. Work on PaleoBank, including the preparation of the data model and this paper, was supported by grants from the National Science Foundation, BSR-9024567 and DEB-9505100, for which we are grateful, and, insofar as PaleoBank is related to the *Treatise on Invertebrate Paleontology*, work was supported by the Raymond C. and Lillian Boggs Moore bequest through the Kansas University Endowment Association.

BIBLIOGRAPHY

Association of Systematics Collections Committee on Computerization and Networking. 1993. Report of the Biological Collections Data Standards Workshop, August 18–24, 1992: An information model for biological collections. Draft, March 1993. 92 p.

Blum, S. D., ed. 1991. Guidelines and Standards for Fossil Vertebrate Databases: Results of the Society of Vertebrate Paleontology Workshop on Computerization. Department of Vertebrate Paleontology. American Museum of Natural History. New York. 129 p.

Brooks, F. P., Jr. 1982. The Mythical Man Month: Essays on software engineering. Addison-Wesley. Reading, Massachusetts. 195 p.

Codd, E. F. 1969. Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks. IBM Research Report RJ599.

——. 1970. A relational model of data for large shared data banks. CACM 13(6):377–378.

Crosby, M. R., & R. E. Magill. 1989. TROPICOS: A botanical database system at the Missouri Botanical Garden, "The Booklet." Missouri Botanical Garden. St. Louis. 39 p.

Date, C. J. 1995. An Introduction to Database Systems, 6th ed. Addison-Wesley. Reading, Massachusetts. 839 p.

Fleming, C. C., & B. von Halle. 1989. Handbook of Relational Database Design. Addison-Wesley. Reading, Massachusetts. 605 p.

Ride, W. D. L., C. W. Sabrosky, G. Bernardi, and R. V. Melville, eds. 1985. International Code of Zoological Nomenclature., 3rd ed. University of California Press. Berkeley. 338 p.

Shlaer, Sally, & S. J. Mellor. 1988. Object-Oriented Systems Analysis: Modeling the world in data. Project Technology, Inc. Yourdon Press. Englewood Cliffs. 143 p.

Teorey, T. J. 1990. Database Modeling and Design: The entity-relationship approach. Morgan Kaufmann. San Mateo. 267 p.