PhyloInformatics
journal for taxonomists

Taxonomer: a relational data model for managing information relevant to taxonomic research

Richard L. Pyle

Department of Natural Sciences
Bishop Museum
1525 Bernice St.
Honolulu, HI 96817, USA

Received: 22 October 2003 - Accepted: 6 January 2004

Abstract

Taxonomic research, as a field of biological sciences, is fundamentally an exercise in information management. Modern computer technology offers the potential for both streamlining the taxonomic process, and increasing its accuracy. Effective use of computer technology to successfully manage taxonomic information is predicated upon the implementation of data models that accommodate the diverse forms of information important to taxonomic researchers. Although sophisticated data models have been developed to manage some information relevant to taxonomic research (e.g., natural history specimen information; descriptive data relating to morphological and molecular characters of specimens), similarly robust models for managing information about taxonomic names and how they are applied to taxonomic concepts, though they exist, have not attained widespread use and adoption.

Herein I describe portions of a relational data model developed to manage information relevant to taxonomic names and concepts. The core entities of the described portions of this model are *Agents, References*, and *Assertions* (along with their associated *Protonyms*). Agents (people and organizations) in this context refer primarily to taxonomic authorities. References are broadly defined as date-stamped information (usually, but not exclusively, in the form of a publication), as documented by the Agents who serve as the Reference authors. Assertions consist of basic elemental information about the treatment of taxonomic names by taxonomic authorities as documented in a particular Reference, and correspond to what many authors refer to as taxon "concepts". Protonyms are a special subset (subtype) of Assertions, which constitute original descriptions of taxonomic names (serving to unite multiple assertions pertaining to the same taxonomic name), and include elements of botanical Protologues and Basionyms.

I also illustrate how these core entities can serve as a foundation for taxonomic names and concepts as integrated with other datasets, such as biological specimens and observations (and, by extension, geographic distributions and character matrices). The broadest data content source used to populate and test the data model is derived from a systematic revision of the reef-fish family Pomacanthidae (marine angelfishes). Additional datasets used to test the implementation of the data model include specimen data from the Department of Natural Sciences, Bishop Museum; nomenclatural data from *The Catalog of Fishes*; and nomenclatural and biogeographic data from two published taxonomic catalogs (insects and terrestrial mollusks in Hawai'i).

An intuitive, feature-rich software application based on Microsoft Access® has also been developed in conjunction with this data model, and will be the topic of a future article.

Introduction

More so than in many other fields of biological research, taxonomy is ultimately about managing and organizing information. New species descriptions, systematic revisions, and biogeographic analyses are based on information associated with and derived from biological specimens. Such information includes details related to the circumstances of the specimens as they were found to occur in nature (geographic location, macro- and micro-habitat details, etc.), as well as morphological and biochemical characters exhibited by those specimens. It also includes the need to track and index historical literature relating to taxonomic names and concepts, going back two and a half centuries (Minelli, 2003). Indeed, unlike many other avenues of biological research (which are usually based on limited data sets obtained from specific experiments designed to test certain hypotheses), taxonomic researchers must draw from a much larger and more diverse pool of data from a variety of disparate sources. This need can present a significant information management challenge.

Throughout history (and continuing to the present), most taxonomists have relied on "manual" systems and techniques to gather, organize, and synthesize the information necessary to conduct their research (*e.g.,* Winston 1999). Published and unpublished references cite information contained in other published and unpublished references; researchers travel (sometimes over great distances) to museums in order to examine specimens directly; species descriptions are usually formatted and generated on a case-by-case basis, synthesizing hand-written notes and data sheets into summarized tables, diagnoses, and descriptions; and distribution patterns of species are compiled manually from many and varied sources (often without consistent documentation of such sources).

Few would dispute the observation that taxonomy, as a field, faces greater perils than it has throughout much of its history (*e.g.,* Lee, 2000; Godfray, 2002; Mims, 2003). An ever-increasing demand for high-quality taxonomic information is falling on the shoulders of an ever-dwindling supply of taxonomists with enough experience and training to provide such high-quality information. In response to this situation, there have been an increasing number of proponents of using computer technology and the internet to facilitate the taxonomic process in ways never-before possible (*e.g.,* Bisby, 2002; Gewin, 2002.; Godfray, 2002; Moretzsohn, 2002). While taxonomy is ultimately limited by a dearth of taxonomic expertise, information technology can improve the efficiency and consistency of work that is performed by existing taxonomists.

Among the earliest to adopt computer technology to assist in the taxonomic process were natural history collections utilizing specimen databases. SELGEM (Creighton & Crocket, 1971) was perhaps the first major effort to use computer technology to organize natural history collections data, using punch cards (and later ticker tape). The database application *MUSE* (Humphries, 1994) was one of the earliest to attain widespread use. In the years that followed, a plethora of similar systems followed suit, such as: *BIBMASTER* (Pando, 2001), *BioLink* (Shattuck & Fitzsimmons, 2000), *BioOffice* (BIOGIS Consulting, 2003); *Biota* (Colwell, 2002); *Biótica* (CONABIO, 2003), *BRAHMS* (Filer, 2001); *Vernon* (Vernon Systems, 2003), *Herbar* (Pando & Anonymous, 2003); *KE EMu* (KESoftware, 2003), *MANTIS* (Naskrecki, 2003); *MVZ Collections Information Model* (Blum, 1996); *SAMPADA* (NCBI, 2002); *Specify* (IBRC, 2003); *TAXIS* (Bio-Tools.Net, 2003); and *Tracy* (Minnigerode, 1998); among others. Most of these models were developed with extant taxa in mind, but Morris (1998) describes a data model designed to accommodate paleontological data.

While many of these specimen-centric data management systems include (sometimes extensive; e.g., *BioLink*) taxonomic components, other computer databases and applications have focused specifically on taxonomic information (*e.g., Linnaeus II* [ETI, 2003]; *MacTaxon* [Dessein & Schols, 2003]; *PISCES* [Eschmeyer, 1995]; *Platypus* [ABRS, 2003 – the progenitor of the taxonomic components of *BioLink*]; *SysTax* [Hoppe et al., 1996; Hoppe & Ludwig, 2003];

Taxon-Object [Saarenmaa, 1995]; etc.). However, Pullan et al. (2000) point out that many of these taxonomic databases are designed to accommodate only a single taxonomic "view" or classification scheme, which imposes serious limitations on the ability to reflect the true dynamic nature of taxonomic nomenclature, as used to represent taxonomic concepts.

Many authors have discussed and described the "concept problem" in taxonomy; that is, the distinction between a taxonomic name, and the scope of organisms implied by the name. Geoffroy & Berendsohn (2003) provide an excellent overview, and I discuss it from a the perspective of the specific data model described herein. In summary, taxonomic *names* (text character strings, as established according to codes of nomenclature) have historically been used to represent taxonomic *concepts* (sets of individual organisms collectively representing a particular taxon circumscription). The "problem" stems from the imprecise correlation between names and concepts: the same taxon concept might be represented by more than one available name; and the same taxon name is often used by different authorities to represent different sets of organisms (*i.e.,* different concepts). This historically pervasive disjunction between names and concepts represents a barrier to modern taxonomic information management.

While this "problem" has been identified and discussed for many years, only relatively recently have a number of more or less independent efforts attempted to address the "concept problem" in the context of data models and information management schemes (Anonymous, 2002; Berendsohn, 1995; 1997; Geoffroy & Berendsohn, 2003; Gradstein et al., 2001; Koperski et al., 2000; Le Renard, 2000; Pullan et al., 2000; Raguenaud, 2002; Ytow et al., 2001; Zhong et al., 1996). Most of these models attempt to define the scope of taxon concepts using either publications or specimens. Although these alternative approaches have many similarities, the differences between them are usually a reflection of different operational paradigms (*e.g.,* botanical taxonomy versus zoological taxonomy) or different information priorities. None has yet risen above the others as the clear path to taxonomic information management "salvation," and most emphasize that they are preliminary, in development, and/or subject to future modification.

The data model described herein (called "*Taxonomer*") is proposed as one specific approach to organizing and managing information about taxonomic names and the concepts they are intended to represent. It is the culmination of nearly fifteen years of development, which began as an effort to manage specimen data for the B.P. Bishop Museum (BPBM) ichthyological collection. The taxonomic component of the model arose from an attempt to integrate an electronic version of the *Catalog of the Genera of Recent Fishes* (Eschmeyer, 1990), and later *The Catalog of Fishes* (Eschmeyer, 1998) as a taxonomic authority for the BPBM fish specimen database. As the system expanded over the years, it grew to encompass other specimen collections at BPBM (Botany, Entomology, Malacology, Vertebrates, Marine Invertebrates) and took on the more generalized purpose of managing a wide range of information associated with taxonomic research activities, broadly including "agents" (people and organizations), publications and other reference citations, taxonomic names and concepts, specimens (and their associated morphological character data), observations, images, geographic place names and descriptions, and an assortment of other related data sets. In addition to the BPBM specimen databases and *The Catalog of Fishes*, the model was tested for its ability to accommodate historical taxonomic data using three separate data-sets. The first is a broad and comprehensive (*i.e.,* spanning the full suite of taxonomic information management needs) set of data concerning the taxonomic revision of the marine fish family Pomacanthidae. The second is an exhaustive catalog of insect taxonomy for the Hawaiian Islands, cross-referenced to an extensive bibliography (Nishida, 2002). The third is a taxonomic catalog of terrestrial and freshwater mollusks of the Hawaiian Islands (Cowie et al., 1995). A wide array of other taxonomic and related data from various sources have been used to test the effec-

tiveness of the model for managing diverse taxonomic data management needs.

The data model was also influenced by personal communication with Stanley D. Blum (currently of the California Academy of Sciences), and by the MVZ Collections Information Model (Blum, 1996). Unless otherwise stated, the structure of the model was developed independently of other data models with analogous functions, and similarities to other such models are, in almost all cases, convergences of design. This last point is emphasized only to suggest that when independent data model developers converge on similar structures, it may reveal fundamentally optimal solutions to common information management needs. Specific examples of such convergences in the context of this and other taxonomic data models are included in the "Discussion" section of this article.

A feature-rich user-interface application was developed concurrently with the data model, using Microsoft Access® software (versions 1.0 through 9.0). The complete system (data model and application) bears the name "*Taxonomer*," though this article describes only the taxonomic components of the data model. A full description of the complete application will be the subject of a future article.

The data model presented here (hereinafter referred to as the *Taxonomer* data model) is not intended as a proposed standard for broader adoption. Rather, it is a detailed description of my own approach to solving taxonomic data management needs, with the hope that some of the ideas and perspectives presented herein will be of use to others who are engaged in similar endeavors.

System and Methods

The "Implementation" section below is divided into four sections, the first three of which describe the three major data components (**Agents**, **References** and **Taxa**), and the fourth section describes how certain other components of the full *Taxonomer* data model interface with these three components. Each of the first three sections is further subdivided into three subsections: an introductory preamble (describing the general context of the section), individual *Table Descriptions* (describing each table and fields), and *Limitations* (acknowledged limitations or aspects of the data not accommodated by the described model). Each *Table Description* section highlights a major table and its associated dependant tables and relationships. Table names are formatted in **bold**, with the "**tbl_**" prefix included. More general references to the entity represented by the table are similarly in **bold**, but lack the "**tbl_**" prefix. Individual attribute (field) names, when referred to in the text, are shown in *italics*.

The key to the meaning of elements in various figure diagrams of physical data models is shown in Figure 1. The "core" table for each major component (*i.e.,* the table to which foreign keys of tables in other major components join) are shown in blue, and supporting tables are shown in white. The top line in each table box is the table name. Four categories of attributes are distinguished:

- **Unique Keys.** The attributes in the this section of each table box represent the uniquely-identifying key fields for each table. All tables have a surrogate Primary Key, which by convention takes the name of the table (minus the "tbl_" prefix), with the addition of an "ID" suffix (indicated in **bold** in the diagrams, with a "**P**" indicator). In my implementation, these surrogate keys are almost always long integers, with automatically-assigned "random" (arbitrary) values (*i.e.,* with no inherent information content). In cases where a table is limited to a relatively small, finite number (<255) of instances of defined values, often with an inherent sort order, the surrogate primary key is of type "Byte," and values are assigned according to the appropriate sort sequence (when applicable). This departure was made to allow improved performance of certain queries, and to simplify coding in the *Taxonomer* application. Also, each table is populated with a single special record having a surrogate key ID value of 0 (zero), that serves as an "Unspecified" indicator.

This place-holder record is provided in each table to allow enforcement of non-null rules for Foreign Keys (*i.e.,* when a Foreign Key would otherwise be left unpopulated, it is instead populated with a value of zero, serving the equivalent informational content of "unspecified value"). In addition to the surrogate keys, when "natural" (information-bearing) keys exist for a table (either as a single attribute, or composite set of attributes), they are similarly listed in this section. In many cases, individual attributes that form part of a composite natural key also represent Foreign Keys to other tables.

- **Foreign Keys.** This section includes all Foreign Key attributes (except those that constitute part of a composite natural key, as described above). These serve as the linking field to the surrogate Primary Key of another table. They are shown in **Red Bold** text, and include an "**F**" indicator.

- **Non-Key Attributes.** These are actual data-bearing Attributes, not representing foreign keys to other tables.

- **Cheat Fields.** These are "artificial" system fields created solely for the purpose of enhancing multi-record processing performance. They are non-data-bearing in the sense that they only contain derived data (*i.e.,* derived from other fields in the containing table or in linked tables). These can be completely eliminated from the model without resulting in any loss in information content, and are thus not correctly represented as "attributes." Users **never** have editing access to these fields – they are maintained en-

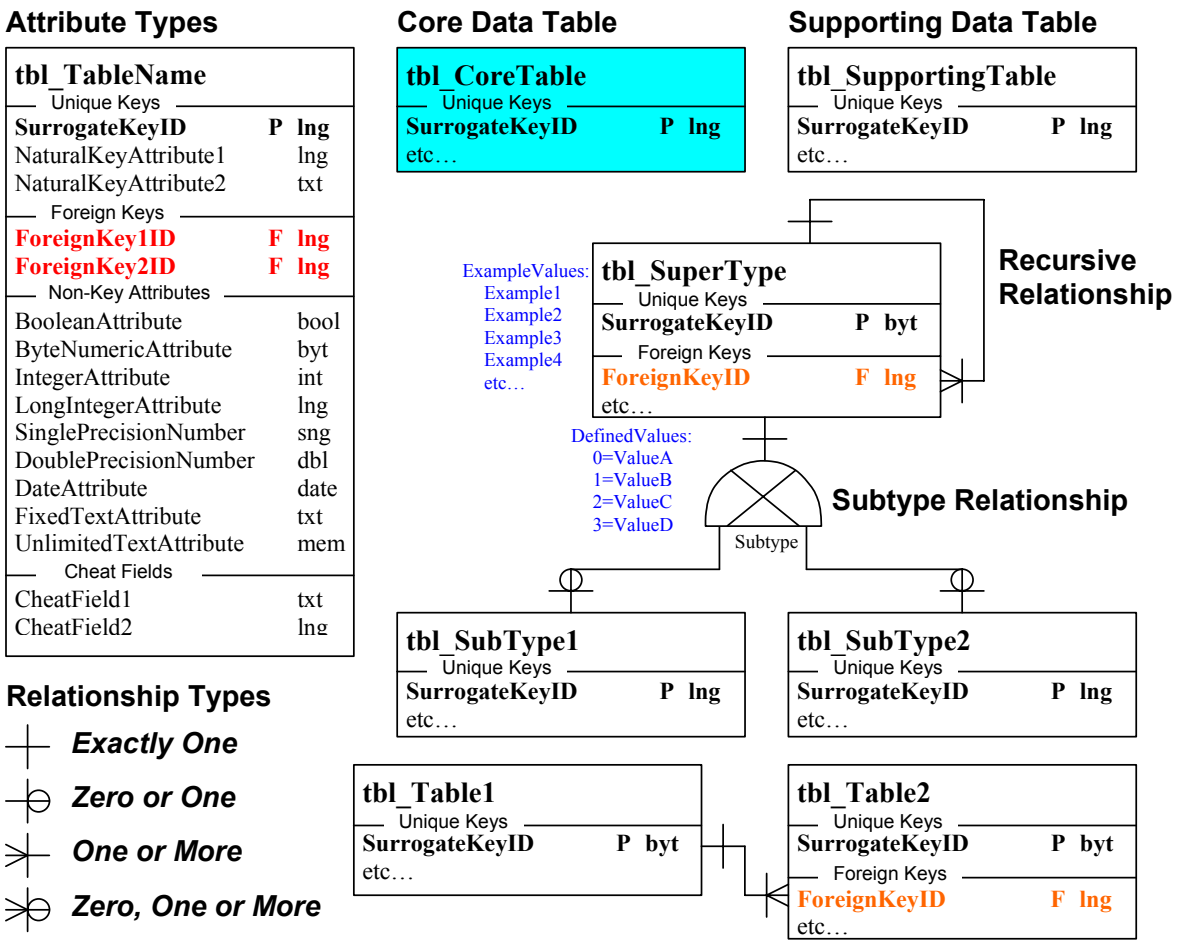

Figure 1. Key to physical data model diagrams.

tirely by software code, and are only exposed to users indirectly to enhance the performance of query/search/sort activities. The fields in this category are so designated by a "Cheat" prefix in their names. Although they have no bearing on the information model, they are described herein to illustrate how application performance costs associated with highly normalized data structures can be mitigated through their judicious implementation.

The right-hand column in each of the table boxes indicates the data type for each attribute. The data type codes are listed in Table 1 along with their corresponding data types as they exist in the Microsoft Access® application. Approximate corresponding data types for Microsoft SQLServer® are also shown, along with the size and value domains for each (based on Microsoft Access®).

Whenever possible, lines representing relationships (joins) between tables are drawn in such a way that they connect directly to the attributes that participate in the relationship. Two consistent exceptions to this are recursive (self) relationships and Supertype-Subtype relations. In cases of the former, the connection point for the "many" side of the relationship is generally aligned with the appropriate field, but the "one" side connects to the top of the table box, with the implication that it joins to the surrogate Primary Key. Supertype-Subtype relationships are indicated with a half-circle symbol. A cross inside the half-circle indicates that Subtypes are mutually-exclusive. In most cases, both sides of Supertype-Subtype relationships connect to the top or bottom of the table box, with the implied connection being between the surrogate Primary Keys of tables on both sides of the join (*i.e.,* "one-to-one"). In other cases where lines could not be aligned with associated attributes, the attributes involved with the relationships are usually evident (*i.e.,* to the surrogate Primary Key). In a few cases, the lines connect directly between the top of one table box and the bottom of another (*e.g.,* between **tbl_Thesaurus** and **tbl_Glossary**, and between **tbl_Reference** and **tbl_ReferenceBibliography**, as shown in Figure 4.)

Four different symbols are used to indicate the nature of each table join, as shown in the lower-left corner of Figure 1. A simple perpendicular line indicates that exactly one record in the corresponding table participates in the join. A perpendicular line with a circle indicates that one or zero records may participate in the join. A perpendicular line with a "crow's foot" (two extra angled lines) indicates that one or more records must participate in the join. Finally, a perpendicular line with both a circle and a "crow's foot" indicates that zero, one, or many records in the corresponding table may participate in the join. One example of a "one-to-many" join is included in Figure1, but joins may include any combination of the four symbols. It is important to note that, in many cases where the join to a Foreign Key attribute is shown with a circle (*i.e.,* allowing for zero linked records), an actual value of 0 (zero; equivalent to "Unspecified" as described above) is entered into the Foreign Key field when it would otherwise be Null, thus allowing enforcement of non-null values in Foreign Keys. This practice is implemented both to enhance output query performance, and to utilize referential integrity rules built into Microsoft Access® application software. Thus, although these joins should techni-

Table 1. Key to data types used in physical model diagrams.

Data Type	Microsoft Access®	Microsoft SQLServer®	Size	Domain (Microsoft Access®)
bool	Yes/No	bit	1 bit	0 or –1
byt	Number (Byte)	tinyint	1 byte	0 to 255
int	Number (Integer)	smallint	2 bytes	–32,768 to 32,767
lng	Number (Long Integer)	int	4 bytes	–2,147,483,648 to 2,147,483,647
sgl	Number (Single)	real	4 bytes	7-decimal precision
dbl	Number (Double)	float	8 bytes	15-decimal precision
date	Date/Time	datetime	8 bytes	Year 100 to 9999
txt	Text	varchar	0-255 bytes	<=255 characters
mem	Memo	text	0-64KB	<= 64,000 characters

cally be represented without the circle symbol (implying the requirement for at least one entry, even if it is the "Unspecified" place-holder zero value) to reflect the actual implemented procedure and business rules, they are shown with a circle because *conceptually* there is no requirement for a joining instance.

For the fields with only a few defined domain values, those values are usually listed in blue text beneath or adjacent to the corresponding table box, with their numeric equivalences to text-string values. In other cases, where there may be a limited number of values within the domain of a field, but where they are not universally known and defined, a similar list in blue text is provided, except without defined numeric equivalencies. In most cases, such "example" lists include "etc…" at the bottom.

Other comments (*e.g.,* business rules) are added to the diagrams for various relationships, to enhance clarity. The *Taxonomer* application makes extensive use of business rules and other data integrity enforcement procedures. Although some of these are described herein (either in the text, or as annotations on the diagrams), the majority are not. The emphasis of this article is to describe the individual data elements, the basic structure of how those elements are arranged in tables, and how tables are joined via relationships. A complete list of business rules and other referential integrity procedures will be included in the forthcoming article describing the *Taxonomer* application.

Implementation

The descriptions herein focus on those components associated with taxonomic information management. This narrowed focus was followed because these are the most well-developed components of the full model; because these components are more in keeping with the scope of this journal; and, perhaps most of all, because these components address an area of biological informatics that is just now coming to the forefront of active development across a broad international community. Neverthe-

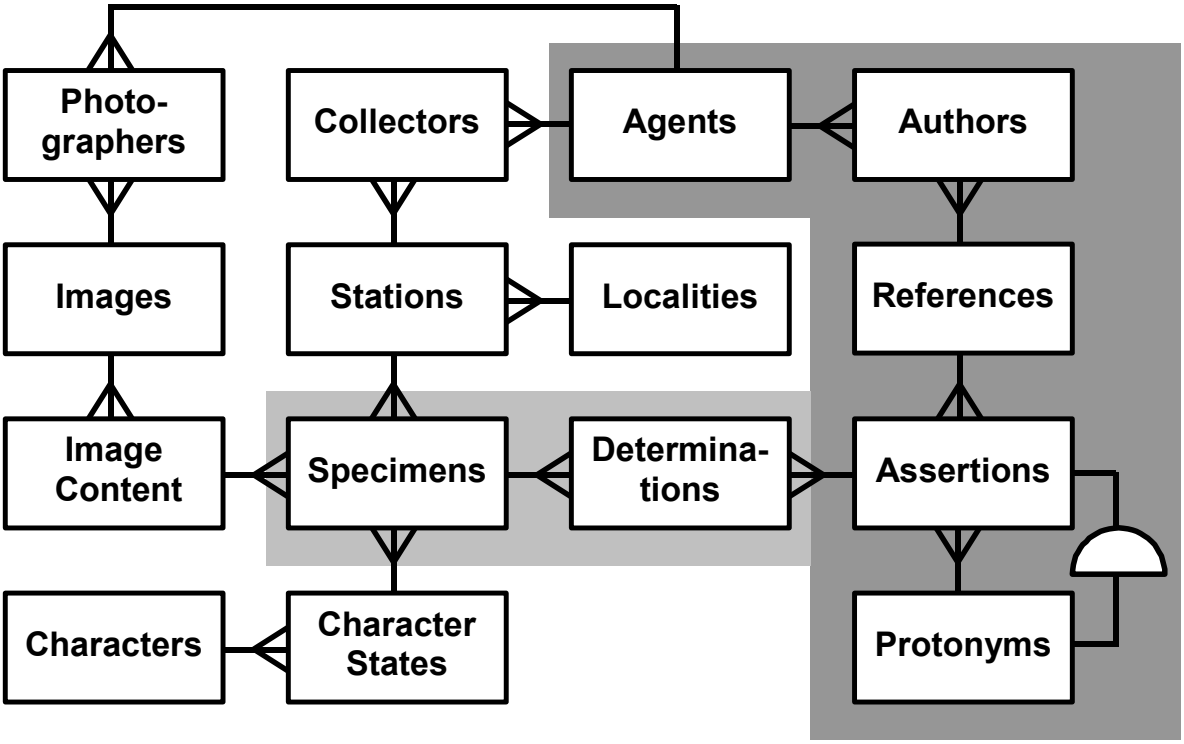

Figure 2. Conceptual overview of the core *Taxonomer* data model (excludes specimen transaction management and population assessment components). Areas highlighted in dark gray constitute the primary focus of this article, and areas shaded in light gray are discussed in terms of how they interface with the primary components.

less, to provide a broader contextual placement of the described components, a highly simplified conceptual schema of a more complete version of the full model is illustrated in Figure 2.

Agents

The physical model for **Agent** data is represented in Figure 3. The term "**Agent**" (synonymous with "Party," as used by Taswell & Peet, 2000, and others) was introduced in the context of biological databases in the ASC data model (ASC, 1993), and applies to an individual human (**Person**), or an organized group of humans (**Organization**). **AgentAssociation** instances may be established between any combination of a **Person**, and/or an **Organization**, and/or an **Address**. A minimum of two of these three values must be included for any single instance of **AgentAssociation** (*i.e.,* no "association" can be made within only one of these three). For each **AgentAssociation**, there may be zero to many **EContacts** (*e.g.,* telephone and fax numbers, telex, email addresses, websites, etc.). For convenience, each **Agent** is indicated by a default **AgentAssociation** instance, to select one of potentially several **AgentAssociation** instances as representing the set of preferred contact details.

tbl_Agent

Every **Agent** instance is assigned a *ValidAgentID*, corresponding to the particular "alias" of the agent that is currently regarded as valid. If *ValidAgentID=AgentID* for a particular instance, then that specific instance represents the "most correct" variation of that **Agent**. If *ValidAgentID≠AgentID*, then the current **Agent** instance is regarded as a "junior alias" of the record indicated by the value of *ValidAgentID*. In all cases, the value in *ValidAgentID* must be drawn from the set of "valid" **Agent** instances (*i.e.,* where *ValidAgentID=AgentID*). The *ValidAgentID* field may not contain a Null value nor a "0" (**Agents** are assumed to be valid). The *ValidAgentID* system is primarily intended to map people or organizations who have used different names over the course of their lives (*e.g.,* maiden name and married name, organization renaming, etc.), however it is also used to record different variations of the same name for a single **Agent** (*e.g.,* when a person serves as the role of **ReferenceAuthor** to different publications using different sets of given-name initials, or different styles of the same multipart last name, or different translations of the same name in different languages, etc.). It is important to clarify that instances within this table do not necessarily represent a single "**Agent**" (**Person** or **Organization**), but actually represent various *NAMES* that have been applied to individual **Agents**. Unique **Agents** can be quickly identified as those instances where *ValidAgentID= AgentID*. This logic cascades to apply to **Organization** and **Person** subtypes.

Every instance of **Agent** is assigned an *AgentTypeID* value that corresponds to an existing instance of the **tbl_AgentType** table, indicating which Subtype the **Agent** represents. This data model currently allows only two *AgentType* values – **Person** and **Organization** – but additional *AgentType* values may be defined in the future (*e.g.,* "Team," which would represent a set of multiple Agents who do not collectively constitute an **Organization**). In addition to the zero-ID "Unspecified" instance in **tbl_Agent**, there is also a more specific "Unspecified" instance for each Subtype (*i.e.,* "Unspecified **Person**" and "Unspecified **Organization**"). Hence, there is at least one instance of **tbl_Agent** for each **AgentType** (as indicated in Figure 3).

The *DefAgentAssociationID* Foreign Key is provided to select one of potentially several different **AgentAssociation** instances as the "default" instance, from which to derive primary contact details (see description of **tbl_AgentAssociation** below).

An **Agent** is flagged as *Ambiguous* if the instance does not represent a specific, identified individual **Person** or **Organization**, but rather a generic **Person** or **Organization** (*e.g.,* "local fisherman," "fish market," etc.).

Each **Agent** has a *BirthDate* (or the founding date of an organization), and a *DeathDate* (or the termination date of an organization). These values are useful for distinguishing different **Agents** with similar or identical names.

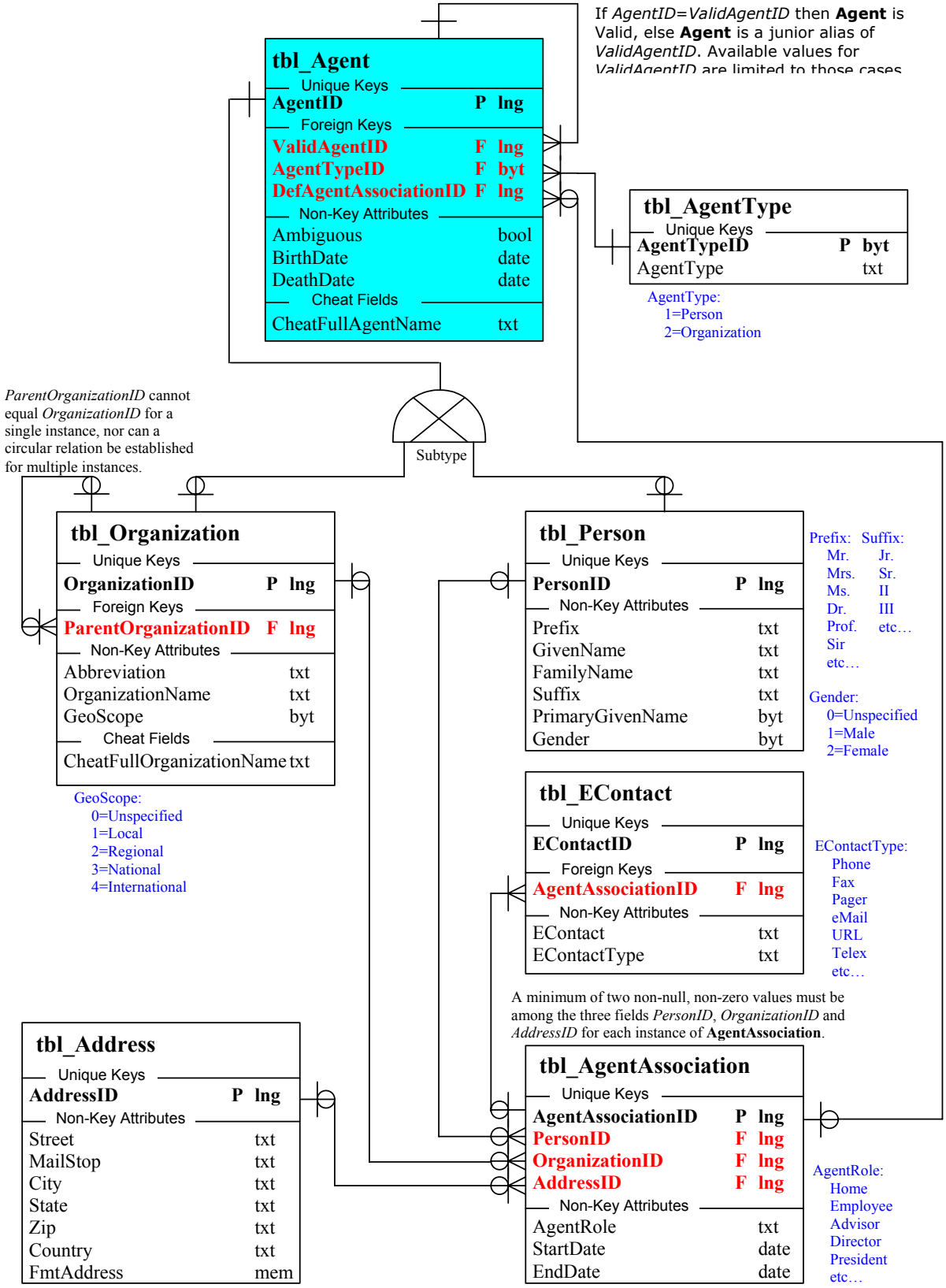

Figure 3. Agent physical data model.

CheatFullAgentName is used store a text string representing a consistently formatted name of the Agent, for faster display in output queries. The format for **Agent** instances of type **Person** is: "*FamilyName, GivenName, Suffix (Prefix)*." The format for instances of type **Organization** is: "*[Parent]OrganizationName; OrganizationName*" (all levels of parent **Organization** names are included, representing the complete organizational hierarchy).

tbl_Organization
Organizations represent one of the defined subtypes of **Agents**. Conceptually, an **Organization** is a place-holder for the collection of individual persons who form the **Organization** (*i.e.,* an "organization of people"). Informal sets of multiple individual persons (*e.g.,* a set of authors for a particular reference, or a set of collectors for a particular specimen) generally do not constitute an **Organization**; rather, **Organizations** exist as a collection of people independently of who those particular people are at any given point in time.

Organizations can be nested hierarchically, such that any **Organization** might be a subset of a "Parent" **Organization**, as indicated by *ParentOrganizationID*. Because no form of systematic "Rank" is applied to individual **Organizations** in this implementation of the model (*e.g.,* "Department," "Division," "Working Group," etc.), code must be used to enforce the business rule that no organization can be its own parent, and no chain of multiple **Organization→[Parent]Organization** links can be circular.

Organizations often have an *Abbreviation* (sometimes thought of as an acronym) and an *OrganizationName*, which are the text-strings used to represent the organization. An organization can be semi-objectively classified according to its *GeoScope*, using pre-defined values ranging from "Local" to "International" (allowing also for "Unspecified").

CheatFullOrganizationName is used differently from **CheatFullAgentName**; whereas the latter provides the full hierarchical-context name of the specific **Organization** in a format suitable for direct output; the former contains embedded *OrganizationID* values, used for parsing in certain kinds of output queries and drop-down lists. A semicolon is used as the delimiter (and also as leading and trailing characters), with alternating values of OrganizationID and OrganizationName for the entire hierarchy:

;ParentOrganizationID;ParentOrganizationName;
…;OrganizationID;OrganizationName;

tbl_Person
The other defined subtype of **Agent** is **Person**. As explained earlier, each unique **Person** may be represented by multiple instances in this entity – one for each different "alias" or name variation. However, the unique individual **Persons** can be easily identified by filtering on cases where *PersonID* is equal to the corresponding *ValidAgentID* in **tbl_Agent** (this applies equally to **Organizations**).

The core fields of this table primarily involve different elements of a **Person**'s name: *Prefix*, *GivenName*, *FamilyName*, and *Suffix*. *Prefix* and *Suffix* are straightforward, with examples given in the diagram. *GivenName* includes all elements of a person's given name, with each element separated by a space. *FamilyName* includes all elements of a person's family name (*i.e.,* including "de," "van der," etc.). *PrimaryGivenName* is a "Byte" integer (*i.e.,* "tinyint") representing which sequential name element of a multi-part *GivenName* is used as the primary given name. For example, for the name "John Edward Smith," the *GivenName* would be entered as "John Edward" (with a space delimiting the two given names). A *PrimaryGivenName* value of 1 would indicate that the name is formatted typically as "John E. Smith," and a value of 2 would indicate "J. Edward Smith." A *PrimaryGivenName* value of 0 indicates an unspecified primary given name. *Gender* indicates whether the person is Female (2), Male (1), or unspecified (0).

tbl_AgentAssociation
The primary function of this table is to track associations between **Organizations** and individual **Persons**. In most cases, this table simply serves to establish a many-to-many

relationship between people and organizations; but the function is more complex than this, because this table also serves the purpose of connecting an Association with an instance of the **tbl_Address** table. Consequently, either of the Foreign Key fields *PersonID* or *OrganizationID* (but not both) can contain a zero (≈null; see discussion above) value, but only if *AddressID* for that instance is non-zero (≈non-null). Such an instance would allow for linking an **Address** directly to either an **Organization** or a **Person**, without the need to establish an **Association** between an **Organization** and a **Person** (*e.g.,* a **Person**'s home address, or an **Organization**'s general address). If both *PersonID* and *OrganizationID* are non-zero (≈non-null) for a given **AgentAssociation**, then *AddressID* may be zero (≈null) for that instance (but doesn't have to be). (see also ASC, 1993).

The *AgentRole* for each instance of **tbl_AgentAssociation** is intended to represent the role played by the **Person** at the associated **Organization**. Examples are given in blue text in the diagram.

Each **AgentAssociation** has a *StartDate* and an *EndDate* to establish the window of time in which the **AgentAssociation** existed.

In principle, no instance should exist in the **tbl_Address** entity, unless it exists in at least one instance of **AgentAssociation**. Thus, the former is a "dependent" entity of sorts, even though it serves on the "one" side of a one-to-many relationship. The individual attributes of **tbl_Address** do not need elaboration, except perhaps for *FmtAddress*, which contains a fully-formatted mailing address to be entered or modified by the user. Usually, this field is automatically generated – derived from the other fields in this table – but it is not treated as a "Cheat" field because the user is allowed to over-ride the auto-formatting, to meet some particular address formatting situation. This should be regarded as an optional, application-defined field, rather than a core field.

Whereas only one **Address** can be linked to any particular **AgentAssociation**, there can be many instances of the **tbl_EContact** table linked to a given **AgentAssociation**. The concept of **EContacts** represents any sort of electronic contact number or text string, such as various telephone and fax numbers, TELEX, email addresses, web URLs, and other such electronic points of contact. The type of **EContact** is indicated by the *EContactType* field, examples of which are given in blue text in the diagram.

Limitations

- **AgentAssociations** cannot be made directly between one **Person** and another **Person**, or between one **Organization** and another **Organization**, except for the special case of "Aliases" (by way of the *ValidAgentID* recursive Foreign Key in **tbl_Agent**), and of an **Organization** linking directly to a "parent" **Organization**. Such associations (*e.g.,* between husband and wife, or between two organizations joined by an MOU or other agreement) are considered to be outside the scope of this data model. Additional tables could easily be appended to this model to track such associations. To accommodate such relationships within the current context, one could re-define the *OrganizationID* and *PersonID* Foreign Keys of **tbl_AgentAssociation** to be *AgentID* and *AssociatedAgentID* (without restriction of which Subtype each is drawn from), but it would need to accommodate tracking directionality of such a relationship (perhaps in place of *AgentRole*).

- To link **EContacts** directly to a single **Person** or **Organization** (without the context of the other), an *AddressID* must be provided for that Person or Organization. This limitation stems from the fact that **tbl_EContact** links to an instance of **tbl_AgentAssociation**, and the latter can exist only if a minimum of two of the three attributes *PersonID*, *OrganizationID*, and *AddressID* have been populated with non-zero values. Relaxing this requirement of having a minimum two out of three populated foreign keys in **tbl_AgentAssociation**, to the more liberal rule of either *PersonID* or *OrganizationID* being populated (regardless of *AddressID*), would remove this limitation.

- Although additional **AgentTypes** can be defined (*e.g.,* "Team"), they would need to be established in such a way that links to **tbl_AgentAssociation** are maintained logically. For example, if the third **AgentType** "Team" were established, then the *OrganizationID* foreign key of **tbl_AgentAssociation** might be redefined as "*TeamOrganizationID*" , indicating that it may be populated either with an *OrganizationID* or a *TeamID* (or the *AgentID* / *AssociatedAgentID* method).

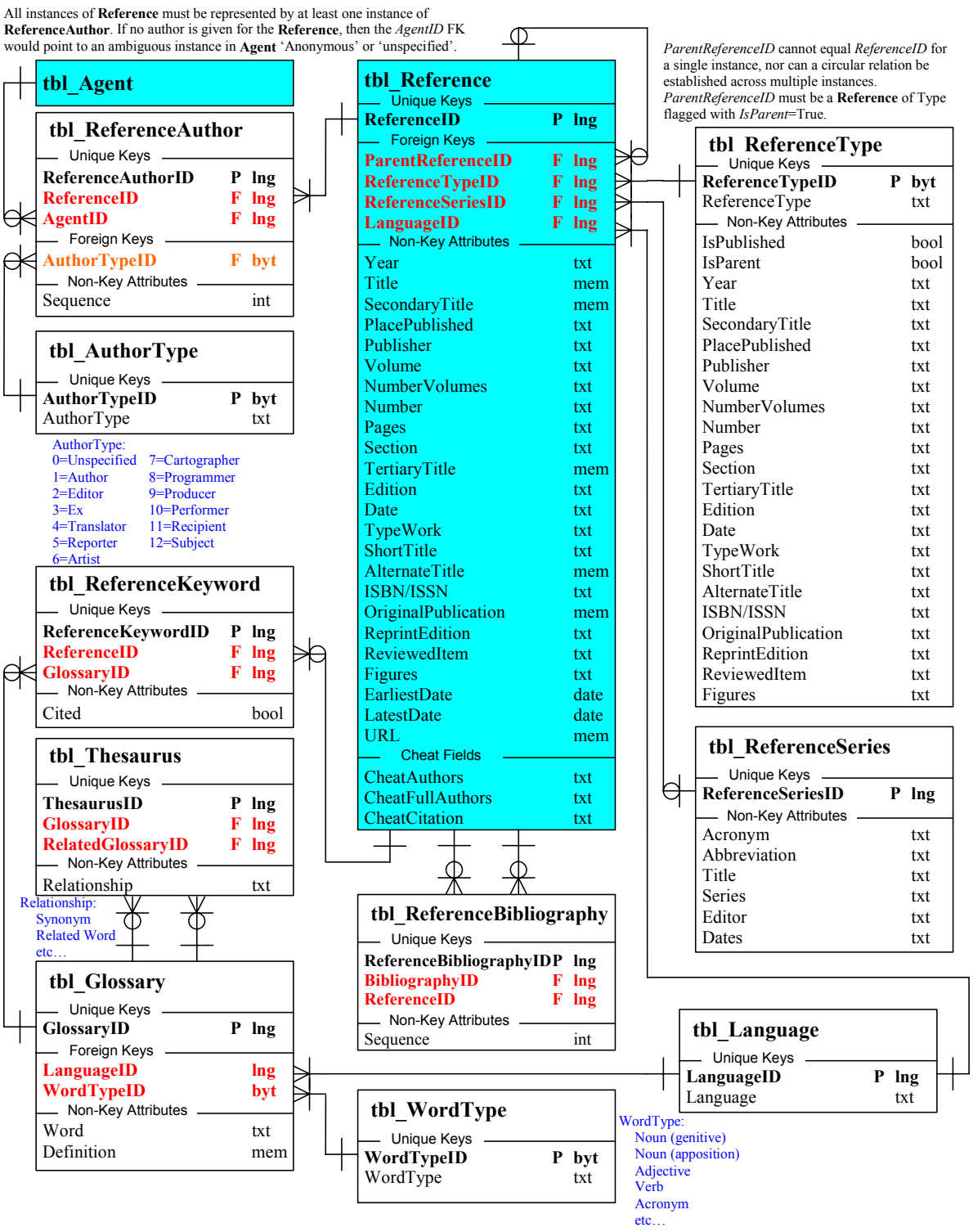

Figure 4. Reference physical data model.

References

The physical model for **Reference** data is represented in Figure 4. Whereas a "Reference" is most often thought of primarily in the context of a publication, the concept is here defined more broadly, in a way best described as a "Date-stamped instance of **Agent(s)**." All **References** must have as their source one or more **Agents** (**ReferenceAuthors**), and each instance of a **Reference** represents a statement of documented information by those **Agents** at a particular moment in time. Another way of expressing this is that a **Reference** may be created whenever any set of one or more **Agents** establishes or asserts some informational content (statement) at a certain point in time. All publications fall within this definition of "**Reference**," because all publications are drafted at the hand of one or more **Agents** (even if the **Agent** can only be identified as "Anonymous" or "Unspecified"), and are published at a particular point in time. Besides publications, however, there are other ways in which a set of one or more **Agents** may assert informational statements at a certain point in time. Familiar examples of unpublished **References** would include correspondence and other forms of personal communications (usually documented in the form of a letter, memo, or other printed but unpublished documentation), and specimen determinations (usually documented in the form of specimen labels or identification tags). All other attributes of **Reference** deal mainly with elements of information that identify the documentation and citation details about the **Reference** voucher, indexing by **ReferenceKeywords**, and cross-referencing **References** via the **ReferenceBibliography**.

tbl_Reference

The basic structure of **tbl_Reference** emulates the structure of EndNote® Version 7 bibliographic software (Anonymous, 2003), which has, to some extent, become an industry standard within academia. This structure was chosen to allow relatively easy transfer of **Reference** data between EndNote® 7 application software and the *Taxonomer* database application. Several aspects of this model expand upon the basic EndNote® 7 structure, primarily with regard to breaking certain data elements out into separate linked tables, but also in the form of extended data recording capabilities. These differences are discussed each in their relevant context below.

References may contain other **References** in a hierarchical fashion. A familiar example would be a book compiled by one set of **Agents** (*i.e.,* editors), which contains chapters authored by different sets of **Agents**. For the purposes of this data model, a more abstract and less traditional example is the 'Sub-Reference', which allows for the designation of less discretely defined portions of a reference to have different set of authors, or more precise page numbers or dates than the containing "parent" **Reference**. This capability is especially important for distinguishing text constituting original descriptions of taxonomic names from the containing **Reference**, in cases where the authorship of the taxon name is not identical to the authorship of the containing **Reference** (see "Taxa" section below for more elaboration). The hierarchy of **References**, when it exists, is tracked by the recursive *ParentReferenceID* linkage. As with **Organizations**, no **Reference** can be its own parent, and no multiple chain of **Reference→[Parent]Reference** can be circular.

Every **Reference** is classified according to its *ReferenceTypeID*, which is drawn from the **tbl_ReferenceType** table. The existing values of *ReferenceTypeID* and their corresponding text values of *ReferenceType* are shown in the first two (**bold**) columns of Table 2. The first 26 rows of Table 2 (corresponding to ID values 0-25) directly emulate the reference types defined in EndNote® 7. The last three (shaded) rows (which could potentially correspond to the three "unused" reference types of EndNote® 7) are defined in the context of the *Taxonomer* data model as 'Book Series', 'Determination' and 'Sub-Reference'. The 'Book Series' *ReferenceType* was added to accommodate citations of entire series, rather than individual volumes in a series. 'Determination' was added to accommodate the special group of unpublished **References** that represent taxonomic identifications of specimens. The 'Sub-Reference' *ReferenceType* is intended to represent a portion of another, more

Table 2. Reference Types and their use of data fields. Modified from EndNote® 7 template (Anonymous, 2003:366-371), re-printed with permission from Thomson ISI ResearchSoft. Shaded rows are not included in EndNote® 7, and are correspond to the three "unused" types available in EndNote® 7.

ID 0	Generic	Year	Title	Secondary Title	Place Published	Publisher	Volume	No. of Volumes	Number	Pages
1	**Journal Article**	Year	Title	-	-	-	Volume	-	Issue	Pages
2	**Book**	Year	Title	Series Title	City	Publisher	Volume	No. Vols.	Number	Pages
3	**Book Section**	Year	Title	Book Title	City	Publisher	Volume	No. Vols.	Number	Pages
4	**Manuscript**	Year	Title	Collection Title	City	-	-	-	Number	Pages
5	**Edited Book**	Year	Title	Series Title	City	Publisher	Volume	No. Vols.	Number	Pages
6	**Magazine Article**	Year	Title	-	-	-	Volume	-	Issue	Pages
7	**Newspaper Article**	Year	Title	-	City	-	-	-	-	Pages
8	**Conference Proceedings**	Year	Title	Conf. Name	Conf. Loc.	Publisher	Volume	No. Vols.	-	Pages
9	**Thesis**	Year	Title	Academic Dept.	City	University	-	-	-	Pages
10	**Report**	Year	Title	-	City	Institution	-	-	-	Pages
11	**Personal Communication**	Year	Title	-	City	Publisher	-	-	-	-
12	**Computer Program**	Year	Title	-	City	Publisher	Version	-	-	-
13	**Electronic Source**	Year	Title	-	-	Publisher	Access Year	Extent	Acc. Date	-
14	**Audiovisual Material**	Year	Title	Collection Title	City	Publisher	-	-	Number	-
15	**Film or Broadcast**	Year	Title	Series Title	City	Distributor	-	-	-	Length
16	**Artwork**	Year	Title	-	City	Publisher	-	-	-	-
17	**Map**	Year	Title	-	City	Publisher	-	-	-	Scale
18	**Patent**	Year	Title	Published Source	Country	Assignee	Volume	No. Vols.	Issue	Pages
19	**Hearing**	Year	Title	Committee	City	Publisher	-	-	Doc. No.	Pages
20	**Bill**	Year	Title	Code	-	-	Code Volume	-	Bill No.	Pages
21	**Statute**	Year	Title	Code	-	-	Code Number	-	Law No.	1st Pg.
22	**Case**	Year	Title	-	-	Court	Reporter Vol.	-	-	-
23	**Figure**	Year	Title	Source Program	-	-	-	-	-	-
24	**Chart or Table**	Year	Title	Source Program	-	-	-	-	-	-
25	**Equation**	Year	Title	Source Program	-	-	Volume	-	Number	-
26	**Book Series**	Year	Title	-	City	Publisher	-	No. Vols.	-	Pages
27	**Determination**	Year	Title	-	-	Institution	-	-	-	-
28	**Sub-Reference**	Year	Title	-	-	-	-	-	-	Pages

Table 2. (Continued) Reference Types and their use of data fields.

ID	Generic	Section	Tertiary Title	Edition	Date	Type of Work	Short Title	Alternate Title	ISBN ISSN	Figures
0	**Generic**	**Section**	**Tertiary Title**	**Edition**	**Date**	**Type of Work**	**Short Title**	**Alternate Title**	**ISBN ISSN**	**Figures**
1	**Journal Article**	-	-	-	Date	-	Short Title	Alt. Jour.	-	Figures
2	**Book**	-	-	Edition	Date	-	Short Title	-	ISBN	Figures
3	**Book Section**	-	Ser. Title	Edition	Date	-	Short Title	-	ISBN	Figures
4	**Manuscript**	-	-	Edition	Date	Type Work	Short Title	-	-	Figures
5	**Edited Book**	-	-	Edition	Date	-	Short Title	-	ISBN	Figures
6	**Magazine Article**	-	-	-	Date	-	Short Title	-	-	Figures
7	**Newspaper Article**	Section	-	Edition	Date	Type Art.	Short Title	-	-	Figures
8	**Conference Proceedings**	-	Ser. Title	Edition	Date	-	Short Title	-	ISBN	Figures
9	**Thesis**	-	-	-	Date	Thesis Type	Short Title	-	-	Figures
10	**Report**	-	-	-	Date	Type Work	Short Title	-	Rpt. No.	Figures
11	**Personal Communication**	-	-	-	Date	Type Work	Short Title	-	-	-
12	**Computer Program**	-	-	Platform	Date	Type Work	Short Title	-	-	-
13	**Electronic Source**	-	-	Edition	Date	Medium	Short Title	-	-	-
14	**Audiovisual Material**	-	-	-	Date	Type Work	Short Title	-	-	-
15	**Film or Broadcast**	-	-	-	Date	Medium	Short Title	-	ISBN	-
16	**Artwork**	-	-	-	Date	Type Work	Short Title	-	-	-
17	**Map**	-	-	Edition	Date	Type Work	Short Title	-	-	-
18	**Patent**	-	-	-	Date	-	Short Title	-	Pat. No.	Figures
19	**Hearing**	-	Leg. Body	Session	Date	-	Short Title	-	-	-
20	**Bill**	Section	Leg. Body	Session	Date	-	Short Title	-	-	-
21	**Statute**	Section	-	Session	Date	-	Short Title	-	-	-
22	**Case**	-	-	-	Date	-	Abb. Case	-	-	-
23	**Figure**	-	-	-	Date	-	-	-	-	-
24	**Chart or Table**	-	-	-	Date	-	-	-	-	-
25	**Equation**	-	-	-	Date		-	-	-	-
26	**Book Series**	-	-	Edition	Date	-	-	-	ISBN	Figures
27	**Determination**	-	-	-	Date	-	-	-	-	-
28	**Sub-Reference**	-	-	-	Date	-	-	-	-	Figures

encompassing **Reference** (excluding cases that can be assigned to the 'Book Section' *ReferenceType*), primarily to accommodate assigning appropriate authorship to taxon names (when such authorship differs from the encompassing Reference – see discussion above, and below in the "Taxa" section).

The top row of Table 2 (*ReferenceTypeID*= 0; *ReferenceType*='Generic') correspond to the generic fields as used in EndNote® 7 to store reference data (with a few exceptions, described below). These columns correspond to most of the Non-Key Attributes shown in Figure 4 for **tbl_Reference** and **tbl_ReferenceType**. Table 2 serves as a matrix to indicate how each of these non-key attributes of **tbl_Reference** are used to store information, according to the value of *ReferenceTypeID* for each instance. For example, if *ReferenceTypeID*=1 ('Journal Article'), then the *Year*, *Title*, *Volume*, *Number*, *Pages*, *Date*, etc. fields of **tbl_Reference** for that instance are used for storing data as indicated in Table 2; and the other fields (*i.e.,* those represented by a '-' in Table 2) are not used for that particular **tbl_Reference** instance. The corresponding attributes in **tbl_ReferenceType** are intended to store metadata used by the *Taxonomer* software application, and will not be described here, except to explain that they were given the same attribute names in order to simplify coding of the *Taxonomer* application. More information on the specific use and purpose of the various attributes included in Table 2 can be obtained from Anonymous, 2003.

The last column in Table 2 ('Figures') does not exist in EndNote® 7, and is here assumed to occupy the 'Custom 1' field provided in EndNote® 7. This attribute of **tbl_Reference** is intended to allow documentation of figures and plates, which may be important for taxonomic purposes.

Conversely, several additional fields used by EndNote® 7 are not included in Table 2. Four of these ('Author', 'Secondary Author', 'Tertiary Author' and 'Subsidiary Author') are accommodated by **tbl_ReferenceAuthor**, as described below. EndNote® 7 allows for 6 'Custom' fields; the first of which is used for *Figures* as described above, and two additional 'Custom' fields are used by *Taxonomer* for the **tbl_Reference** attributes *EarliestDate* and *LatestDate*. These two date fields are used by *Taxonomer* to establish the narrowest possible range in time when the **Reference** was published (whereas the *Date* attribute allows for a text description of when the **Reference** was published). The other three 'Custom' fields in EndNote® 7 are as yet unassigned in the *Taxonomer* model, but could potentially be used to store values of the Foreign Key attributes *ParentReferenceID*, *ReferenceSeriesID*, and *LanguageID*, if those values need to be preserved during a data export to EndNote® 7.

The 'Accession Number', 'Call Number', and 'Label' fields in EndNote® 7 are accommodated by the **tbl_CodeNumber** and **tbl_CodeNumberSeries** portions of the *Taxonomer* data model. Similarly, the 'Abstract' and 'Notes' fields are accommodated by **tbl_Excerpt** and **tbl_Comment**. All four of these tables are described in the "*General Data Management*" section later in this article. 'Keywords' of EndNote® 7 are linked to **tbl_Reference** from **tbl_Glossary** via **tbl_ReferenceKeyword** (as described below). 'Author Address' is accommodated by **tbl_AgentAssociation** and related tables, described earlier in the "**Agents**" section; and "Image" and "Caption" of EndNote® 7 are dealt with in a different part of the *Taxonomer* model, not described herein. The 'URL' field of EndNote® 7 is directly mapped to the *URL* attribute of **tbl_Reference**. It is not included in Table 2 (and not among the attributes of tbl_ReferenceType) because its purpose is the same regardless of the value of ReferenceTypeID: to store a standard internet URL address, when one is available. Two additional boolean attributes of **tbl_ReferenceType** – *IsPublished* and *IsParent* – are used by the *Taxonomer* application to indicate which **ReferenceTypes** are published, and which can serve as a "parent" **Reference** to another **Reference** (respectively).

Two additional Foreign Key attributes of **tbl_Reference** remain to be described. Depending on which *ReferenceTypeID* is

selected for the particular **Reference** instance, there may be a link to the **tbl_ReferenceSeries** via the *ReferenceSeriesID* Foreign Key. The reference types that can be linked to a **ReferenceSeries** include 'Generic', 'Book', 'Book Section', 'Conference Proceedings', 'Edited Book', 'Journal', 'Magazine Article', and 'Newspaper Article'. Attributes of **tbl_ReferenceSeries** are indicated in Figure 4, and are not as yet rigidly defined. The use of **tbl_ReferenceSeries** leads to several of the deviations from standard EndNote® 7 field usage, compared with what is presented in Table 2 (*i.e.,* EndNote® 7 uses the 'Secondary Title' field to store the same information as the **tbl_ReferenceSeries** link provides in *Taxonomer*).

Finally, each **Reference** instance may be associated with the **tbl_Language** table, via the *LanguageID* Foreign Key, to indicate which language the **Reference** was primarily written in.

There are three "Cheat" fields within **tbl_Reference**: *CheatAuthors*, *CheatFullAuthors*, and *CheatCitation*. *CheatAuthors* is used to store formatted single- and dual-author *FamilyNames*, or first-author *FamilyName* plus "et. al" for multi-authored **References**. *CheatFullAuthors* is used to store formatted author names as they generally appear in bibliographies – *FamilyName* and initials of *GivenNames* for each individual author. *CheatCitation* is a concatenation of *CheatAuthors* and *Year* field. All three of these "Cheat" fields are used to enhance output performance.

tbl_ReferenceAuthor

Every **Reference** instance must be linked to one or more **Agent(s)** representing the author(s) of the **Reference**, via the **tbl_ReferenceAuthor** table. In cases where the specific author is not known, a link is established to an ambiguous instance of **Agent** representing 'Anonymous' or 'Unspecified'. The important point here is that a **Reference** is *defined* in the context of its authoring **Agent(s)**; hence the requirement for at least one instance of **tbl_ReferenceAuthor** for each instance of **tbl_Reference**.

The *AuthorTypeID* Foreign Key to tbl_AuthorType denotes the nature of the relationship between the **Agent** and the **Reference** (defined values displayed in blue text in Figure 4). In most cases, Agents serve the role of 'Author' or 'Editor'. Other values of *AuthorType* are mostly self-evident, but three warrant elaboration. *AuthorType* 'Ex' (*AuthorTypeID*=3) is used to flag those specific authors who are authors of taxon names, but not authors of the **Reference** itself. For example, suppose a **Reference** is linked to **ReferenceAuthors** Smith, Jones, and Johnson, with Johnson indicated by *AuthorTypeID*=3. Taxon names linked to this **Reference** (see "Taxa" section) would treat the authorship of that Protonym as "Smith and Jones (ex Johnson)." Additionally, if this **Reference** happens to be of type 'Sub-Reference', which itself is included within a publication authored by (for example) Jones and Wilder, then the authorship for the taxon name would be interpreted as "Smith and Jones (ex Johnson) in Jones and Wilder." *AuthorType* 'Recipient' (*AuthorTypeID*=11) is used to denote who the recipient of a Personal Communication **Reference** was. Finally, *AuthorType* 'Subject' (*AuthorTypeID*=12) is included for references that include biographical information, to allow indexing of who the biographical information pertains to. The *Sequence* attribute is used to establish the sequence of authors for multi-authored **References**. The value of this field is only meaningful within the context of a set of authors that are of the same *AuthorType*.

tbl_ReferenceBibliography

The **tbl_ReferenceBibliography** table is used to record which **References** (*BibliographyID*) cite which other references (*ReferenceID*) in their bibliography (or elsewhere). This can be useful in deciphering implied taxonomic concepts, to indicate whether or not one **Reference** explicitly had access to another **Reference** at the time a taxonomic concept was formulated. The *Sequence* field is used to establish the sequence of cited **References**, as they appear in the citing **Reference**. This table is useful both for constructing bibliographies of **References**, and also for creating a "Citation Index" for **References**.

tbl_Glossary
A generic system of defining words is established via the **tbl_Glossary** table. Each *Word* exists in the context of a *Language* (linked from **tbl_Language** via the *LanguageID* Foreign Key), and is assigned a *WordType* (linked from **tbl_WordType** via the *WordTypeID* Foreign Key – examples of *WordType* shown in blue text in Figure 4). A short *Definition* is provided for each *Word*.

Individual words can be cross-referenced to other words via the **tbl_Thesaurus** table. The nature of the relationship between the two words (*e.g.,* 'Synonym', 'Related Word', etc.) is indicated in the *Relationship* field. Such relationships are not automatically treated as symmetrical, so in the case of a symmetrical relationship (*e.g.,* 'Synonym'), two instances are required in the **tbl_Thesaurus** table. Future versions of this data model may define a **tbl_RelationshipType** table as a separate linked entity, allowing additional attributes for each relationship type (*e.g., IsSymmetrical*, etc.).

Individual instances of **tbl_Glossary** are linked to instances of **tbl_Reference** via the **tbl_ReferenceKeyword** table. If the indicated keyword was designated in the linked **Reference** itself, then the *Cited* field is set to 'True'. Otherwise, it is assumed that Keyword assignment was created by the database user.

Limitations

- The general limitation of the whole **Reference** structure stems from its foundation in the EndNote® 7 model. A somewhat denormalized flat **tbl_Reference** structure (as opposed to establishing multiple subtypes of **References**) is taken as a compromise to maintain simplicity of import and export capability with EndNote® 7 and other bibliographic citation data standards.

Taxa

As summarized in the "Introduction" section of this article, there is a well-acknowledged subtle but important distinction between a "Taxon Name" and a "Taxon Concept" (*e.g.*, Berendson, 1995; Le Renard, 2000; Geoffroy & Berendsohn, 2003). A taxonomic name is an objective entity, and exists (and is defined) in the form of printed text. The name itself is a string of text characters (which can, under certain circumstances, change in spelling), and is objectively linked to the biological world via a properly designated type specimen (a more subjective link between a name and the biological world is often represented in the form of characters that define a taxonomic concept). Most attributes of each name (*e.g.*, publication date, original spelling, authorship, etc.) are usually unambiguous, and not open to subjective interpretation (except in a few specific cases). New names are created in accordance with strict and detailed rules of nomenclature; *i.e.,* ICBN (Greuter et. al, 2000); ICZN (ICZN, 1999); ICNB (Lapage et al. 1992); LBSN (Euzéby, 2003); ICVCN (Francki et al. 1990; Murphy et al. 1995; van Regenmortel et al. 2000); and ICNCP (Trehane et al., 1995). For the most part, information pertaining to taxonomic names is objective in nature. Taxon Names can be thought of as the individual "words" comprising the dictionary of the diversity of life.

A "Taxon Concept," on the other hand, is a purely abstract, subjective construct that ultimately exist only in the mind of a taxonomist (see Geoffroy & Berendsohn, 2003). Concepts are much less discretely defined entities, the creation or establishment of which are not governed by Codes of nomenclature, and whose attributes are considerably more ambiguous than those of a taxon name. Whereas a Taxon Name is generally anchored to the biological world via a single specimen, a Taxon Concept is intended to circumscribe a large (potentially vast) collection of individual organisms, living, dead, and yet-to-be-born, all of which share a level of common ancestry (kinship) and morphological/genetic similarity so as to be regarded as belonging to the same taxon (*e.g.,* species). Taxon Concepts can be thought of as the definitions of those Taxon-Name "words" that comprise the dictionary of the diversity of life.

Unlike the definitions of most words in a conventional dictionary, however, the mapping of Taxon Concepts to Taxon Names has been far from consistent among practitioners of taxonomy. Some taxono-

mists tend to prefer more generalized concepts (definitions), which leads to more of the names (words) being synonymous with other names (words). Others prefer more specific concepts (definitions), thereby maintaining distinctions between different names (words). The basic problem is that most published and unpublished documentation about taxa use only the names (words), without necessarily including explicit details about how those names are circumscribed (defined). Thus, the task at hand is to find a way to consistently and objectively map Names (words) to their various respective implied Concepts (definitions).

In order to map the Names to the Concepts, the first step is to apply an unambiguous "handle" on each Name and Concept, and then build an index to map the Name handles to the Concept handles. The easiest and most straightforward way to put a handle on a taxon name is to attach that handle to the Basionym of the name. Although the word "Basionym" is more frequently used in botanical contexts than in zoological contexts, the basic concept applies equally to both (and is becoming more commonly used in zoological contexts). The Basionym can be thought of as a pointer to a name's original description – the moment when a string of text characters becomes legitimately available for taxonomic use (in accordance with the various codes of nomenclature) – and therefore as the handle to a name. Another term used frequently in botanical contexts is "Protologue", which represents the set of elements constituting an original description of a name. After much contemplation and discussion with colleagues, I have decided that the confusion that may result from attempting to use either one of these pre-existing terms to represent a concept that is not really quite either, would be greater than the confusion of introducing a new term that is intended to represent certain elements of both. For a number of reasons, I have chosen to use the word "Protonym" instead of either "Basionym" or "Protologue" for the *Taxonomer* data model (see further discussion in the "Limitations" sub-section below).

Hence, the use of **Protonym** (hereinafter shown in **bold** text) in this data model serves as a common linkage between the original presentation of a taxonomic name, and subsequent use of that same name in (potentially) different Concept contexts.

The textual representation of a **Protonym** takes the form of:

Name OriginalAuthor(s), OriginalYear

(although "OriginalYear" is often excluded for botanical names).

As described in detail within the "References" section of this article, a "**Reference**" is generally defined as a documented instance of "date-stamped Author(s)," which can also be read as "Author(s), Year." Thus, the most convenient handle for a **Protonym** can be thought of as:

Name OriginalReference

The method for applying a handle to a Taxon Concept is less consistent, and not often as unambiguous as applying a handle on a Name. However, one common approach is to cite a name in the context of another **Reference**, in the form of:

Name OriginalReference sensu *OtherReference*

(Geoffroy & Berendsohn, 2003, use the abbreviation "sec." instead of *sensu*).

In the case of the Taxon Concept associated with the **Protonym** itself, the representation would be:

Name OriginalReference sensu *OriginalReference*

Reducing this one step further, "*Name OriginalReference*" can be substituted with "**Protonym**" (as defined above), and the Concept can then be thought of as:

Protonym sensu *Reference*

(where "**Reference**" is either "*OriginalReference*" in the case of the Concept attached to the original name creation, or "*OtherReference*" in all other cases). Thus, whereas the handle for a Taxon Name can be thought of as the **Protonym**, the handle for a Taxon Concept can be thought of as the intersection of a **Protonym** and a **Reference**.

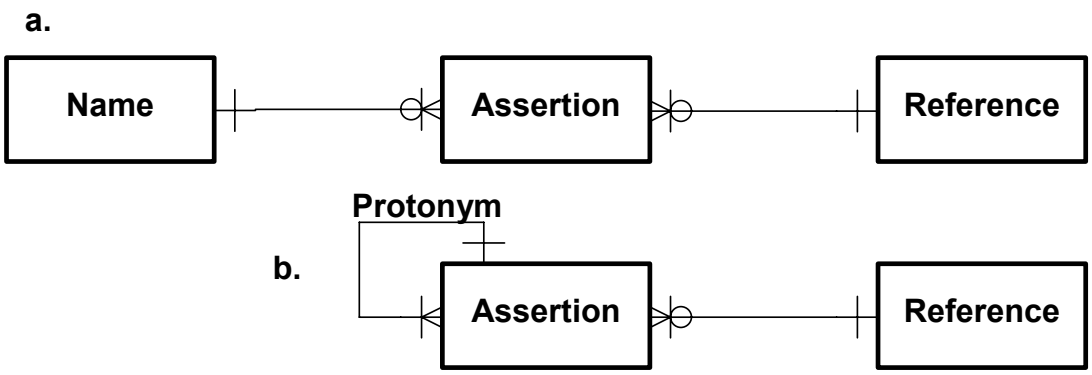

Figure 5. Conceptual representation of Names, Assertions, and References. a) traditional view; b) perspective presented herein.

I have used the term **Assertion** to represent this **Protonym**-**Reference** intersection, which has previously been diagramed (*e.g.,* Taswell & Peet, 2000) as in Figure 5a. This diagram implies a "One to ***Zero***-to-Many" relationship between **Names** and **Assertions**. However, a Name cannot exist without at least one **Assertion** – the **Assertion** in which the Name was first proposed (the original description). Therefore, the relationship between Names and **Assertions** should be "One to ***One***-to-Many." Taking this one step further, a Name in the context of the Reference that provided its original description has been defined above as the **Protonym**. Because a **Protonym** exists in the context of the **Reference** that originally established it, a **Protonym** can itself be represented as an **Assertion** (*i.e.,* "*Name OriginalReference sensu OriginalReference*"). Given that a Name cannot exist without its **Protonym**, the relationship between a "Name" (**Protonym**) and an **Assertion** becomes recursive, as shown in Figure 5b. Therefore, the conceptual handle for the name and the handle for the concept are one and the same, with the former being a special-case subtype of the latter.

Stated another way, all Names initially become available through the **Reference** that constitutes its original description (the **Protonym**). These original descriptions did themselves assert a Taxon Concept to be applied to the Name as proposed, and therefore also represent **Assertions**. As a subset of the broader scope of **Assertions** (which include potentially many **Reference** treatments of names other than the original description), **Protonyms** represent the ideal linkage point to joint multiple **Assertions** based on the same original name. Thus, whereas all **Assertions** represent the handle to a Taxon Concept, the subset of **Assertions** constituting **Protonyms** represent dual-purpose handles to both Taxon Concepts *and* Taxon Names.

It is worth clarifying at this point that, although the handle to a taxon concept can be thought of as an instance of an **Assertion**; not all **Assertions** necessarily represent implied Taxon Concepts. For example, one form of publication is a "Type Catalog," wherein all type specimens in a Museum's collection are listed according to the Names that they typify. In such publications, the authors will list taxon names (generally as unaltered **Protonyms** in this case), and hence establish an intersection between a **Reference** (the type catalog publication itself) and a **Protonym** – but without necessarily implying a Taxon Concept to go along with that name (*i.e.,* literally only the type specimens are asserted in such cases, without any implications about the scope of non-type individual kin organisms to be included within a Taxon Concept represented by the name). In such cases, an instance of an **Assertion** exists without an implied Taxon Concept. For this reason, an **Assertion** should be regarded as representing a "Potential Taxon [Concept]" (*sensu* Berendsohn, 1995). In the majority of **Name-Reference** intersections (**Assertions**), however, the author(s) of the **Reference**

had a Taxon Concept circumscription in mind when invoking the Taxon Name, even if the scope of that circumscription is not defined (or even alluded to) within the **Reference** itself. Thus, in the vast majority of cases, **Assertion** instances can be used as a direct "handle" to an implied taxon concept circumscription (which, in many cases, will be precisely identical to the circumscriptions implied by many other **Assertions** for a given Taxon Name). Because the definition of a **Reference** herein is not restricted to publications, it can be said that all Concepts that map to taxon names can be identified by an **Assertion**, whether or not they appear in published form.

Before describing the "Taxa" components of the data model in detail, it is worthwhile to outline alternative distinct "resolutions" at which circumscription scopes are often defined:

Name-Resolution Circumscription Definitions

This is the coarsest, and most often-used resolution of circumscription scope expression in published taxonomic references. Such circumscriptions are defined merely by treating taxon names as either valid, or as junior synonyms of other taxon names. Because taxon names are anchored to the biological world via type specimens, this method of defining circumscriptions can be thought of in a sense as Specimen-resolution circumscription definitions, except limiting it to only those particular specimens that represent primary types of taxon names. To list Taxon Name 'B' as a junior synonym of Taxon Name 'A', is to assert that "the primary type specimen of Taxon Name 'A' and the primary type specimen of Taxon Name 'B' share close enough kinship to each other that they should be regarded as belonging to the same taxon circumscription" (in this case, with the relevant Code bestowing the name 'A' with nomenclatural priority over the name 'B'). Conversely, to list Taxon Name 'B' as valid and distinct from Taxon Name 'A', is to assert that "the primary type specimen of Taxon Name 'A' and the primary type specimen of Taxon Name 'B' are sufficiently distant in kinship to each other that they should be regarded as belonging to different taxon circumscriptions." In this way, the full scope of the implied circumscription is represented by the set of **Assertions** within a **Reference** that include a Name that is treated as valid, plus all **Assertions** of Names that are treated as junior synonyms of that valid Name (the handle on the **Assertion** being maintained as the one represented by the Name treated as valid).

The primary weakness of this form of circumscription definition is as follows:

When a **Reference** does not treat all relevant Names that are available at the time the **Reference** is established (*e.g.,* when not all potentially valid taxa are treated, or not all potentially relevant synonyms are assigned to Names that are treated as valid), then the circumscription definitions within the context of the **Reference** are incomplete.

Even when a **Reference** does treat all relevant Names available at the time the **Reference** is established, the **Reference** may be later rendered incomplete by subsequent descriptions of new Names for closely-related taxa.

Using only Name-level circumscription definitions (*i.e.,* without elaborating the character-based criteria used to delineate different circumscriptions), greatly inhibits the ability to secondarily assign individual non-type specimens to these circumscriptions.

These weaknesses notwithstanding, Name-resolution circumscription definitions represent the bulk of documented taxonomic information (inclusive of all References citing taxonomic names with lists of synonyms), and therefore serve as an ideal "core" information content base around which the foundation of a data model should be built.

Specimen-Resolution Circumscription Definitions

The most fundamental (and finest) resolution at which circumscriptions are mapped is via individual specimens (beyond the limited scope of primary type specimens). The source **Reference** for corresponding **Assertions** can either be in the form of a publica-

tion (as when a published **Reference** lists museum specimen catalog numbers under a particular Taxon Name), or in the form of an unpublished "Determination"-type **Reference** (*i.e.,* identification labels on the actual museum specimens themselves).

Other Circumscription Definition Resolutions
It could be argued that "Character-Resolution Circumscription Definitions" represent a another resolution at which circumscriptions can be defined. For reasons not elaborated herein, I see this as a fundamentally different approach to mapping the scope of taxon circumscriptions, because it transcends the individual organism (considered to be the basic unit of a taxon). While this question is certainly ripe for discussion, it goes beyond the intended scope of this article.

Also, circumscriptions are sometimes defined in terms of populations of organisms. This resolution of circumscription definition represents cases where a **Reference** ascribes specific populations to Taxon Names, thereby extending the resolution of circumscription boundary delineation beyond the relatively course type-specimen anchor points, but not as precise as specimen-resolution definitions. This kind of circumscription definition usually takes the form of biogeographic treatments (*i.e.,* mapping taxon names directly to geographic regions, bypassing the more fundamental connection between names and locations via specimens).

The core "Taxa" data model represented here, illustrated in Figure 6, is intended to directly document "Name-Resolution" circumscription definitions, while also providing a tangible "handle" to a circumscription (*i.e.,* an **Assertion** instance) that can be more precisely defined at higher resolution (*e.g.,* specimen resolution) via additional "modules" of data entities (as described in the next section).

tbl_Assertion
The central "anchor" entity of the taxon portion of this data model is the **Assertion**. As previously stated, an **Assertion** is defined as the intersection of a **Protonym** and a **Reference**, as indicated by the Foreign Keys, *ProtonymID* and *ReferenceID*. Because **Protonyms** themselves represent **Assertions** (*sensu* the original authors of the **Protonym**), it would be possible to represent the relationship of *ProtonymID* to *AssertionID* via a direct recursive link. However, because certain attributes apply only to **Protonyms** and not all **Assertions** (*e.g.,* nomenclatural attributes such as *Availability* in the case of names governed by Codes, and "Type Species" and *Gender* in the case of generic-level names – described in more detail along with other **Protonym** attributes below), and also for reasons of enforcing business rules and improving performance of certain query operations, the table **tbl_Protonym** is represented as a subtype of **tbl_Assertion**. The recursive linkage between any particular **Assertion** instance and its associated **Protonym** is made via the **tbl_Protonym** subtype; first from the *ProtonymID* Foreign Key field of **tbl_Assertion** to the *ProtonymID* Primary Key of the subtype **tbl_Protonym**, and then recursively back to the **tbl_Assertion** table via the One-to-One subtype/supertype link to *AssertionID*. The domain of **Assertion** instances that are represented by instances in **tbl_Protonym** are (by definition) those specific instances of **tbl_Assertion** where *AssertionID*= *ProtonymID*.

The *ReferenceID* Foreign Key of **tbl_Assertion** is a straightforward linkage to the **Reference** in which the **Assertion** is made. A fundamental component of the *Taxonomer* data model is that Taxon Name authorship is derived directly from the authorship of the **Reference** to which the corresponding **Protonym** is linked. This straightforward authorship derivation has often been avoided in other similar data models, because the authorship of a Taxon Name is not necessarily identical to the authorship of **Reference** in which the Taxon Name was originally described. Rather than establish two separate relationships between **Protonyms** and author **Agents** (one indirectly via the link to the original description **Reference**, and one representing the taxonomic authors of the Name itself), I have instead established the concept of a "Sub-Reference" (see discussion in the previous section on **References**).

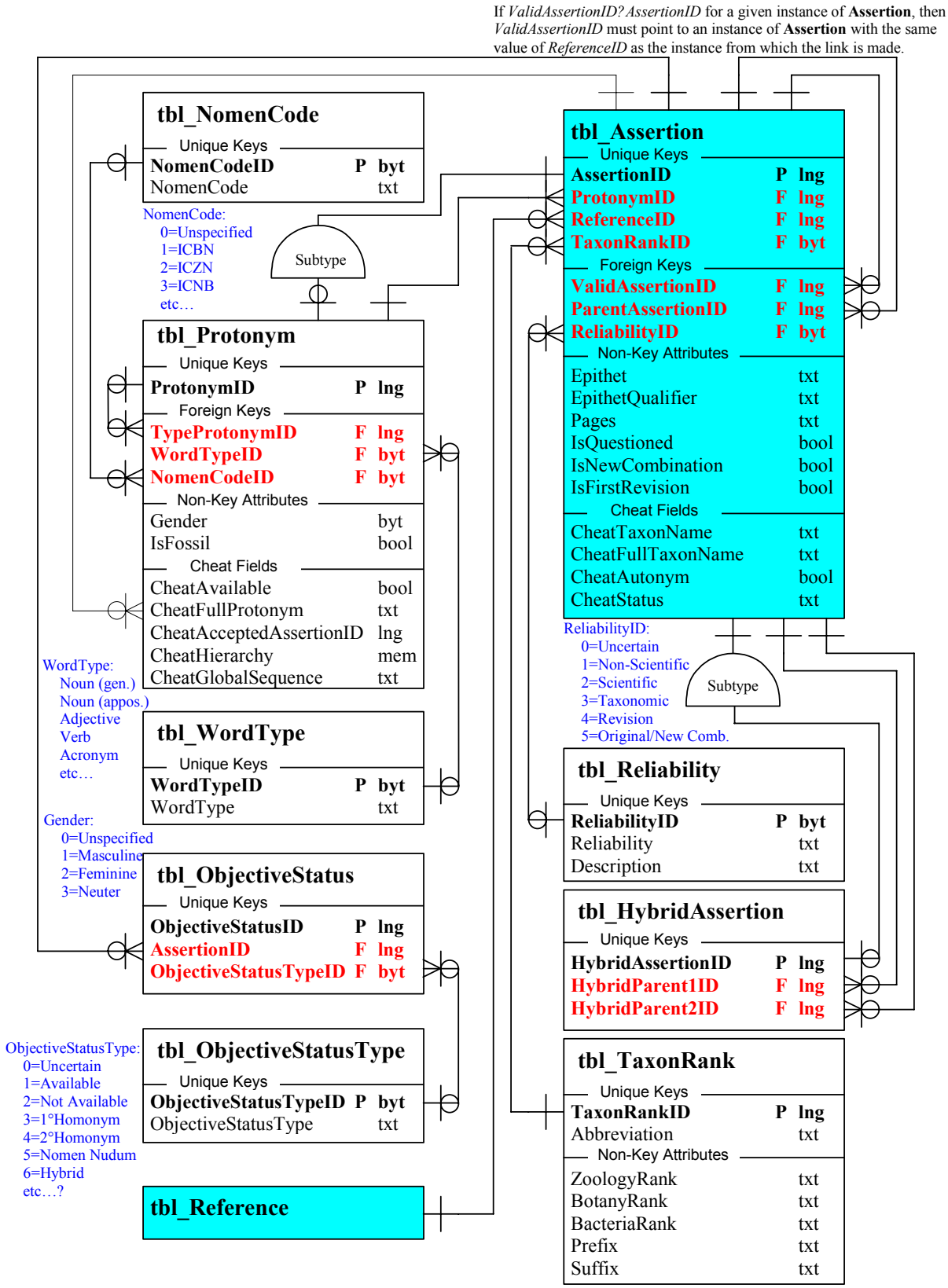

Figure 6. Taxonomic physical data model.

As described above in the "References" section, the **ReferenceType** "Sub-Reference" was established and defined to represent a sub-section of another **Reference** (other than more traditional cases of Parent-Child **References**, such as Chapters in a Book). A Sub-Reference has its own set of **ReferenceAuthors**, and its own publication *Date*, which may or may not be the same as the corresponding values of **ReferenceAuthors** and/or *Date* for the Parent **Reference**. Thus, in cases where the authors (or *Date*) of a Taxon Name are not identical to the authors (or *Date*) of the **Reference** in which the Name was originally described, a "Sub-Reference" is created (linked to the appropriate Parent **Reference** via *ParentReferenceID*) with the appropriate set of **ReferenceAuthors** and *Date*, and the **Protonym** instance for the Taxon Name is linked to that Sub-Reference. This solution to the "Taxon Name authorship problem" is logically appropriate, because technically the authors of a Taxon Name are deemed to be the authors of the portion of the published work that constitutes the original description of the Name. Hence, the description of the Taxon Name can be seen to represent a sub-section of a **Reference** unto itself – a **Reference** within a **Reference**. It should be noted that business rules require that all Sub-References be established as a child of another Reference instance (which itself is not a Sub-Reference), via the *ParentReferenceID* link.

In the vast majority of cases, the two Foreign Keys *ProtonymID* and *ReferenceID* would (by themselves) uniquely identify every **Assertion** instance. However, in the special case of autonyms, representing nominotypical taxa (*e.g.,* the subfamily Chaetodontinae within the family Chaetodontidae; or the subgenus *Chaetodon* (*Chaetodon*); or the subspecies *Chaetodon unimaculatus unimaculatus*) represent cases where a single **Protonym** can be used within a single **Reference** as representing two distinct Taxon Concepts. For this reason, the *TaxonRankID* Foreign Key, which identifies the exact taxonomic rank at which the **Protonym** is used within the **Reference**, must also be included among the uniquely-identifying attributes of a particular **Assertion** instance.

The *TaxonRankID* Foreign Key establishes a link to the **tbl_TaxonRank** table. Each record of this table represents a taxonomic rank that is in current use, or may have been in historical use, in any of the three major taxonomic disciplines (Botanical, Zoological, or Microbial). The reason for including ranks that are no longer in current use is that the Assertion table is intended to track all historical uses of Taxon Names, at whatever rank they may have been assigned to. Unfortunately, the different ranks, and the names assigned to each rank, are not universally established for all of biology. For this reason, three separate attributes (*ZoologyRank*, *BotanyRank*, and *BacteriaRank*) are needed to record the rank label used within each of the three corresponding major Codes of nomenclature. The contents of **tbl_TaxonRank** are shown in Table 3. When a value for *ZoologyRank*, *BotanyRank*, or *BacteriaRank* is empty, the corresponding **TaxonRank** is believed to have never been used within the respective branch of nomenclature (further investigation should allow the additional elimination of certain ranks from certain branches; particular Bacterial). Additional attributes of this sort could be established for other rank-based Codes of nomenclature (*e.g.,* LBSN, ICVCN, ICNCP), but as yet have not been added to the *Taxonomer* model. For convenience, the corresponding values used in the "rank_ID" field of the ITIS data model (ITIS, 2003), when they exist, are provided in the left-most column of Table 3.

Unlike most surrogate Primary Key fields of tables within the *Taxonomer* data model, *TaxonRankID* does, in fact, contain information. First, its value conveys the sequence of ranks within the established hierarchy (thereby allowing the enforcement the business rule that prevents establishing *ParentAssertionID* links to **Assertions** of equal or lower rank). Second, the numbers are assigned within clusters of ten, such that the first digit of each two-digit *TaxonRankID* represents the major rank grouping (except in the case of *TaxonRankID*=0, which is consistent with the use of 0 as "Unspecified" elsewhere in the data model). For example, values less than 10 are above the rank of "Kingdom"; values 10-19 are reserved for ranks within the "Kingdom" group; values

Table 3. Contents of tbl_TaxonRank, with corresponding ITIS "rank_ID" values.

TaxonRankID	ZoologyRank	BotanyRank	BacteriaRank	Abbreviation	Prefix	Suffix	ITIS
0	<Unspecified>	<Unspecified>	<Unspecified>	UNK			
05	Domain	Domain	Domain	DOM			
08	Superkingdom	Superkingdom	Superkingdom	SPK			
10	Kingdom	Kingdom	Kingdom	KGD			10
13	Subkingdom	Subkingdom	Subkingdom	SBK			
18	Superphylum	Superphylum	Superphylum	SPP			
20	Phylum	Division	Phylum	PHY			30
23	Subphylum	Subdivision	Subphylum	SBP			40
28	Superclass	Superclass	Superclass	SPC			50
29	Grade	Grade	Grade	GRD			
30	Class	Class	Class	CLS			60
33	Subclass	Subclass	Subclass	SBC			70
34	Infraclass	Infraclass	Infraclass	INC			80
35	Division			DIV			
36	Subdivision			SBD			
37	Infradivision	Infradivision	Infradivision	IND			
38	Superorder	Superorder	Superorder	SPO			90
40	Order	Order	Order	ORD			100
43	Suborder	Suborder	Suborder	SBO			110
44	Infraorder	Infraorder	Infraorder	INO			120
47	Section[Order]	Section[Order]	Section[Order]	SEC			
48	Superfamily	Superfamily	Superfamily	SPF			130
50	Family	Family	Family	FAM			140
53	Subfamily	Subfamily	Subfamily	SBF			150
55	Tribe	Tribe	Tribe	TRB			160
56	Subtribe	Subtribe	Subtribe	SBT			170
60	Genus	Genus	Genus	GEN			180
61		Nothogenus		NOG	X		
63	Subgenus	Subgenus	Subgenus	SBG	(	)	190
64	Division[Genus]			DIG			
65	Section[Genus]	Section[Genus]	Section[Genus]	SEG	sect._		200
66	Subdivision[Genus]	Subsection[Genus]	Subsection[Genus]	SUG			210
67	Group[Genus]	Group[Genus]	Group[Genus]	GRG			
68	Superspecies	Superspecies	Superspecies	SPS	supsp._		
69	Aggregate	Aggregate	Aggregate	AGG	aggr._		
70	Species	Species	Species	SPE			220
71		Nothospecies		NOS	X		
72	Microspecies	Microspecies	Microspecies	MSP	msp._		
73	Subspecies	Subspecies	Subspecies	SBS	subsp._		230
74	Variety	Variety	Variety	VAR	var._		240
75	Subvariety	Subvariety	Subvariety	SBV	subvar._		250
76	Form	Form	Form	FRM	forma_		260
77	Subform	Subform	Subform	SFR	subforma_		270
80	Infraspecies	Infraspecies	Infraspecies	INF	infra._		
81	Natio	Natio	Natio	NAT	nation_		
82	Race	Race	Race	RAC	race_		
83	Group	Group	Group	GRP	gruppe_		
84	Morph	Morph	Morph	MOR	morpha_		
85	Type	Type	Type	TYP	type_		
86	facies	facies	facies	FAC	facies_		
87	Pattern	Pattern	Pattern	PAT	ptrn._		
88	color	color	color	COL	col._		
89	Aberrancy	Aberrancy	Aberrancy	ABR	aberr._		
90		Cultivar		CUL	cv._		
92	MSName	MSName	MSName	MSN	"	" (MS)	
95	Unnamed	Unnamed	Unnamed	UNM	"	"	
100	Hybrid	Hybrid	Hybrid	HYB			

20-29 are reserved for ranks within the "Phylum" group; 30-39 for the "Class" group; 40-49 for the "Order" group; 50-59 for the "Family" group; 60-69 for the "Genus" group; and 70-79 for the "Species" group. Within these groups, the first value is used for the over-arching Ranks (10=Kingdom, 20= Phylum, 30=Class, 40=Order, etc.); the third value is reserved for "Sub" ranks (13=Subkingdom, 23=Subphylum, 33= Subclass, 43=Suborder, etc.); and the eighth value is reserved for "Super" ranks for the next group (08=Superkingdom, 18=Superphylum, 28=Superclass, 38= Superorder, etc.). Values 80-89 are reserved for non-traditional infraspecific ranks that are not currently used by any modern Code of nomenclature (but are needed in this data model in order to track historical uses of Taxon Names). Ranks of 90 and above are reserved for other names not governed by traditional Codes of scientific nomenclature, but are nevertheless needed for complete taxonomic data management. The three ranks within this last category that have so-far been defined include "Cultivar" (used for botanical cultivar names), "MSName" (used for names intended to eventually become formal scientific names under an appropriate Code of nomenclature, but have not yet been published in accordance with respective Code requirements), and "Unnamed," which at the moment is used very generally for informal scientific name designations such as "sp. A," "n.sp. from Maui," etc. Additional ranks in this category may yet be defined, as need becomes apparent. Finally, a *TaxonRankID* value of 100 is used for all hybrids, other than the botanical ranks of "Nothospecies" and "Nothogenus" (*i.e.,* named hybrid taxa; see further discussion of hybrids below).

The other attributes of **tbl_TaxonRank** (in its current form) include *Abbreviation*, *Prefix*, and *Suffix*. These are not strictly core attributes of each **tbl_TaxonRank**, but are used by the *Taxonomer* application for formatting purposes. *Abbreviation* is a 3-character abbreviation of each rank used as a delimiter within the *CheatHierarchy* field of **tbl_Protonym** (see below). Because these values are unique for all ranks, they represent a somewhat "natural" unique key for **tbl_TaxonRank**. *Prefix* and *Suffix* are used to format the *CheatTaxonName* field of **tbl_Assertion**. They include characters that immediately precede or follow a particular *Epithet* (see definition later in this section), and are used mostly for names at ranks below "Species" (although they are used for a few higher ranks as well). The underscore character ("_") included at the end of some values of *Prefix* denote the requirement of a space character (" ") to be inserted between the *Prefix* and the *Epithet*.

Although *TaxonRankID* technically serves as a component of the unique identifier for each **Assertion** record, it only serves a function in this capacity for those relatively few cases involving autonyms for nominotypical taxa. In a broader sense, *TaxonRankID* is one of the five basic elements of an **Assertion** (see further discussion below). As emphasized above, an **Assertion** instance serves as a handle to a Taxon Concept. The implication is that the author(s) of the **Reference** linked to the **Assertion** instance had in mind a Taxon Concept, within which they included the primary type specimen of the Taxon Name represented by the **Assertion**'s linked **Protonym**. Because details necessary for ascertaining the full scope of the Taxon Concept circumscription (i.e, beyond the primary type specimens of the relevant Taxon Names) are not consistently provided in taxonomic **References**, **Assertions** are taken to represent "Name-Resolution Circumscription" units (described above). As mentioned above, a minimum of five attributes are needed to establish each Assertion as Name-Resolution Circumscription. The first three of these five attributes have already been defined: *ProtonymID* (to indicate the Name entity), *ReferenceID* (to indicate the **Reference** in which the **Assertion** is made), and *TaxonRankID*. The last of these is necessary to define the Name-Resolution Circumscription because the same Taxon Name may be used to represent different taxonomic ranks, even outside the context of nominotypical taxa. The other two basic elements of an **Assertion** include "Validity" (*i.e.,* whether or not the name was treated by the **Reference** as a valid Taxon Name, or as a junior synonym of another Taxon Name), and the taxonomic hierarchical context.

In the *Taxonomer* model, the "Validity" of a Taxon Name as used in a **Reference** is documented via the *ValidAssertionID* Foreign Key, which recursively links back to either the same or a different instance of **tbl_Assertion**. All **Assertion** instances must indicate a value for *ValidAssertionID*. Cases where the **Reference** treated the name as a valid taxon are indicated by *ValidAssertionID=AssertionID* (almost by definition, this includes all **Assertions** that are included in the **tbl_Protonym** subtype). In cases where the **Reference** treated the Name as a junior synonym of another Name, *ValidAssertionID* instead points to the (different) **Assertion** instance that represents the indicated senior synonym (*i.e., ValidAssertionID≠AssertionID*). The only other possibility is to set *ValidAssertionID*=0. Logic dictates that such instances imply that the **Protonym** was treated with "Unspecified" validity by the **Reference**. By convention, this situation is applied in those specific cases where a Taxon Name appeared in a **Reference**, but no Taxon Concept was implied (*e.g.,* Type Catalogs, etc.). This allows the use of **tbl_Assertion** to index the appearance of Taxon Names in **References**, without forcing all **Assertion** instances to imply a Taxon Concept.

In all cases, the *ReferenceID* value for the **Assertion** instance indicated by *ValidAssertionID* must be the same *ReferenceID* indicated in the current **Assertion**. When *ValidAssertionID=AssertionID*, this rule is enforced by default. In cases where *ValidAssertionID≠AssertionID*, the domain for values of *ValidAssertionID* is restricted to **Assertion** instances linked to the same *ReferenceID*. Stated another way, inter-**Assertion** linkages via *ValidAssertionID* must be established *within* a single **Reference**. While it may be tempting to establish inter-**Reference** linkages with this structure (*e.g.,* when a **Reference** explicitly bases its concept of a taxon name on that of another **Reference**), the most fundamental and explicit Taxon Concept mapping is within a single Reference. For example, consider the following **Assertion** instances:

Protonym1 sensu *ReferenceA*

Protonym2 sensu *ReferenceB*

If "ReferenceA" explicitly states the equivalent of "We regard [Protonym1] to be a junior synonym of [Protonym2] *sensu* [ReferenceB]," one could set the *ValidAssertionID* value for the **Assertion** representing "**Protonym1 *sensu* ReferenceA**" to be the *AssertionID* value for the **Assertion** instance representing "**Protonym2 *sensu* ReferenceB**." In doing so, however, the linkage would span disjunctions in two separate component attributes (*i.e., ProtonymID* and *ReferenceID*). Given the hypothetical statement quoted in the first sentence of this paragraph, it is unambiguously implied that ReferenceA regards "Protonym2" to be valid. Thus, the third assertion, "**Protonym2 *sensu* ReferenceA**" can be safely inferred, and assigned to a new **Assertion** instance. Fundamentally, *this* **Assertion** of "Protonym2" (*i.e.,* within "ReferenceA") is the one most unambiguously representing what "ReferenceA" regarded "Protonym1" to be a junior synonym of. For this reason, it should be noted that the domain for *ValidAssertionID* is restricted even further to those **Assertions** linked to the same *ReferenceID* where *ValidAssertionID=AssertionID*.

Of course, intra-**Reference** concept mapping is an important component to any robust taxonomic data model. The primary intent of this article is to describe the "core" components of the *Taxonomer* model used for managing taxonomic information. Such intra-**Reference** concept mapping is accomplished within a different "module" of the *Taxonomer* data model, as described in the next section under "Concept Mapping."

The last of the five basic elements of an **Assertion** is the hierarchical context. In the Taxonomer model, this is accomplished via the *ParentAssertionID* attribute of **tbl_Assertion**. *ParentAssertionID* links an **Assertion** instance to another **Assertion** instance that represents the most immediate parent taxon in which the first taxon was placed as indicated within the **Reference**. There is no restriction on rank gaps that may occur between a parent and child Assertion instance, but gaps that exceed one rank-group cluster (*e.g.,* a Genus Name linked directly to an Order Name, or a Family Name linked directly to a Class Name) are

treated as cases of "*Incertae Sedis*" on standardized output formats. As with *ValidAssertionID*, the domain for *ParentAssertionID* is restricted to other **Assertions** that share the same *ReferenceID*, and where *ValidAssertionID=AssertionID* (*i.e.,* Names treated as valid). The domain is further restricted to those **Assertion** instances with a lower value of *TaxonRankID* (*i.e.,* higher taxonomic rank) than the current instance. Therefore, unlike the case with *ValidAssertionID*, *ParentAssertionID* cannot be equal to *AssertionID* for a given instance of **Assertion** (for obvious reasons).

In addition to these restrictions, *ParentAssertionID* must link to the *most direct* parent **Assertion** within the **Reference**. For example, if a **Reference** places species '*c*' within the subgenus "*B*" of the genus '*A*', then the *ParentAssertionID* for the Assertion of species '*c*' links to the **Assertion** of subgenus '*B*' (within the same **Reference**). Like *ValidAssertionID*, *ParentAssertionID* can be set to "0" (the functional equivalent of a Null value, as described earlier). The first reason for this is that for Taxon Names treated above the rank of "Species," **References** often do not specify what parent taxon a given Taxon Name is asserted to be included with (indeed, very few taxonomic **References** explicitly state full hierarchical context all the way up to the rank of "Kingdom," so at some point most **References** cite a Taxon Name without placing it within a parent taxon). Another reason is that, for **Assertions** made about non-valid taxon names (*ValidAssertionID≠AssertionID*), there technically is no asserted parent taxon (*i.e.,* the synonymously treated name automatically inherits the *ParentAssertionID* value of the indicated *ValidAssertionID*). Thus, another business rule of the *Taxonomer* data model is that all cases where *ValidAssertionID≠AssertionID*, the corresponding value of *ParentAssertionID* must be set to zero ("Unspecified").

Beyond the five basic elements of **Assertions** described above, a sixth attribute is needed to fully define the treatment of a Taxon Name: the Name itself. Although it may seem that the "Name" should be treated as an attribute of **tbl_Protonym**, taxonomic practice allows for variance in the specific string of characters used to represent a name. As such, a given **Protonym** may be represented by slightly different strings of characters in different **References**. There are several reasons why this may be. Some Names may have different endings depending on the specific rank at which they were treated (*e.g.,* in Zoology, family names end with "-idae," whereas subfamily names end with "-inae"). The suffix of species and subspecies epithets used as adjectives may change depending on the gender of the genus in which they are treated ("-*a*," "-*us*," "-*um*"). Finally, names may be consistently misspelled in certain **References**. Thus, the actual string of characters representing the name itself is best treated as an attribute of **tbl_Assertion**, rather than **tbl_Protonym**.

I have chosen the word *Epithet* for the attribute that stores the string of characters representing a taxonomic name as it appears in an **Assertion**. As emphasized elsewhere in this article, a **Protonym** entity is regarded as applying only to the terminal component of a multinomial (*e.g.,* a species epithet, rather a genus-species binomial), and therefore "*Epithet*" seems appropriate to emphasize this point. The main problem with the using word "*Epithet*" for this purpose is that, within a biological context, it is sometimes defined specifically as "the part of a taxonomic name identifying a subordinate unit within a genus" (*e.g.*, Merriam-Webster, 1993). Because Assertions span all taxonomic ranks (*i.e.,* including those above the rank of Genus), a strict definition of *Epithet* in this sense renders it somewhat inappropriate. However, a more general definition of *Epithet* (as it appears in *Webster's*), is "a characterizing word or phrase accompanying or occurring in place of the name of a person or a thing."; or, "a term, phrase, expression" (OED – Simpson & Weiner, 1989). At the risk of contrasting with broader practice, the term *Epithet* is herein defined as any monomial unit of a taxonomic name (at any rank), or as a complete hybrid formula (including all relevant ranks).

Epithet is populated with the exact character string that the corresponding **Reference** used when citing the associated Taxon Name. The main purpose of this field is to

document the exact spelling (including hyphens, numbers, and other symbols, where applicable) of the name as it appeared within the **Reference**. Only the "terminal" epithet is included for binomials, trinomials, and other multinomials (including subgenera). In the special case of hybrids, the ***complete*** hybrid formula (including names of genera and all other applicable ranks) is entered, exactly as spelled, abbreviated, and punctuated in the **Reference**. In cases where a non-hybrid Taxon Name is spelled in different ways within a single **Reference**, the *Epithet* can either be taken as the most frequently used spelling within the **Reference** (if one spelling is used with much greater consistency than any other), with the alternative misspelling(s) relegated to a *Comment*; or, two or more "Sub-Reference" instances may be defined for the **Reference**, for each alternate spelling. When a hybrid formula appears in more than one form in a **Reference**, the *Epithet* is taken to be the most complete version (*i.e.,* fewest abbreviations).

Related to *Epithet* is the *EpithetQualifier* attribute. This attribute stores any additional textual information applied to the *Epithet* in the **Reference** (*e.g.,* "c.f.," "*sensu stricto*," "*sensu lato*," non-**Reference**, etc.), but that does not strictly constitute part of the *Epithet* character string itself.

The *ReliabilityID* Foreign Key links to the look-up table **tbl_Reliability**. This attribute is intended to be a semi-objective guide to how reliable an interpretation may be. Although some degree of subjectivity is inevitable in assigning this value, the domain of six discrete values is designed to be as objectively-discernable as possible, while still providing some meaningful function. The values range from 0-5, and are described in Table 4. A value of 5 represents the highest reliability, and is limited to only those **References** constituting the original description of a taxon name, or a first "New Combination" **Assertion**. All **Assertions** representing **Protonyms** would be assigned this value. A value of 4 corresponds to other taxonomic revisionary work that explicitly treats the associated Taxon Name within the context of the revision. A value of 3 indicates that the **Reference** making the **Assertion** did so within a taxonomic context, but not as a revisionary work for the particular Taxon Name. A value of 2 indicates that the **Reference** was scientific in nature, though not specifically a taxonomic work (*e.g.,* an ethological or ecological publication). A value of 1 is used for popular literature and other non-scientific **References**. This same scale can be applied (more or less) to Assertions that are not published (*e.g.,* specimen determinations), based on the nature of circumstances and qualifications of the **Agent(s)** providing the determinations. A value of 0 (default) indicates that the nature of the **Reliability** has not been reliably ascertained. It should be emphasized that the *ReliabilityID* value, as an attribute of **tbl_Assertion**, applies only to a particular **Protonym**-**Reference** combination (*i.e.,* not to an entire **Reference**). For instance, a single article might describe new species and establish new binomial combinations (*ReliabilityID*=5) as part of a taxonomic revision that includes many previously-described species and genera (*Reliabili-*

Table 4. Description of defined values of *ReliabilityID*, as used within tbl_Assertion.

ReliabilityID	Reliability	Description
0	Uncertain	Nature of **Reliability** not known or uncertain.
1	Non-Scientific	Taxon Names within non-scientific **References**, or **Determinations** made by lay persons.
2	Scientific	Taxon Names within non-taxonomic **References**, or **Determinations** made by a scientists who do not specialize in taxonomy.
3	Taxonomic	Taxon Names within taxonomic **References** that are not part of a revisionary work, or **Determinations** made by taxonomists who do not specialize in the particular taxonomic group.
4	Revision	Taxon Names as used within the context of a taxonomic revision, or **Determinations** made by taxonomists during their revisionary work.
5	Original Description/ New Combination	**Protonyms** and other **Assertion** instances that represent new combinations.

tyID=4), and also make reference to other Taxon Names not included within the scope of the revisionary work (*ReliabilityID*=3). However, in most cases, *ReliabilityID* values of 1 and 2 will apply unilaterally for all **Assertions** within an entire **Reference**, as these categories tend to apply more to the nature of the **Reference**, rather than the use of Taxon Names within the **Reference**.

Another important attribute of an **Assertion** is *Pages*. In the current implementation of the model, this attribute is a simple text field to allow entering whatever information is necessary to designate where, within the corresponding **Reference**, an **Assertion** can be located. Future implementations of the model might break this information out into a separate table, or at the very least, split it into two separate attributes; one for pages, and one for illustrations (including figures and plates).

The *IsQuestioned* attribute is a simple boolean flag to indicate that the **Reference** expressed uncertainty in its specific treatment of a Taxon Name. Information about what, exactly, was questioned, and why it was deemed questionable, should be included in linked **Excerpts** or **Comments** (see next section). Future versions of the *Taxonomer* model may provide more robust mechanisms for characterizing the nature of questionable **Assertions**.

The last two data-bearing attributes of **Assertion** are provisional, and may be rendered redundant depending on what additional subtypes of **Assertion** are created (other than **Protonym**). Both fields (*IsNewCombination* and *IsFirstRevision*) are boolean values with self-evident meaning, intended to flag special-case **Assertions** which have important taxonomic or nomenclatural meaning. Either of these could be expanded to full (non-exclusive) subtypes, if additional attributes relevant to each category are deemed worthy of documenting.

An earlier version of **tbl_Assertion** included the attribute *Sequence*. The purpose of this field was to record the actual sequence in which a series of Taxon Names were listed in a **Reference**, within the context of a single parent taxon. This information is sometimes useful, because it may represent an effort to provide some sort of interpreted phylogenetic context of a taxon among related taxa. Because the meaning of such *Sequence* information is not standardized and its application within **References** is inconsistent, however, it was excluded from this version of the model.

CheatTaxonName is formatted as the complete Taxon Name (identical to *Epithet* for ranks of genus and higher, or complete binomial, trinomial, or other multinomial for ranks lower than Genus). The values for Names of infrageneric ranks are derived from recursive concatenation of *Epithets* up to the rank of Genus, and the value for hybrids represents the complete hybrid formula as derived from the linkages established in the **tbl_HybridAssertion** table (see below), which may differ somewhat from the hybrid formula as actually written in the **Reference** (*i.e.,* the contents of *Epithet* for hybrid **Assertions**). *CheatFullTaxonName* is simply the value of *CheatTaxonName*, expanded to include all appropriately-formatted authorships. *CheatNominotypical* is simply a boolean field used to flag those **Assertions** that represent autonyms (nominotypical taxa; *i.e.,* the *ProtonymID* value of an **Assertion** instance equals the *ProtonymID* value of the **Assertion** indicated by the *ParentAssertionID* Foreign Key). *CheatStatus* is a standardized "natural language" statement representing the combination of the core **Assertion** elements (validity, hierarchical placement, rank, and *Epithet*; *e.g.,* "Valid as originally described.," "Junior Synonym of {OtherTaxonName}," "Valid {TaxonRank} within {ParentTaxonName}," etc.)

tbl_Protonym

As has been alluded to previously, a **Protonym** always represents a monomial Name. In cases of infrageneric Names (subgenera, binomials, trinomials, and other multinomials), the **Protonym** refers only to the terminal unit of the Name. Thus, new **Protonym** instances are not created for each different combination of binomial, trinomial, or other multinomial. This point is emphasized to avoid confusion, as the word "Name" is often used in reference to a full-context multinomial, and within the context of this data

model, the concept of a **Protonym** is being used to represent a Taxon Name. For clarity, a **Protonym** should be thought of as the nomenclatural basis of a monomial (or only the terminal epithet of a multinomial), in the context of its original creation (*i.e.,* Code-compliant original description).

Also as mentioned earlier, **tbl_Protonym** represents a subtype of **tbl_Assertion**, indicating those special-case **Assertion** instances that constitute original descriptions of Taxon Names (*i.e.,* **Protonyms**). The recursive relationship between this table and **tbl_Assertion** has been described above. The other attributes of **tbl_Protonym**, described here, are data elements specifically associated with **Protonyms** (not with non-**Protonym Assertions**).

The Foreign Key *TypeProtonymID* is a recursive link, and is used primarily for names at the genus-group and family-group ranks, to indicate which species-group or genus-group (respectively) **Protonym** was designated as the "Type Species" or "Type Genus" for the genus-group or family-group name. This is an attribute of **tbl_Protonym**, rather than **tbl_Assertion**, because Taxon Names of all ranks are ultimately typified by the primary type specimen of the terminal type **Protonym**, and thus not by a Taxon Concept. Name-based type designations (*e.g.,* type species of a genus) are interpreted here as place-holders to establish a complete link between a higher-rank name and a primary type specimen. Therefore, links to Name-based types are established via a **Protonym** instance, rather than an **Assertion** instance. However, this highly simplified approach to recording type taxa may need to be changed in future versions to accommodate cases where the type taxon was designated in a subsequent **Reference**.

The Foreign Key *WordTypeID* links to the same **tbl_WordType** that was described earlier under the **tbl_Glossary** heading of the "References" section of this document. The purpose for allowing this link is to specify what word form the *Epithet* of a **Protonym** takes (*e.g.,* "Noun (apposition)," "Adjective," etc.), which can be useful for determining proper name spelling (*e.g.,* whether the spelling of a species *Epithet* changes when treated in a Genus of a different gender).

NomenCodeID establishes a link to an instance of **tbl_NomenCode**, indicating the particular Code of Nomenclature under which a particular **Protonym** is governed. In cases of names at ranks higher than those governed by the relevant codes, the value indicates which Code the child taxa fall under. This field is important for determining specific formatting rules of authorships, etc.

The two non-key attributes of tbl_Protonym are *Gender* and *IsFossil*. The *Gender* field mirrors the field of the same name in **tbl_Person** of the "Agents" section (with the addition of "Neuter"), and is used to indicate the gender of genus-group **Protonyms**. *IsFossil* is a boolean flag field set to "True" if the **Protonym** applies to a Taxon Name created for a fossil taxon.

CheatFullProtonym is used to store a standardized formatted name, including authorship. The format is generally as "*Epithet*, OriginalParent Authorship" (*e.g.,* "*speciesname*, *Genusname* Author-Name(s)." The reason this field exists separately from *CheatFullTaxonName* is that the latter is formatted as it would generally appear in print (*i.e., Genusname speciesname* AuthorName[s]), whereas the former is formatted with the terminal unit of a multinomial (*i.e.,* the *Epithet* represented by the **Protonym** instance) listed first. *CheatAcceptedAssertionID* indicates which **Assertion** the user of the database system has decided to follow as representing the "correct" current status of each **Protonym**. The reason this is considered a "Cheat" field for **tbl_Protonym** is that it will eventually be derived from a different set of tables that will track multiple "accepted" **Assertions**, as designated by different institutional authorities. The final design of these tables has not yet been determined, and depends to some extent on how various taxonomic services document their preferred Taxon Name statuses (see further elaboration in the "Accepted Status" section of this article). *CheatHierarchy* is a specially-formatted long text (memo) string that includes the full-

context taxonomic hierarchy for each **Protonym**, as determined through a recursive series of values of *CheatAcceptedAssertionID* for each name at each rank through the hierarchy.

tbl_HybridAssertion
Hybrid names (hybrid formulae) are treated in many respects the same way that non-hybrid Taxon Names are treated. As discussed earlier, the *Epithet* of an **Assertion** representing a hybrid Name is recorded just as the hybrid formula appears in **Reference** (unlike non-hybrid *Epithets*, which are monomial). **Protonyms** are created for hybrid names, just as they are for traditional taxon names. Except for the botanical ranks of "Nothospecies" and "Nothogenus," however, **Protonyms** representing hybrids do not have formal "original descriptions" as governed by Codes of nomenclature. By convention, the **Protonyms** of such hybrids are taken as the first appearance of the hybrid cross in any **Reference** (with the first published **Reference** citing the hybrid taking priority over an earlier unpublished **Reference**). Also hybrids are assumed to be symmetrical. That is, if one **Reference** cites a hybrid as "*SpeciesA* x *SpeciesB*," and another cites it as "*SpeciesB* x *SpeciesA*," they are taken to represent the same hybrids, and share a common **Protonym**. Again, except for the botanical ranks of "Nothospecies" and "Nothogenus," all Assertions representing hybrids are assigned to *TaxonRankID*=100.

All names constituting hybrids (including botanical "Nothotaxa") are represented by instances in **tbl_HybridAssertion**. This table represents another subtype of **tbl_Assertion** (non-exclusive of **tbl_Protonym**), that is populated with **Assertions** that constitute hybrid Names. At present, the only purpose of this subtype table is to record the pair of **Assertions** that represent the two "parent" taxa of the hybrid. As with *ValidAssertionID* and *ParentAssertionID* Foreign Keys in **tbl_Assertion**, all three values in a given instance of **tbl_HybridAssertion** (*HybridAssertionID*, *HybridParent1ID*, and *HybridParent2ID*) must point to three different **Assertion** records, all of which share the same value of *ReferenceID*. By convention, *HybridParent1ID* links to the alphabetically-first member of a hybrid, and *HybridParent2ID* links to the alphabetically-second member. In cases of secondary hybrid crosses (*e.g.,* "*SpeciesA* x [*SpeciesB* x *SpeciesC*]"), *HybridParent1ID* preferentially links to the non-hybrid "parent," and *HybridParent1ID* links to the **Assertion** representing the hybrid parent (*e.g.,* the Assertion representing the cross "*SpeciesB* x *SpeciesC*"). If both parents are hybrids (*e.g.,* "[SpeciesA x *SpeciesB*] x [*SpeciesC* x *SpeciesD*]"), the *HybridParent1ID* and *HybridParent2ID* are, again, determined by alphabetical priority. An alternative convention would be to define *HybridParent1ID* as the female parent, and *HybridParent2ID* as the male parent; however, a more effective approach would be to record this information in one or more dedicated attributes.

tbl_ObjectiveStatus
In most cases of Names treated as synonyms (*i.e., ValidAssertionID≠AssertionID*), the treatment of Name as such is a subjective assertion. However, there are cases of objective synonymy and other objective nomenclatural statuses as dictated by the appropriate Code of nomenclature. Examples include two different Names sharing the same primary type specimen, cases of Homonymy (and their corresponding replacement names), and other forms of objective unavailability of Names. A robust system for managing such objective nomenclatural status has not yet been developed in the *Taxonomer* data model. However, a simple indexing of such is accomplished using **tbl_ObjectiveStatus**, linked to **Assertion**. This table is linked to **tbl_Assertion**, rather than **tbl_Protonym**, because different **References** might have different interpretations of what the correct **ObjectiveStatus** of a Name should be. The relationship is one-to-many because a **Reference** might acknowledge more than one **ObjectiveStatus** for any given Name. Each instance of **ObjectiveStatus** is classified by type, indicated by the value of *ObjectiveStatusType* in the corresponding instance of the **tbl_ObjectiveStatusType**, table, linked via *ObjectiveStatusTypeID* (examples shown in blue text in Figure 6). A more robust management scheme for this sort of information would cross-link to tables

describing individual Articles as they appear in the respective Codes of Nomenclature, and would likely include other attributes to qualify the nature of the **ObjectiveStatus** in greater detail.

One important point that should be clarified about **tbl_ObjectiveStatus** is that each objective status instance applies only to the Taxon Name; not the Taxon Concept. As mentioned above, the linkages to the **Assertion** table (rather than the **Protonym** table) are established only as a convenient way to document each **ObjectiveStatus** instance in the context of a particular **Reference**.

Limitations

- The "Taxa" components of the *Taxonomer* data model described above do not allow for the mapping of Taxon Concepts to other Concepts. Rather, these components track nomenclatural information, and provide a basic unit of Name-Resolution Circumscriptions (**Assertions**) which in most cases represent "Potential Taxa" (*sensu* Berendsohn, 1995). The correlation of **Assertions** to other intra-Reference **Assertions** is described below, under the "Mapping Concepts" section.

- Another limitation imposed by the present structure is that typification of taxon names by other taxon names (*e.g.,* type-species of Genus names) is assumed to be straightforward and objective. In reality, this relationship is not always so clear. The process of typification varies among the different Codes of nomenclature, and can sometimes be quite complex. A more advanced model would break *TypeProtonymID* out from **tbl_Protonym**, and would instead include a more robust structure to accommodate nomenclatural typification.

- Another limitation of the *Taxonomer* model in its current implementation is that there is no direct means to address misapplication of Taxon Names. Ultimately, a misapplication of a Taxon Name is defined as a case where a **Reference** applies a Name to a Taxon Concept from which the **Reference** would exclude the Primary Type specimen of the Name. Stated more simply, the **Reference** applied a Name based on a misunderstanding of the correct typification of a name. Such misidentifications can only be revealed through the context of a subsequent Reference, and as such can be mapped via a **tbl_AssertionRelation** instance of **RelationType** "excludes." However, a more direct approach to misapplied names may be more appropriate, and will be considered for future versions of the *Taxonomer* data model.

- Although not a 'limitation' *per se*, it is worth discussing further the connection between the word "**Protonym**" and the word "Basionym." "Basonym" (without the "i") is defined in *Merriam-Webster's Third New International Dictionary* as:

  > *"The earliest validly published name of a taxon, being in the case of a binomial or trinomial the source of the valid specific or subspecific epithet when the taxon is transferred to a new combination and in technical usage always accompanied by the name of the original author. (Crataegus spicata Lamark:Amelanchier spicata)"*

  Following this definition, the use herein in place of **Protonym** would seem appropriate. Although "Basionym" is used primarily in botanical contexts, it could easily be extended to represent the same meaning in Zoological contexts (as is already being done in some zoological contexts). However, "Basionym," strictly defined, *includes* the genus-species[-subspecific] combination of names (binomial, trinomial, etc.); but only the terminal epithet is implied by the **Protonym**, as defined herein. Moreover, the term "Basionym" is usually used only in the context of lower-level taxonomic ranks (genus, species, subspecies, etc.), but **Protonym** is here extended to apply to *all* taxonomic ranks. Another problem with the word "Basionym" is that it implies that a name has achieved legitimacy within the relevant nomenclatural Code. Strictly speaking, this would appear to restrict its use to include only

"formal" scientific names after they have been published in accordance with the relevant nomenclatural Code. However, there are many applications that need to cite a Taxon Concept ***before*** it has received a formal Code-compliant Name, and there is also a need to refer to hybrid formulae, which are not represented by Code-compliant original descriptions. Finally, whereas the word "Basionym" typically refers to the actual *name* only, **Protonym** is here extended to imply the authorship (or more directly, the Reference association) that was involved with the original establishment of the Basionym. In this sense, a **Protonym** includes components of both a Basionym and a botanical "Protologue," making the term "**Protonym**" (which could be interpreted as an amalgamation of "Protologue" and "Basionym") somewhat appropriate.

The term "Protonym" is defined by the *OED* (Simpson & Weiner, 1989) as:

"*The first person or thing of the name; that from which another is named.*"

In this case, the "thing" is the Code-compliant original description of a taxonomic name, and subsequent uses of that name in different contexts to represent potentially different Taxon Concepts constitute examples of how "another is named." Although this term is still bound by the "nym" suffix to apply strictly to a "Name" (rather than a Name-Reference intersection, as would be a subtype of **tbl_Assertion**), it seems to be a more appropriate term than "Basionym" in this context. Its implied meaning as a "name" *per se* is not entirely inappropriate, because even if it represents a subtype of an **Assertion**, it is intended to represent only the original *name* component of that **Assertion**.

Interface With Other Relevant Information

The previous three sections ("**Agents**," "**References**," and "**Taxa**") describe what I consider to be the "core" components to a taxonomic data management model. With only these components, one can develop a comprehensive index of taxonomic nomenclature and how individual Names have been used in both published and unpublished form, throughout the history of taxonomy. This, by itself, would constitute a very powerful tool for modern taxonomists needing to research the history of Taxon Names. However, the Taxonomer data model does not stop there. Described below are examples of how these "core" components of the model can be used and applied in versatile ways to accommodate broader information management needs. The first five sub-sections ("*Accepted Status*," "*Mapping Concepts*," "*Specimen Determinations*," "*Taxon Excerpts*," and "*Common Names*,") describes ways in which **Assertions** can serve a variety of roles for broader data management. The last sub-section ("*General Data Management*") describes how certain general data management needs (Code Numbers, Comments, and the logging of data edits) are met within the *Taxonomer* model. These components have been lumped together in this section partly because they represent secondary applications of the data, and in some cases, partly because the associated tables and their attributes have not yet been robustly developed.

Accepted Status

A common criticism (*e.g.,* Pullan et al., 2000; Raguenaud, 2002) of simple taxonomic data models is that many of them accommodate only a single taxonomic "view." While this may be seen as a limitation of these simpler models in a broader context, several of these models are specifically intended to represent a single taxonomic view. For example, the primary objective of the Integrated Taxonomic Information System (ITIS, 2003) is to provide a single taxonomic view of available nomenclature, to facilitate conformance of various U.S. federal agencies to a common taxonomic standard. Indeed, many potential users of taxonomic data do not want or need to be presented with multiple taxonomic views, but rather would implicitly accept the view as determined by some taxonomic authority.

While a well-populated **Assertion** table as described above serves a valuable function to practicing taxonomists in documenting the historical treatment of Taxon Names, there is nothing within **tbl_Assertion** itself that identifies the "correct" or "accepted" status

of any particular Name. A simple algorithm could be developed to assume that the most recent **Assertion** (according to the *Date* of the linked **Reference**) flagged by a minimum **Reliability** value (*e.g., ReliabilityID*>=3) automatically represents the "accepted" status of the associated **Protonym**. While such an algorithm would be reasonably objective, it would probably be deemed by most to be inadequate. In most cases, the designation of an "accepted" status of a name requires careful assessment by a qualified taxonomist.

One approach would be to identify certain "meta"-authorities for Taxon Names. Such meta-authorities could have restricted taxonomic scope (*e.g.,* the online version of the *Catalog of Fishes* for fish taxa; the Mammal Networked Information System [MaNIS] for mammal taxa; etc.); restricted geographic scope (*e.g.,* ITIS for North America; Hawaii Biological Survey [HBS] for the Hawaiian Archipelago; etc.); or all-encompassing (*e.g.,* Species2000; BIOSIS; etc.). These meta-authorities could serve as authors to **References**, to which **Assertions** are linked for every **Protonym** that falls within the taxonomic or geographic scope of each respective meta-authority. Once a database is populated with such **Assertions** linked to **References** authored by established meta-authorities, the database user could define a priority scale among several of these meta-authorities. The database application could automatically establish the current status of each Name based on the most recent **Assertion** by the meta-authority with the highest-defined priority (among those meta-authorities that provide an **Assertion** for a given Name). In cases where a particular Name has not been treated by any of the established meta-authorities (*i.e.,* Names with no **Assertions** linked to **References** authored by meta-authorities), the database application could resort to some default method of determining current status (*e.g.,* the most recent **Assertion** of a defined minimum rank, as described above).

Though this solution would likely be effective, it requires a potentially vast number of additional **Assertion** instances to be created – at least one for each Name multiplied by the number of meta-authorities that treat them (more if treatments by any particular meta-authority change over time). This, in itself, is not necessarily a major problem; however, in most cases it is not reflective of the actual information structure. In most cases, these meta-authorities do not themselves provide true taxonomic assertions about each name they treat, but instead generally select one pre-existing status, from among various recent and relevant publications and other expert opinion, that they chose to follow. Based on this perspective, one possible data structure is represented in Figure 7.

By this structure, rather than create new **Assertion** instances for each name as treated by each meta-authority, **tbl_AcceptedAssertion** allows for meta-authorities to identify specific pre-existing

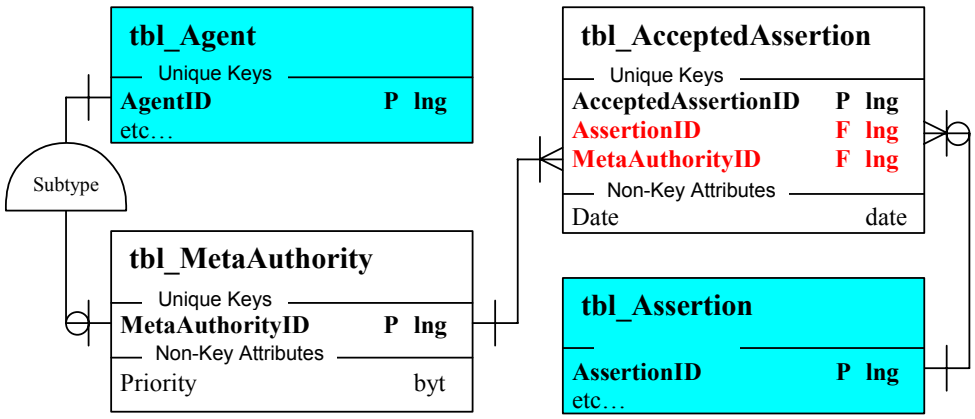

Figure 7. Possible table structure to determine Assertions that represent "accepted" status, according to defined "meta-authorities."

Assertions (which in most cases would be contained in major published taxonomic works, with *ReliabilityID* values of 3 or higher) as representing the current "accepted" status for the corresponding **Protonym**. **tbl_MetaAuthority** would represent a non-exclusive subtype of **tbl_Agent**, containing links to those **Agents** deemed by the database user to be meta-authorities for determining current status of **Protonyms**. The *Priority* attribute would be a simple priority-ranking value for each **MetaAuthority** instance, used in establishing the preferred "accepted" status of any particular **Protonym** treated by more than one **MetaAuthority**. Each instance of **tbl_AcceptedAssertion** would link to one **Assertion** instance and one **MetaAuthority** instance (via Foreign Keys *AssertionID* and *MetaAuthorityID*), and would be time-stamped via the *Date* attribute of **tbl_AcceptedAssertion**. Because *MetaAuthorityID* constitutes an *AgentID*, one could technically be replaced by a **Reference** instance (*i.e., MetaAuthorityID* and *Date* together comprise a date-stamped **Agent**). With sets of **tbl_MetaAuthority** and **tbl_AcceptedAssertion** that are well-populated, a straight-forward algorithm could be implemented within a database application using the *Priority* attribute (along with some method of establishing a default accepted **Assertions** for **Protonyms** not treated by any **MetaAuthority**) to automatically derive values of *CheatAcceptedAssertionID* for each **Protonym** instance.

The method described above for obtaining sets of **AcceptedAssertion** values for each **Protonym** would only be practical if some sort of standard were to be adopted by each designated **MetaAuthority** to automatically provide and update such values electronically. If no such standard existed (as it does not currently exist), and database users were forced to manually assign each instance of **tbl_AcceptedAssertion**, then the structure described above would serve little more than the function of logging how the database user arrived at each value of *CheatAcceptedAssertionID*. There are, of course, other ways to derive some sort of "*AcceptedAssertionID*" for each **Protonym**. A final solution within the context of the *Taxonomer* data model has yet to be determined; however, it will almost certainly take the form of a method to identify one specific **Assertion** for each **Protonym** to represent its "accepted" status.

Mapping Concepts

As mentioned earlier, the data model components described in the "Taxa" section above do not include a mechanism for recording relationships among intra-**Reference** Taxon Concepts. Geoffroy and Berendsohn (2003) provide a description of how taxonomic concepts can potentially relate to each other (summarized in their Table 3). In many cases, two Concepts may be identical to each other (*i.e.,* they are *congruent*). In other cases, one Concept may entirely contain another Concept. For instance, suppose "Protonym1" is created by "Reference1" to represent a broad population of organisms, and "Reference2" later divides the broad population into two sub-populations, retaining "Protonym1" for one of the sub-populations (*i.e.,* that which included the Primary Type specimen of "Protonym1"), and establishing the new "Protonym2" for the other sub-population. In this case, **Protonym1** ***sensu*** **Reference1** *includes* both **Protonym1** ***sensu*** **Reference2** and **Protonym2** ***sensu*** **Reference2**. Conversely, both **Protonym1** ***sensu*** **Reference2** and **Protonym2** ***sensu*** **Reference2** are *included in* **Protonym1** ***sensu*** **Reference1**. Other, less-frequently encountered kinds of relationships that may exist between two **Assertions** include cases where one **Assertion** *overlaps* with another **Assertion**, and where one **Assertion** *excludes* another **Assertion**. In the latter case, the assumption is that the respective **Protonyms** for each **Assertion** are identical. In all other cases, relationships may be defined between **Assertions** linked to either identical or different **Protonyms**.

The table structure used to establish these relationships between inter-**Reference Assertions** (*i.e.,* Concept Mapping) is shown in Figure 8. The main table, **tbl_AssertionRelation**, includes two Foreign Keys to **tbl_Assertion** (*AssertionID*, and *RelatedAssertionID*), representing the two **Assertions** for which a relationship is established. *AssertionID* links to the more recent of the two **Assertions** (*i.e.,* the one

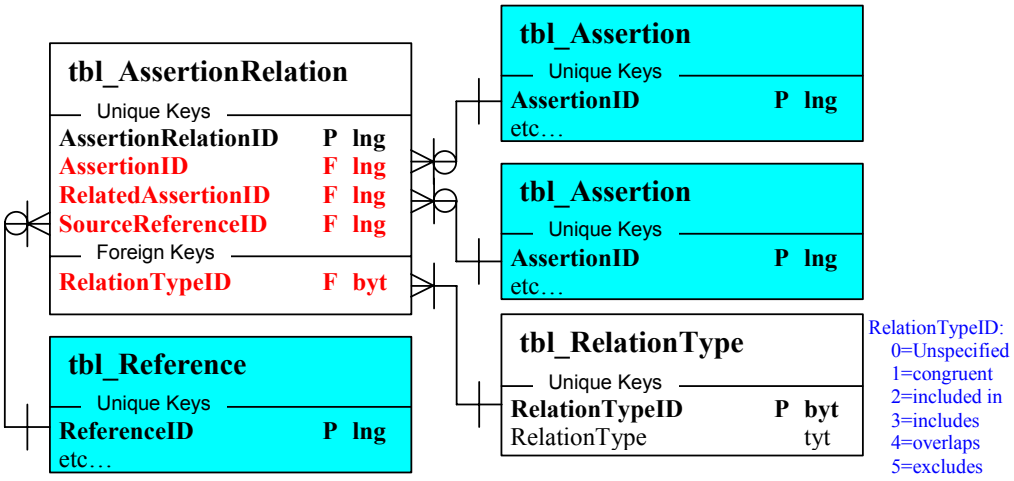

Figure 8. Table structure for mapping Assertions to other Assertions (*i.e.,* Taxon Concept Mapping).

whose corresponding **Reference** was dated most recently), and *RelatedAssertionID* links to the older of the two **Assertion**. This chronology is established by convention because a more recent **Reference** can establish relationships between its own **Assertions** and **Assertions** of a previous **Reference**, but not the other way around.

SourceReferenceID establishes a link to the **Reference** that determined the relationship between the two **Assertions**. This may be the same value as *ReferenceID* of the **Assertion** instance linked to *AssertionID* of **tbl_AssertionRelation** (in the case where a **Reference** explicitly states how its Concept of a **Protonym** relates to a Concept in another **Reference**), or it may be an entirely different **Reference** (either published, or created as an unpublished **Reference** specifically for the purpose of establishing a relationship between two **Assertions**; but in either case dated more recently than the more recent of the two related **Assertions**). Because of the imposed direction of chronology with respect to the **Assertions** linked by *AssertionID* and *RelatedAssertionID*, *SourceReferenceID* cannot be the **Reference** of the **Assertion** linked by the *RelatedAssertionID*.

RelationTypeID links to the surrogate Primary Key of **tbl_RelationType**, which includes the five types of relationships defined above ("*Congruent*," "*Included In*," "*Includes*," "*Overlaps*," and "*Excludes*"; see first paragraph of this section above).

Specimen Determinations

As described in the "Introduction" section of this article, the *Taxonomer* data model began as a way to establish a taxonomic authority for specimen databases. This purpose has been retained, and is accomplished by way of the **tbl_Determination** table. This table establishes links between **tbl_Assertion** and **tbl_Specimen**, and is the only way by which Taxon Names are assigned to **Specimens**.

A full description of the entire structure for "Specimen" components of the *Taxonomer* data model is beyond the scope of this article. In summary, an instance in **tbl_Specimen** may be one of three types: "Vouchered" (physical specimens collected and preserved in a Natural History collection), "Unvouchered" (specific living organisms that were only observed or photographed in the environment, but not physically collected and deposited in any Natural History museum collection), and "Virtual." The latter is a special-case type, representing an abstract **Specimen** or **Specimens** that may or may not have actually existed physically. It is used primarily as a placeholder to establish necessary links between Taxon Names and certain kinds information contained within **References**, but not associated directly with specific vouchered

or unvouchered **Specimens** (*e.g.,* character states and geographic distributions). The fundamental intent of this method of categorizing **Specimen** data is to provide a unified approach to organizing "occurance" data (see Morris, 1998).

Traditionally, databases of **Specimens** record not only a Taxon Name assigned to the **Specimen**, but also the name of one or more individuals who determined the **Specimen(s)** to be identifiable to that Taxon Name, as well as some form of date indicating when the determination was made. In the vocabulary of this data model, this relationship could be stated as:

Specimen determined to be a representative of **Protonym** by **Agents** on *Date*

This can be further simplified in two additional steps:

Specimen determined to be a representative of **Protonym** by **Reference**

and

Specimen determined to be a representative of **Assertion**

This reduction in data structure is consistent with the true nature of information established when a specimen is determined to be identifiable to a Taxon Name. As emphasized earlier, an **Assertion** represents a Taxon Concept. The informational content of a **Specimen Determination** is that the **Specimen** is a member of a taxonomic circumscription as represented by the determined Taxon Name. More specifically, the **Specimen** has been determined to belong to the Taxon Concept envisioned by the determiners themselves, on the date when the **Determination** was made. Thus, the determiners can be represented as authors to a **Reference**, which established an **Assertion** about a **Protonym**, to which the **Specimen** belongs.

As illustrated in Figure 9, **tbl_Determination** establishes a many-to-many relationship between **Assertions** and **Specimens**, via the Foreign Keys *AssertionID* and *SpecimenID*. As indicated in the diagram, all **Specimens** must be qualified by at least one **Determination** (even if only to Kingdom). The critical point about this approach to **Specimen Determinations** is that it establishes the "determiner(s)" as author(s) of a **Reference**, which provided an **Assertion** with reference to a **Protonym**. When cited in publications, specimens are assignable to the corresponding **Assertion** for the identified Taxon Name within the publication. When specimens are determined directly (*i.e.,* not in the context of a published **Reference**), then the Determiner(s) serve as author(s) to a **Reference** of type "Determination" (with a *Date* value corresponding to the date on which the **Determination** was made). By allowing a broad interpretation of a **Reference** as fundamentally an instance

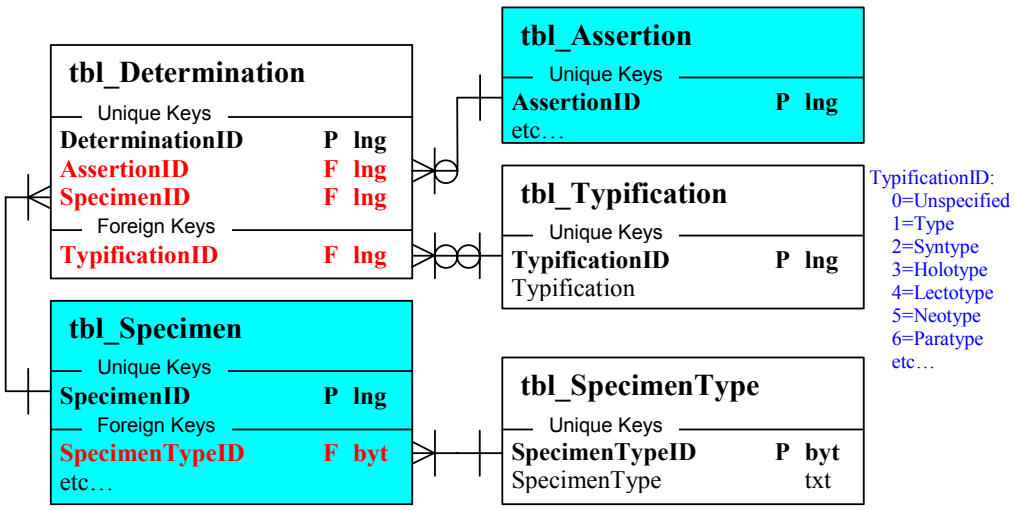

Figure 9. Physical data model for Determinations of Specimens.

of date-stamped **Agent(s)**, the *Taxonomer* data structure allows citations of specimens in publications and direct specimen **Determinations** to be handled identically: via **Assertions**. In both cases, the logic of the data model is consistent with the true informational content: **Specimens** are determined by **Agents** to belong to a Taxon Concept (as represented by a Taxon Name), at a particular point in time.

The Determination data model also includes a very simple way of representing **Specimens** as Types of Taxon Names, via the *TypificationID* Foreign Key to **tbl_Typification**. This link is established only when the **Reference** of the linked **Assertion** formally establishes the linked **Specimen** as a nomenclatural Type of the **Protonym** represented by the linked **Assertion**.

It should be underscored that all linkages between taxa and information content linked to **Specimens** (*e.g.,* Character data, geographic distributions, images, etc.) occur via the **tbl_Determination** relationships (see Figure 2).

Taxon Excerpts

One of the components related to both **References** and **Assertions** not shown in Figures 4 or 6 is **tbl_Excerpt**. The function of this table is to record quoted excerpts from **References**. As indicated in Figure 10, the table has six attributes in addition to its surrogate Primary Key. Three of these attributes are Foreign Keys. *ReferenceID* links to the **Reference** in which the **Excerpt** appeared. *LanguageID* indicates what **Language** the **Excerpt** was originally written in. *TranslatorID* links to the Agent who served the role of translator, if the **Excerpt** was translated to a different **Language** from its original. The three non-key attributes include *ExcerptType* (a general category of the **Excerpt** content or type; examples shown in blue text), *Excerpt* (the actual quoted text), and *Pages* (the specific Page[s] on which the **Excerpt** occurred within the **Reference**).

Although **tbl_Excerpt** is intended for very general application, a specific link to **tbl_Assertion** is established via **tbl_AssertionExcerpt**.. There are no other attributes of this table, other than the respective Foreign Keys *AssertionID* and *ExcerptID*. The purpose of this table is to establish a link between any specific **Protonym** and a free-form text quote related to the **Protonym** as it appears in the **Reference**. The domain of *AssertionID* within **tbl_AssertionExcerpt** is restricted to those **Assertion** instances that link to the same *ReferenceID* that the corresponding **Excerpt**

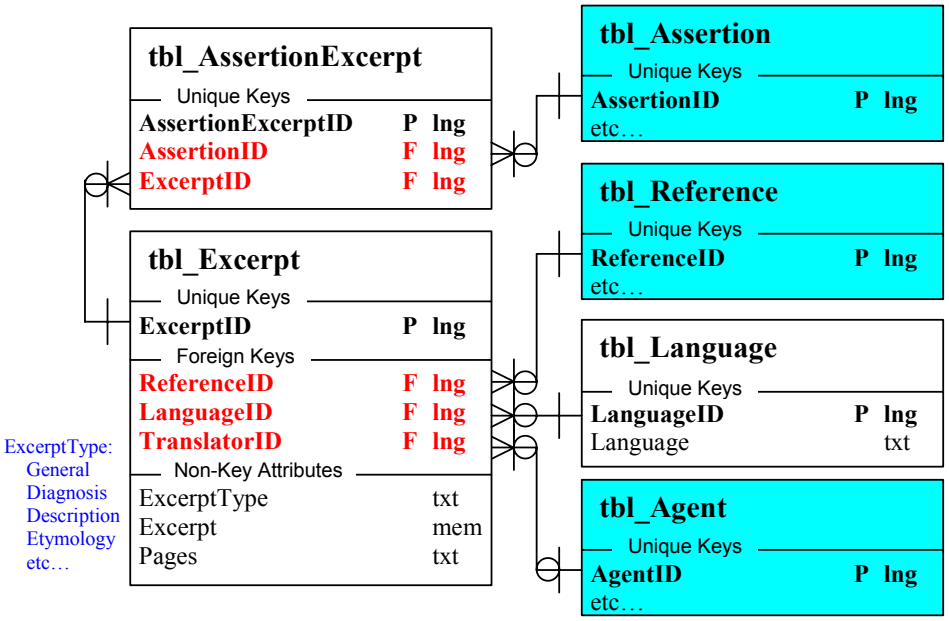

Figure 10. Excerpts of References, and their relations to Assertions.

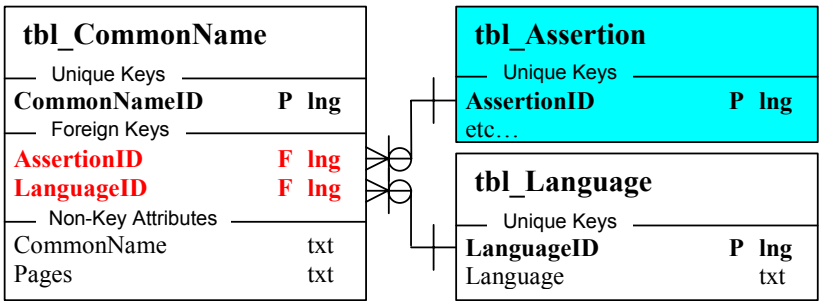

Figure 11. Physical data model for CommonNames.

instance links to. This general structure is extremely useful for a wide variety of taxonomic purposes, such as quoting entire sections of a taxon account as it appears within a reference (*e.g.,* Diagnosis, Etymology, etc.), quotes that describe relationships among several taxa, or other textual information that appears in a **Reference**.

Common Names

Although **TaxonRank** values within the range of 90-99 are reserved for informal taxon name designations not governed by Codes of nomenclature, these are not intended for "Common" or "Vernacular" names of organisms. It would be technically possible to treat such common names as yet another subtype of **Protonyms**, but this would require the cumbersome task of designating a single instance of each unique common name as the "original" instance (*i.e.,* **Protonym**). Moreover, extensive homonymy and a lack of a consistent hierarchy structure of common names justifies their treatment in a different way from more rigid taxonomic nomenclature.

Although not serving as the basis of **Assertions** themselves, the link between common names and scientific names is established via **Assertions**. Figure 11 illustrates that **tbl_CommonName** has two Foreign Keys: *AssertionID*, and *LanguageID*. By linking a **CommonName** to an **Assertion**, the **Reference** in which that **CommonName** appeared is automatically included (as linked to the **Assertion**). *LanguageID* specifies what **Language** the **CommonName** is representative of. The actual text string is stored in **CommonName**, and the *Pages* attribute is provided to indicate what page(s) the **CommonName** appears on within the associated **Reference**.

General Data Management

Several core components of the *Taxonomer* data model apply more or less equally to the entire model as a whole. These components manage general information that may have relevance to a wide variety of data entities, including those described herein. Three such general components are described here, and illustrated in Figure 12.

One of general components is **tbl_CodeNumber** and associated tables. Many core entities have numbers or other codes associated with them. Vouchered **Specimen** objects are assigned to various Catalog Numbers, Collector Numbers, Accession Numbers, and other codes. **References** may have Call Numbers, Reprint Numbers, or Accession Numbers. **Agents** are sometimes referred to by some sort of code or number (*e.g.,* Social Security Numbers, Employee Numbers, etc.). Even Taxon Names can have Code Numbers assigned to them (*e.g.,* the Taxonomic Serial Number [TSN] assigned to Taxon Names by ITIS). Many of these entities have multiple versions of a number that can change over time (*e.g.,* vouchered **Specimens** transferred from one natural history collection to another). Rather than sprinkle **CodeNumber** and **CodeNumberSeries** attributes across many of the core tables in the *Taxonomer* data model, a generalized **CodeNumber** documentation system has been implemented.

As shown in Figure 12, **tbl_CodeNumber** is linked to **tbl_CodeNumberType** and **tbl_CodeNumberSeries** via the two Foreign Keys, *CodeNumberTypeID* and *CodeNumberSeriesID*. **tbl_CodeNumberType** classifies the CodeNumber according to its general type (as shown in the blue text list).

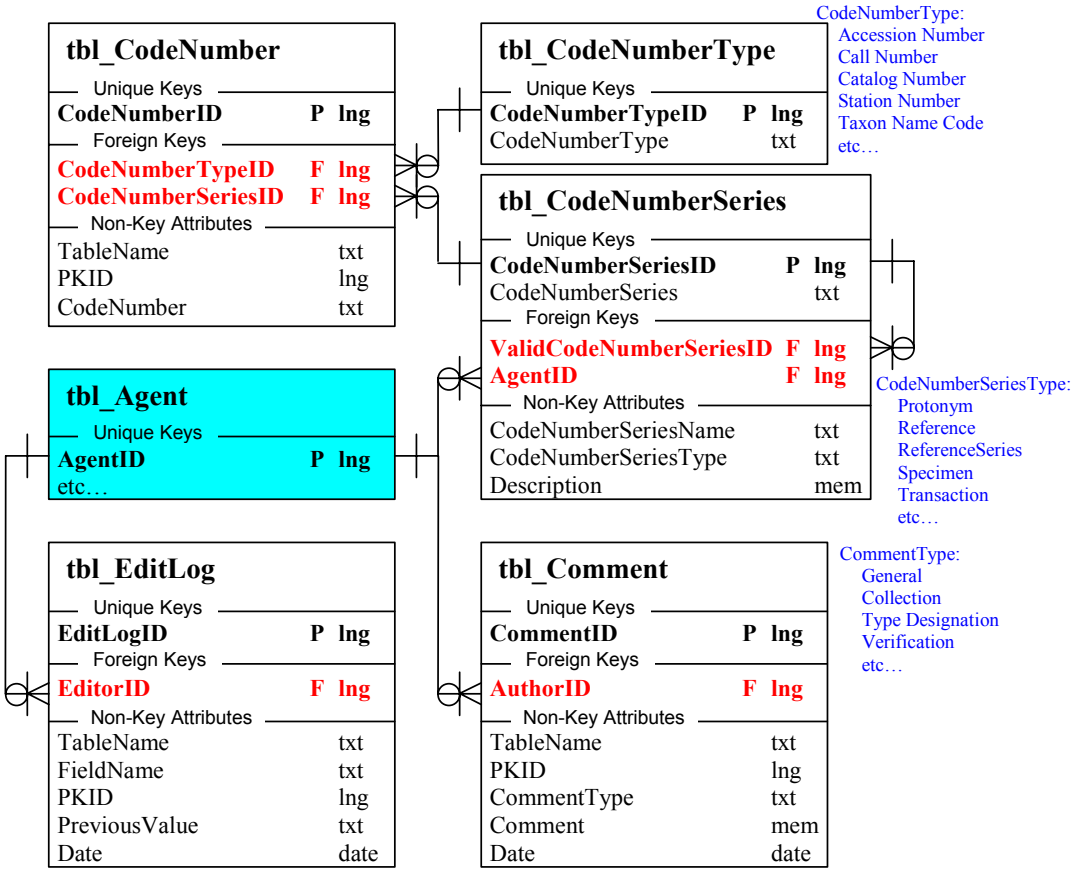

Figure 12. General data management tables.

tbl_CodeNumberSeries identifies the specific series of **CodeNumbers** (*e.g.,* a particular catalog number series in a **Specimen** collection). The *CodeNumberSeries* attribute is intended for the short, unique identifier of a number series, such as an institutional acronym for a **Specimen** collection (*e.g.,* "BPBM" for B. P. Bishop Museum; "CAS" for the California Academy of Sciences; etc.). The recursive Foreign Key *ValidCodeNumberSeriesID* is used primarily to track the changes in *CodeNumberSeries* values for a given number series (*e.g.,* when the British Museum of Natural History [BM(NH)] changed to The Natural History Museum [NHM]). The AgentID Foreign Key links to the **Person** or **Organization** that owns or created the **CodeNumberSeries**. *CodeNumberSeriesName* is the full-text name of the number series. *CodeNumberSeriesType* categorizes the **CodeNumberSeries** according to the core element that it refers to. In some cases, a **CodeNumberSeries** may apply to more than one core element (*e.g.,* "Call Numbers" can apply to both **References** and **ReferenceSeries**), in which case both values are entered in the *CodeNumberSeriesType* field, separated by a semicolon. The *Description* attribute allows for a longer text explanation of what the **CodeNumberSeries** is used for, and what its constraints and informational content are (if any).

The first two non-key attributes of **tbl_CodeNumber** are *TableName* and *PKID*. These generalized attributes are used in this table and in the two tables described below (**tbl_Comment** and **tbl_EditLog**), and serve the function of providing basic context to the values in other attributes of the table. *TableName* is entered as the name of the table in which the numbered instance appears (but without the "**tbl_**" prefix). For example, if the **CodeNumber** is a catalog number of a specimen, the value of *TableName* would be "Specimen." If a Social Security Number, the value would be "Person"; although it could also be "Agent," in this case, because **tbl_Person** is a

subtype of **tbl_Agent** with identical corresponding surrogate Primary Key values. The *PKID* attribute is used to record the surrogate Primary Key value of the instance in the table indicated by *TableName* to which the **CodeNumber** applies. Together, *TableName* and *PKID* effectively represent a Foreign Key (of sorts) to any instance of any Table in the *Taxonomer* application database.

The final attribute of **tbl_CodeNumber** is *CodeNumber*, which stores the actual value of the CodeNumber itself. This attribute is a text field, rather than a numeric field, to allow for textual or alphanumeric *CodeNumber* values.

The second general data management component shown in Figure 12 is **tbl_Comment**. This table stores any sort of free-form textual comment that can, as with **CodeNumbers**, be applied to any instance of any table in the *Taxonomer* application. **Comments** differ from **Excerpts** in that they are usually created by the database user to record meta-information regarding the nature of the data as contained in the database itself, whereas **Excerpts** generally exist outside the context of the database. As with *CodeNumbers*, this generalized approach was taken to consolidate **Comments** into a single table, rather than sprinkle various "Comments" and "Remarks" attributes throughout the various tables. *AuthorID* is a Foreign Key to **tbl_Agent**, indicating the **Person** or **Organization** who authored the **Comment**. *TableName* and *PKID* serve the same function they do for **tbl_CodeNumber**. CommentType allows general categorization of **Comments**. Although examples are shown in the blue text list in Figure 12, values for this field are defined within the context of the *TableName* to which they apply, and other database user needs. *Comment* is a long-text field that stores the text of the **Comment** itself. The final attribute of **tbl_Comment** is *Date*, which records the date and time at which the **Comment** was created. Although technically the combined attributes of *AgentID* and *Date* within **tbl_Comment** could be treated instead as a **Reference**, **Comments** are taken to be more ad-hoc annotations to the database, not really acquiring the status of a **Reference**. The model could, of course, be modified to have a single Foreign Key link to **tbl_Reference** instead of the *AgentID* and *Date* attributes, but this would not only create potentially enormous numbers of additional **References**, but would also obscure the distinction between **Comments** and **Excerpts**.

The *Taxonomer* application incorporates a very simple table called **tbl_EditLog** to record data edit history for all values of all attributes of all tables (except "Cheat" fields). It includes the following seven attributes: *EditLogID* (long-integer surrogate Primary Key field); *EditorID* (Foreign Key to **tbl_Agent** indicating the database user who made the edit); *TableName* (same function as in **tbl_CodeNumber** and **tbl_Comment**); FieldName (analogous to *TableName*, except identifies the field within a table for which the data were modified); *PKID* (same function as in **tbl_CodeNumber** and **tbl_Comment**); *PreviousValue* (the value of the specified field in the instance indicated by *PKID* before it was edited or deleted); and Date (a date and time-stamp for when the addition, edit, or deletion occurred). Whenever a new record is added to any table, an instance of **tbl_EditLog** is created with the appropriate values of *TableName*, *PKID*, *EditorID*, and *Date*; an asterisk ("*") is entered in *FieldName* (indicating all fields in the table), and the text "ADDED" is entered into *PreviousValue*. When a record is edited, the appropriate values of *TableName*, *FieldName*, *PKID*, *EditorID*, and *Date* are entered, and *PreviousValue* is set to the value of the edited field as it was before being edited. When a record is deleted, all non-null field values are recorded as they would be for a data edit, and an additional instance of tbl_EditLog is created with appropriate values of *TableName*, *PKID*, *EditorID*, and *Date*; an asterisk ("*") in *FieldName*, and the text "DELETED" in *PreviousValue*.

Consolidating all data edit history information into a single table eliminates the need to sprinkle attributes such as "CreatedBy," "CreateDate," "EditedBy," and "EditDate" across many different tables. In a sense, **tbl_EditLog** serves the same function as a the transaction log incorporated into many

more sophisticated relational database applications.

Discussion

A fundamentally important characteristic of the relational data model proposed herein is that major subsets of the data are modularized somewhat hierarchically, such that "lower" modules are entirely independent of "higher" modues. For example, the entities clustered together in Figure 3 that accommodate **Agent** data serve their function of storing and organizing relational data about **Agents** independently of the "higher-module" data subsets that link to **Agents** in some way (*e.g.,* as authors of **References**, collectors of specimens, etc.). **References**, representing a slightly higher module, depend on the existence of **Agents** (to represent Authors), but are entirely independent of still higher modules (*e.g.,* **Assertions**).

This article describes in detail the aforementioned three modules (**Agents**, **References**, and **Assertions**), and provides examples of how **Assertions** can serve as a handle to a Taxon Concept, which can be utilized in a variety of ways via additional data modules (*e.g.,* **AssertionRelations**, **Determinations**, etc.). The main emphasis of this article is to demonstrate how, with a broad interpretation of a **Reference** as any documented information provided by **Agent(s)** at a particular point in time (*i.e.,* date-stamped **Agents**), the concept of a **Protonym** as a subtype of **Assertion** can provide a self-contained and highly generalized approach to representing Taxon Names and Taxon Concepts.

Indeed, the **Assertion** is here regarded as a fundamental unit of taxonomy, and the general "currency" of information management concerning taxonomic entities. Although **Assertions** generally represent subjective treatments of Taxon Names in relation to the scope of living organisms circumscribed by those Names, the **Assertion** instances themselves are objective entities. While one may disagree with "AuthorX" that "SpeciesA" should be treated as a junior synonym of "SpeciesB," there is usually no ambiguity in the fact that "AuthorX" did indeed assert that "SpeciesA" should be treated as such (within the context of a **Reference**). An **Assertion** instance constitutes the documentation of that fact, and can be seen as a representation of the Taxon Concept explicitly stated or indirectly implied within the corresponding **Reference**.

Although many of the ideas and concepts presented here are not new, the basic approach to modeling information as it applies to Taxon Names and Taxon Concepts differs from other taxonomic data models. Comparisons with specific data models are made below.

Comparison with the *HICLAS* model

The HICLAS (HIerarchical CLAssification System) data model (Zhong et al., 1996; 1999) was among the earliest efforts to distinguish taxon names from taxon concepts and provide for multiple taxonomic views. Their definition of "Classification" was somewhat similar to an **Assertion** (in that both represent the treatment of a taxon or taxa, as would be the case for the full set of **Assertions** linked to a particular **Reference**). However, they restricted the scope of **References** that can provide Classifications to exclude checklists and certain other scientific or non-scientific citations of Taxon Names, even though such **References** likely represent a Taxon Concept (though generally less explicitly defined within the **Reference** itself). The *Taxonomer* model can enforce similar restrictions (and much more flexibly) by filtering on *ReliabilityID* of **tbl_Assertion**.

Closer to the essence of an **Assertion** is what Zhong et al. (1996; 1999) defined as a "Taxon View." Taxon Views are represented by four elements: Taxon Name, Author or Authority, Year, and Publication Number. Taxon Name is comparable to a **Protonym**, and the other three elements are all attributes of a **Reference** as linked to an **Assertion** (Author in the case where an **Assertion** is also the **Protonym**, and Authority in the cases of subsequent citations of a **Protonym**). Thus, these four elements are contained within the combined values of *ProtonymID* and *ReferenceID* as attributes of **tbl_Assertion**. It should be noted that whereas the "Publication Number" of *HICLAS* uniquely identifies the associated

Reference, the "Taxon Name" is somewhat more ambiguous, given both problems of homonymy and alternate spellings of what otherwise constitutes the same Name.

The equivalent of "Taxon-View Groups" of *HICLAS* are easily obtained from the *Taxonomer* model described here by filtering **tbl_Assertion** by a single *ProtonymID* value. The "Primary" and "Secondary" taxon views of *HICLAS* are identifiable by the conditions of *ProtonymID=AssertionID* and *ProtonymID≠AssertionID* (respectively) within **tbl_Assertion**. The Parent/Child Taxon View of HICLAS is represented by the recursive series of *ParentAssertionID* within **tbl_Assertion**, for a given *ReferenceID*. In the *Taxonomer* model, such links are generally only applied in cases of explicit referral. However, the word "explicit" as used here is somewhat liberal, requiring only the appearance of both parent and child names within the **Reference**, and some form of unambiguous representation of a hierarchical relationship between the two. Absent the explicit occurrence of a text-string name within a **Reference**, however, no implied parent/child relationships can be assumed. For example, if a **Reference** discusses a family of fishes, it is generally implied that the author regarded the Family to belong to at least the Kingdom "Animalia," if not something more specific. However, unless the **Reference** explicitly includes "Animalia" in its text, no **Assertion** instance should be created for that **Reference** linked to the **Protonym** of Animalia (leaving nothing for a *ParentAssertionID* to link to).

The *HICLAS* model defines seven "operations" to establish Lineages among Taxon Views: "Origination," "More," "Merge," "Partition," "Promotion," "Demotion," and "Recognition." Origination is represented in the *Taxonomer* model by a **Protonym**. The "Move" operation represents a lateral transfer of a taxon within the same rank, but under a different parent. Certainly treatments of a given **Protonym** by different **References** as belonging to different "Parent" taxa are adequately accommodated by the structure of **tbl_Assertion** (specifically via *ParentAssertionID*), and as such the "movement" of a **Protonym** among different Parent taxa over time can be easily documented. However, this notion of a "Move" operation is not treated as a meaningful informational entity within the *Taxonomer* model, for two reasons. First, the "movement" of a taxon concept from one Parent to another implies that some sort of tangible and universal entity changes its nature upon the publication of a new taxonomic treatment. From the perspective of the *Taxonomer* model, no real entity "moved"; rather, a common entity (a **Protonym**) was represented in a different hierarchical context. Secondly, the circumscription of a taxon involved with a "Move" operation doesn't actually change: it still is implied to contain exactly the same scope of living organisms that it did before the "Move." As such, this operation only involves change in perceived taxonomic affinities, not a change in the scope or definition of the Concept itself. Nevertheless, the *Taxonomer* model does accommodate tracking of the directionality of such lateral "Move" operations, through an instance of **tbl_AssertionRelation** joining a later **Assertion** (with one Parent) to an earlier **Assertion** (with a different Parent), via a **RelationType** of "congruent."

The "Merge" operation is also supported by sets of **tbl_AssertionRelation** instances. A single value of *AssertionID* in this table may be represented by multiple instances with different values of *RelatedAssertionID*, each indicated as being of **RelationType** "includes." The same applies for a "Partition" operation, except in that case the **RelationType** would be "included in."

The "Promotion" and "Demotion" operations are dealt with in *Taxonomer* in the same way that the "Move" operation is; that is, within **tbl_Assertion** (via different values if *TaxonRankID*). As with the other operations, the directionality and scope of these changes can be represented via **tbl_AssertionRelation**. Similarly, the "Recognition" operation is easily accommodated by an instance in **tbl_AssertionRelation**, established with a **RelationType** of "congruent."

Zhong et al. (1999) discussed the differences between hierarchies based on nomenclature, and hierarchies based on

phylogenetic analysis. The revised (1999) *HICLAS* model endeavored to accommodate both kinds of hierarchies. Although the *Taxonomer* model could likely accommodate phylogenetic representations with relatively minor modifications, its intended purpose at the present time is focused on nomenclatural classifications.

Comparison with the *Berlin* (*IOPI*) model

Berendsohn (1997) described the "IOPI" (International Organization for Plant Information) taxonomic data model, and Berendsohn et al. (2002) and Berendsohn et al. (2003) updated it and referred to it as the "Berlin" model. There are many fundamental similarities between the *Berlin* model and the *Taxonomer* model. Indeed, the virtually independent convergence on such similar data management solutions suggests that some level of optimality may be approached, especially when considering that the *Berlin* model was developed primarily around the needs of botanical taxonomy, whereas *Taxonomer* was driven more directly by zoological taxonomy.

Berendsohn et al. (2003:15) wrote: "The taxonomic model has to incorporate nomenclatural rules and the traditional taxonomic relationships (synonymy, taxonomic hierarchy, etc.). In addition, it has to be capable of representing different taxonomic views in order to enable the system to express arbitrary relationships between potential taxa." The *Taxonomer* model achieves the former through **tbl_Assertion** and its related tables, and achieves the latter through **tbl_AssertionRelation**.

Perhaps the biggest difference between the two models is how Taxon Name entities are treated. As described in detail above, in *Taxonomer* information concerning original descriptions of taxon names is embedded within the **Assertion** and **Protonym** tables. In the *Berlin* model, Name data are stored in an entirely different set of tables, as explained in Berendsohn et al. (2003), to more strictly separate nomenclatural data from potential taxon data. This separation is accommodated in the *Taxonomer* model in that links to Taxon Concepts (potential taxa) are made to **tbl_Assertion**, whereas links intended to only represent the nomenclatural components are made to **tbl_Protonym**. Although the specific approach to managing various aspects of Name information (*e.g.,* unnamed taxa, cultivars, nothotaxa and hybrids formulae, etc.) are quite different in the two models, both are capable of managing very similar informational content (with the *Taxonomer* model being somewhat more normalized, relying instead on various "Cheat" fields to improve concatenation performance of multinomials, and relying more heavily on business rules embedded within the application tier to manage different information elements applied differently to Names of different taxonomic rank). The functions of the Berlin model's table "RelName" (the relationships among names) are accommodated in several ways. Relationships between subsequent treatments of a Name and its basionym are accommodated by the *ProtonymID* attribute of **tbl_Assertion**. Other such relationships (*e.g.,* 'is later homonym of') can be accommodated in **tbl_ObjectiveStatus** (in the context of other relationships included within the corresponding Assertion). Still others are accommodated by **tbl_HybridAssertion** and *TypeProtonymID* of **tbl_Protonym**. The Berlin model also includes the "NomStatusRel" table which serves essentially the same function as **tbl_ObjectiveStatus** as described herein. Both tables are used to categorize the nature of a Code-mandated relationship between Taxon Names, as asserted by a **Reference**. Many of the nomenclatural tables in the Berlin model include "…RefFK" and "…RefDetailFK" linkages to "Reference" and "RefDetail" tables. By using Assertions as the unit of nomenclatural information management, the *Taxonomer* model consolidates those linkages into a single link (*i.e.,* via the *ReferenceID* attribute of **tbl_Assertion**).

Another difference between the *Berlin* model and the *Taxonomer* model is how nomenclatural authors are tracked. The *Berlin* model explicitly defines "teams" of authors, which are linked directly to Name entities. The *Taxonomer* model derives authors of Taxon Names via the associated *ReferenceID* of a **Protonym's** corresponding **Assertion**. This eliminates the need for additional relationships between **Agents** and Names, and between **References** and Names (see

earlier discussion under the "References" section describing the use of the "Ex" **AuthorType** and the use Sub-References for delineating original descriptions from their containing **References**, when necessary). The *Taxonomer* model does not establish an entity to represent an "Author Team," but one could easily be derived from the set of Agents linked to any particular **Reference** via **tbl_ReferenceAuthor**. However, the need to establish this somewhat artificial entity (which seems to be based solely on the desire to establish direct relationships between each Name and its individual authors) is obviated by the way in which Taxon Names derive their authorships within the *Taxonomer* model. Standard botanical abbreviations for authors are accommodated in the Taxonomer model via the **CodeNumber** components, described earlier (which can accommodate any number of abbreviations based on any number of defined abbreviation standards). The function of the "RelAuthor" table in the Berlin model is essentially duplicated by *ValidAgentID* in the *Taxonomer* model.

The bibliographic components of the Berlin model are functionally similar to the Reference components described herein. The "Reference" table of the *Berlin* model is analogous to **tbl_Reference** described herein; and the "RefCategory" table of the *Berlin* model is analogous to **tbl_ReferenceType**. Most differences between the two approaches are in detail only, and largely stem from the adherence of the *Taxonomer* model to the structure used by EndNote® 7 software. One aspect of the *Berlin* model that is more robust that the *Taxonomer* model is the "RefDetail" table, which pinpoints positions within **References** in a more normalized way than the "Pages" attributes of several *Taxonomer* tables. As discussed earlier, future versions of *Taxonomer* may include a more robust management scheme analogous to "RefDetail."

At the heart of the *Berlin* model is the "Potential Taxon" (Berendsohn, 1995; 1997; Geoffroy & Berendsohn, 2003), which is almost identical to the **Assertion** as defined herein (more specifically referred to in Berendsohn et al., 2003, as a "Taxonym"). Both approaches establish the intersection of a **Reference** and a Taxon Name as the handle to a taxon concept. Both approaches also use this intersection instance as the basis through which other factual information is linked to taxa. The main difference between the two structures is that the *Berlin* establishes a unique set of "Status" alternatives for each name, whereas the *Taxonomer* embeds this information into the same **Assertion** instance (see further discussion below). One apparent limitation of the Taxonomer model is that only one "kind" of synonym (*i.e.,* a direct nomenclatural synonym) can be defined within **tbl_Assertion**. As pointed out by Berendsohn et al. (2003), other "kinds" of synonyms ('partial synonym" and 'pro parte synonym') "…are actually cases of concept synonomy…" (p.37), but explicitly stated within the "taxonym"/**Assertion Reference** itself. Such "**Assertion** mapping" is accommodated by the *Taxonomer* model via **tbl_AssertionRelation**. In such cases where the **Reference** contributing the **Assertion** explicitly states such concept synonyms, the *SourceReferenceID* attribute of **tbl_AssertionRelation** links to the same **Reference** as indicated by the corresponding *ReferenceID* value in the associated **Assertion** instance. In this way, the *Taxonomer* model clearly separates nomenclatural synonymies from concept synonymies. The "RelPTaxon" table of the *Berlin* model serves three functions, that are accommodated by the *Taxonomer* model by (respectively), *ValidAssertionID* of **tbl_Assertion** (for traditional synonymy); *ParentAssertionID* of **tbl_Assertion** (for hierarchical classification), and **tbl_AssertionRelation** and its associated **tbl_RelationType** (for concept synonymy). The reason that the different functions are handled differently in the *Taxonomer* model is that the former two are best addressed on an intra-**Reference** basis; whereas the latter involves inter-**Reference** relationships of taxon concepts (note that intra-**Reference** linkages via *ValidAssertionID* and *ParentAssetrtionID* simultaneously reflect nomenclatural relationships *and* concept relationships, as they necessarily are the same within a single **Reference**; but even these can be further qualified via an instance in **tbl_AssertionRelation**). The main weakness of the *Taxonomer* model in this regard

is that it does not have a robust structure for accommodating misapplied names. As discussed in the "Limitations" sub-section of the "Taxa" section of this article, such information can be accommodated indirectly, and may be more directly addressed in future versions via a more robust dedicated table structure.

"Cheat" fields defined herein are somewhat analogous to "Cache fields" of Berendsohn et al. (2003:17) in that they enhance performance, but differ in that they can only be created by the application from the atomized components (*i.e.,* they are strictly derived fields). By contrast, "Cache fields" in the Berlin model are also used to store imported concatenated data prior to parsing into more atomized fields. These differences only affect application-tier issues, and data importing protocol; they have essentially no bearing on the core data structure.

Finally, Kusber et al. (2003) describe an extension of the Berlin model to robustly model taxonomic typification. The *Taxonomer* model, by contrast, includes only rudimentary typification documentation (via *TypeProtonymID* of **tbl_Protonym**, and *TypificationID* of **tbl_Determination**). A more robust approach to managing typification is planned for a future version of the *Taxonomer* model, but it is worth noting that such an enhancement would primarily be in the form of additional "modules," without substantial modification to the core *Taxonomer* data structure described herein.

Comparison with the *Prometheus* model

The *Prometheus* taxonomic data model (Pullan et al., 2000; Raguenaud, 2002) is intended to provide a mechanism for objectively defining the scope and extent of taxonomic circumscriptions, by way of **Specimens** as reference points. Like the previous two discussed models (*HICLAS* and the *Berlin* model), it has botanical origins. Like the *Taxonomer* model, the *Prometheus* model attempts to structure data according to how it is actually used for taxonomic activities.

As emphasized by the *Prometheus* model, specimens are the only objective means to establish congruency (or lack thereof) between any given pair of Taxon Concepts (see earlier discussion under "Specimen-Resolution Circumscriptions"). The Taxonomer data model supports such Specimen-Resolution circumscription definitions by virtue of the fact that **Determinations** are linked to **Assertions**. This linkage allows direct indexing of specific **Specimens** as definitive markers to the biological (*i.e.,* real-world) scope of the taxon circumscription represented by the **Assertion** instance. When more than one **Reference** includes **Determination** instances for the same pool of **Specimens**, the respective scope of the corresponding sets of **Assertions** for each **Reference** can be objectively compared.

Because of the complexity of mapping instances of "Potential Taxa" to the physical specimens upon which they were based, the "Potential Taxon" is portrayed by Berendsohn (1997) and Pullan et al. (2000) as a "compromise" method for managing taxon concepts. In contrast, I do not see **Assertions** as representing any form of "compromise" at all, but rather as a different basis of information indexing. That some **References** fail to explicitly "anchor" their implied Taxon Concepts to biological reality in the form of **Specimen** citations does not negate the fact that the authors of the **Reference** had a clear Taxon Concept in mind when they represented it by a Taxon Name. **Assertions** of such **References** should not, therefore, be excluded from the pool of potential taxonomic concepts, simply because their concepts cannot be objectively scoped or cross-referenced to other **Assertions**. Indeed, it could be safely argued that many (a majority?) of **References** that *do* cite specific **Specimens**, do *not* draw from a sufficiently large and overlapping pool of **Specimens** as cited in other **References** establishing **Assertions** for the same set of **Protonyms**. The *Taxonomer* model was intentionally designed to exploit **Specimen** citations when they exist (via the use of **Assertions** as the taxonomic link to **Determinations**), but not to exclude other **Assertions** (that lack extensive specimen citations) from the overall pool of managed taxonomic information.

Comparisons between the *Prometheus* model and the *Taxonomer* model are

necessarily limited, given the difference in fundamental data structure, and to some extent, the different intended purposes of each. Nevertheless, some comparisons can be drawn. Like the *Berlin* model, the *Prometheus* model goes to great lengths to distinguish nomenclatural information from 'classification' (circumscription) information. The *Taxonomer* model rigorously (but subtly) maintains a distinction between Name entities and Concept (circumscription) entities, without extensive de-normalization or duplication of the data structure. This is accomplished simply by the implied rule that links established via *AssertionID* values apply to both circumscriptions and nomenclature (the latter provided automatically and simultaneously), whereas links established via *ProtonymID* are exclusively nomenclatural (even though **Protonyms** are subtypes of **Assertions**, which contain an implied circumscription). The *Prometheus* model defines the two separate entities of "Nomenclatural Taxa" and "Circumscribed Taxa"; each with its own set of links to ranks, publications (and associated authors, either by extension or directly), specimens, and hierarchical recursion. These roughly correspond to the **Protonym** and **Assertion** entities in the *Taxonomer* model. Separate links to "Rank" and "Publication" (=**Reference**) entities in the *Prometheus* model are consolidated in the *Taxonomer* model. The link between Circumscribed Taxa and Specimens in **Prometheus** are comparable to **Determination** instances of *Taxonomer*, but the direct link between Nomenclatural Taxa and Specimens (via typification) in *Prometheus* does not exist in *Taxonomer* (instead, **Protonyms** are connected to their type **Specimens** via **Determinations** by the corresponding **Assertion** instance that established the typification). The Rejection/Conservation Status of Nomenclatural Taxa (and associated entities) in *Prometheus* is most closely emulated by **ObjectiveStatus** components of *Taxonomer*. Like the *Prometheus* model, the *Taxonomer* model does not treat the relationship between a Name and its homotypic Basionym (**Protonym**) by the same mechanisms that synonyms are established.

Pullan et al. (2000) cautioned against the use specimen determination labels for delineating circumscriptions, due to the fact that the determination is limited in temporal scope to the date on which it was applied to the specimen. However, this only potentially limits the extent to which such determinations can be objectively cross-referenced to published circumscriptions. By treating **Specimen Determinations** as a defined **ReferenceType**, the *Taxonomer* model allows such **Determinations** to stand on their own as representing Taxon Concepts, independent of published works citing the same specimens. In most cases, **Specimens** cited in publications will also have **Determination** labels associated with them, by the same or similar authors as the publication. This allows objective cross-verification of congruency among **Determination**-based circumscriptions and their published counterparts. While it is true that **Determinations** are technically dated on the day on which the **Determination** was applied to the **Specimen**, there are many cases when clusters of **Determinations** spanning a series of consecutive or near-consecutive dates can be logically consolidated. One example is when a taxonomist visits a Museum and establishes a series of **Determinations** within a span of several days or weeks. Another example is when a taxonomist borrows a series of **Specimens**, and returns them as a batch with new **Determinations**. Such sets of **Assertions** can be reliably cross-referenced as congruent using **tbl_AssertionRelation**.

The *Taxonomer* model differs from the *Prometheus* model in the way that "auxillary data" or "factual data" are joined to taxonomic components in that *Taxonomer* establishes such links via **tbl_Assertion**, whereas *Prometheus* links such data to Nomenclatural Taxa. This distinction is merely a result of the way *Taxonomer* establishes **Assertion** instances associated with the **Reference** instances that provide the auxiliary data (and establishes the link between the auxiliary data and the taxonomy via these **Assertions**).

Comparison with the *Nomencurator* model

The most recent of the published data models for managing multiple taxonomic views is *Nomencurator* (Ytow et al., 2001;

2002). Conceptually, there are many similarities between the *Nomencurator* model and the *Taxonomer* model. The "Publication" entity of *Nomencurator* is functionally equivalent (though more restrictive) to the **Reference** entity of *Taxonomer*. Ytow et al. (2001) describe two types of links within Publications: internal and external. These correspond to what are referred to in this article as "*intra*-**Reference**" links, and "*inter*-**Reference**" links, respectively. The term "taxonomic opinion" as used in Ytow et al. (2001) is conceptually identical to the **Assertion** described herein (*i.e.,* "…the term 'taxonomic opinion' will be used to describe the taxon concept as it existed for an author at the time of publication. A taxonomic opinion can be identified without ambiguity by specifying a pair of tangible objects; the name as printed and the publication in which it appeared." p.84).

The *Nomencurator* model is fundamentally based on a three-layered approach to defining informational units. The "instance" layer is defined by Ytow et al. (2001:84-85) to represent "specimens or lower taxa," the "taxon layer" refers to Taxon Concepts, and the "name layer" refers to Taxon Names. The "name" layer in *Nomencurator* is directly comparable to the **Protonym** entity of *Taxonomer*, and the "taxon" layer in *Nomencurator* is directly comparable to the **Assertion** entity of *Taxonomer*. In *Nomencurator*, names are portrayed as "tags" linked to Taxon Concepts within the context of a publication. Similarly in *Taxonomer*, **Protonym** "tags" are linked to **Assertions** (via the *ProtonymID* Foreign Key of **tbl_Assertion**), in the context of the **Reference** linked to the same **Assertion** instances.

Of the "instance" layer, Ytow et al. (2001:84) write: "…an instance is a conceptual object, not a physical specimen." Ultimately, however, "lower taxa" are merely abstracted representations of implied sets of specimens, so specimens are the true conceptual foundation of the "instance" layer, even if abstracted conceptual entities (*i.e.,* "lower taxa") are used as surrogate instances for higher taxa. The distinction is important when examining how the *Taxonomer* model represents instances. In the case of taxa at super-specific ranks (and Species that are further divided into infra-specific ranks), the equivalent "instances" of a given "**AssertionX**" are the set of lower-rank intra-**Reference** (*i.e.,* internally linked) **Assertions** that themselves link back to "**AssertionX**" via *ParentAssertionID*. This relationship is recursive through all taxonomic ranks down to (but not including) the terminal infra-generic rank (*i.e.,* species or infra-specific rank). In such cases of "terminal" species and infra-specific taxa, the "instances" are derived from links to **Specimens** via **tbl_Determination**. The dichotomy between the structural treatment of instances for higher-rank Taxon Concepts and lower-rank Taxon Concepts is justified by virtue of the fact that **Specimen** entities are fundamentally distinct from lower-rank Taxon Concept entities. Indeed, when one considers (as described above) that lower-rank taxa, when treated as instances of higher-rank taxa, are merely infra-**Reference** surrogate abstractions of sets of physical **Specimens** (vouchered or not), the structure of the *Taxonomer* is logically consistent.

At the physical implementation level, some additional similarities between *Nomencurator* and *Taxonomer* are evident. In particular, the relationships between "Publications" and "Authors," and between "Authors" and "Affiliations," is nearly identical to the corresponding relationships in *Taxonomer* between **References**, **Agents**, and affiliations among agents (via **tbl_AgentAssociation**). Pursuing the physical implementation further, however, reveals that the two models diverge. Although the "NameRecord" entity of *Nomencurator* roughly corresponds with an **Assertion** ("…in essence the potential taxon concept…the data structure combination of the name and its publication…" p.89), the entity relationships are somewhat different. For instance, NameRecords link to Publications via "Appearances." Thus, **Assertions** represent a combination of NameRecords and Appearances. Annotations in the *Nomencurator* model serve functions addressed by **tbl_AssertionRelation** and **tbl_ObjectiveStatus** of the *Taxonomer* model.

Another difference between the *Nomencurator* model and the *Taxonomer* model worth mentioning is in how each model defines the scope of allowable **References**. The *Nomencurator* model, through its use of the term "Publication" restricts such instances to published **References**. Presumably, this would exclude, for example, an unpublished manuscript that is "in press," even though such a manuscript contains precisely the same information as it would after it is actually published. While this example can be (correctly) seen as "nit-picking," the problem is the existence of an essentially unbroken continuum from such a "mature" manuscript, downward through chapters in theses, rough drafts of manuscripts, correspondence among taxonomists expressing taxonomic opinions, researchers' notebooks, specimen determination labels, spoken words, and even (taken to the extreme) undocumented thoughts. Selecting a point along this continuum to limit the scope of a **Reference** is somewhat subjective and arbitrary, and does not necessarily correlate with taxonomic "reliability." One possible point of delineation would be the peer-review process of most scientific publications. However, this criterion would exclude many valuable forms of taxonomic information that are not subjected to peer review (*e.g.,* many published books). It would also exclude a wealth of potentially important and insightful unpublished information. Within the context of the *Taxonomer* model, I have chosen to delineate the scope of a **Reference** to include any "documented" instance of information as presented by one or more authors (including, but not limited to, publications). "Documented" in this context, can be roughly defined as any medium that can be represented in a broadly interpretable way via a standard digital format (*e.g.,* text, digital manuscripts in various formats, databases and spreadsheets, images, and potentially even audio and video recordings). The reason for using the "digitizable" standard in this context is that the topic here discussed relates to electronic (digital) databases, and in its ultimate incarnation would directly interface with digital representations of **Reference** sources. In any case, by defining the data model in such a broad way as to be more inclusive of different information sources, the user is always left with the option of filtering data output according to more restrictive criteria (*e.g.,* only those **Assertions** linked to **References** of **ReferenceTypes** flagged as *IsPublished*.) Conversely, restricting the scope of sources at the data model layer disallows the electronic capture of potentially useful information. Thus, the broader scope of **Reference** defined herein is seen as providing a more generalized approach to taxonomic data management.

Acknowledgements

I wish to thank Stan Blum for his willingness to engage in seemingly endless discussions about a wide range of topics related to bioinformatics, and for exposing me to the larger universe of biological database activities. Thanks are also due to Nozomi "James" Ytow and Jim Croft, who have generously provided general insights. The conversion of data from Cowie et al. (1995) into the *Taxonomer* data structure was funded in part by a grant from the Pacific Basin Information Node (PBIN), of the U.S. National Biological Information Infrastructure (NBII).Microsoft Access® and Microsoft SQLServer® are a registered trademark of Microsoft Corporation. EndNote® is a registered trademark of ISI ResearchSoft.

Literature Cited

ABRS [Australian Biological Resources Study]. 2003. Platypus: A database package for taxonomists. http://www.deh.gov.au/biodiversity/abrs/online-resources/software/platypus/index.html (3 September 2003).

Anonymous. 2002. VegBank taxonomic data models. Ecological Society of America. http://vegbank.nceas.ucsb.edu/vegbank/design/planttaxaoverview.html (23 July 2003).

Anonymous. 2003. EndNote 7…bibliographies & more made easy™. Thompson ISI Researchsoft, Carlsbad. 589 pp.

ASC [Association of Systematics Collections]. 1993. An information model for biological collections. Report of the Biological Collections Data Standards Workshop, August 18-24, 1992. http://www.nscalliance.org/bioinformatics/resources.asp (14 January 2004).

Berendsohn, W.G. 1995. The concept of "potential taxa" in databases. *Taxon* 44: 207-212.

Berendsohn, W.G. 1997. A taxonomic information model for botanical databases: the IOPI Model. *Taxon* 46: 283-309.

Berendsohn, W.G. (ed.) 2003. MoReTax: Handling factual information linked to taxonomic concepts in biology. Schriftenreihe für Vegetationskunde, Vol. 39. Federal Agency for Nature Conservation, Bonn, Germany. 113+xvii pp.

Berendsohn, W.G., M. Geoffroy, A. Güntsch and J.-L.Li 2002. The Berlin taxonomic information model. http://www.bgbm.org/biodivinf/docs/bgbm-model/ (8 July 2003).

Berendsohn W.G., M. Döring, M. Geoffroy, K. Glück, A. Güntsch, A.Hahn, W.-H. Kusber, J.-L. Li, D. Röpert and F. Specht. 2003. The Berlin model: a concept-based taxonomic information model. *In*: MoReTax: Handling factual information linked to taxonomic concepts in biology (ed. W.G. Berendsohn). pp. 15-42. Schriftenreihe für Vegetationskunde, Vol. 39. Federal Agency for Nature Conservation, Bonn, Germany. 113+xvii pp.

Bio-Tools.Net. 2003 TAXIS – Taxonomic information system. http://www.bio-tools.net/ (3 February 2003).

BIOGIS Consulting. 2003. BioOffice - Datenbank und geographisches Informationssystem. http://www.biooffice.at/ (23 September 2003).

Bisby, F. 2002. The quiet revolution: biodiversity informatics and the Internet. *Science* 289: 2309-2312.

Blum, S.D. 1996. The MVZ Collections Information Model. Conceptual and Logical Models. University of California at Berkeley, Museum of Vertebrate Zoology. (see http://www.mip.berkeley.edu/mvz/cis/)

Colwell, R.K. 2002. Biota: The biodiversity database manager. http://viceroy.eeb.uconn.edu/biota (26 May 2002).

CONABIO. 2003. Biótica Information System© Version 4.0. http://www.conabio.gob.mx/informacion/biotica_ingles/doctos/distribu_v4.0.html (31 March 2003).

Cowie, R.H., N.L. Evenhuis and C.C. Christensen. 1995. Catalog of the Native Land and Freshwater Molluscs of the Hawaiian Islands. Backhuys Publishers, Leiden. 248 pp.

Creighton, R.A. and J.J. Crockett. 1971. SELGEM: A system for collection management. *Smithsonian Inst. Information Systems Innovations*, 2(3):1-26.

Dessein, S. and P. Schols. 2003. MacTaxon. http://www.kuleuven.ac.be/bio/sys/ (23 September 2003).

Eschmeyer, W.N. 1990. Catalog of the genera of recent fishes. California Academy of Sciences, San Francisco: i-v + 1-697.

Eschmeyer, W.N. 1995. The role of taxonomic databases, with special emphasis on fishes. Chapter 19. *In:* Marine and Coastal Biodiversity in the Tropical Island Pacific Region. Volume 1. Species Systematics and Information Management Priorities (eds. J.E. Maragos, M.N.A. Peterson, L.G. Eldredge, J.E. Bardach, H.F. Takeuchi). pp. 333-338. Program on Environment, East-West Center, Honolulu, Hawaii. 424 pp.

Eschmeyer, W.N. 1998. The catalog of fishes. California Academy of Sciences, San Francisco. Vol. 3: 1821-2905.

ETI [Expert Center for Taxonomic Identification]. 2003. Linnaeus II 2.x. http://www.eti.uva.nl/Products/Linnaeus.html (23 September 2003).

Euzéby, J.P. 1997. List of bacterial names with standing in nomenclature: a folder available on the Internet (http://www.bacterio.cict.fr/). *Int. J. Syst. Bacteriol.*, 47: 590-592.

Filer, D.L. 2001. BRAHMS -- Botanical research and herbarium management system. *BioNET News* 8: 6-7. (see: http://www.bionet-intl.org/html/whatsnew/newsletters/newsletters.htm)

Francki, R.I.B., C.M. Fauquet, D.L. Knudson and F. Brown. 1990. Classification and nomenclature of viruses. *Archives of Virology Supplement* 2:1-445.

Geoffroy, M. and W.G. Berendsohn. 2003. The concept problem in taxonomy: importance, components, approaches. *In*: MoReTax: Handling factual information linked to taxonomic concepts in biology (ed. W.G. Berendsohn). pp. 5-14. Schriftenreihe für Vegetationskunde, Vol. 39. Federal Agency for Nature Conservation, Bonn, Germany. 113+xvii pp.

Gewin, V., 2002. Taxonomy: All living things, online. *Nature* 418: 362-363.

Godfray, H.C.J. 2002. Challenges for taxonomy. *Nature* 417: 17-19.

Gradstein, S.R., M. Sauer, M. Braun, M. Koperski and G. Ludiwig. 2001. TaxLink, a program for computer-assisted documentation of different circumscriptions of biological taxa. *Taxon* 50: 1075-1084.

Greuter, W., J. McNeill, R. Barrie, H.-M. Burdet, V. Demoulin, T.S. Filguerias, D.H. Nicolson, P.C. Silva, J.E. Skog, P. Threhane, N.J. Turland And D.L. Hawksworth (eds. & compilers). 2000. International Code of Botanical Nomenclature (Saint Louis Code) adopted by the Sixteenth International Botanical Congress, St. Louis, Missouri, July-August 1999. Regnum Vegetabile 138. Koeltz Scientific Books, Königstein. 474 pp.

Hoppe, J.R., E. Boos and G. Gottsberger. 1996. The database system SysTax - an aid for systematics and taxonomy and the management of botanical gardens and herbaria. *Albertoa* 4(9): 107-108.

Hoppe J.R. and T. Ludwig. 2003. SysTax - a database system for systematics and taxonomy. http://www.biologie.uni-ulm.de/systax/index-e.html (23 September 2003).

Humphries, J.M. 1994. MUSE: A tutorial and reference manual. Cornell University, Ithaca, New York. 80 pp. [Privately distributed document, PDF file available online at: http://www.biodiversity.uno.edu/muse/manual.pdf]

IBRC [Informatics Biodiversity Research Center, University of Kansas]. 2003. Specify: biodiversity collections management. http://usobi.org/specify/ (10 September 2003).

ICZN (ed.). 1999. International Code of Zoological Nomenclature. Fourth Edition. The International Trust for Zoological Nomenclature, 306 pp.

ITIS [Integrated Taxonomic Information System]. 2003. Standards and Database Documentation. http://www.itis.usda.gov/standard.html (19 April 2003).

KESoftware. 2003. KE Emu™ Electronic Museum. http://www.kesoftware.com/emu/index.html (23 September 2003).

Koperski M., M. Sauer, W. Braun and S.R. Gradstein. 2000. Referenzliste der Moose Deutschlands. *Schriftenreihe für Vegetationskunde* 34: 1-519.

Kusber, W.-H., K. Glück, M. Geoffroy and R. Jahn. 2003. Typification – an extension of the Berlin model. *In*: MoReTax: Handling factual information linked to taxonomic concepts in biology (ed. W.G. Berendsohn). pp. 57-70. Schriftenreihe für Vegetationskunde, Vol. 39. Federal Agency for Nature Conservation, Bonn, Germany. 113+xvii pp.

Lapage, S.P., E.F. Lessel, S.P. Oapage and P.H.A. Sneath. (eds.). 1992. International Code of Nomenclature of Bacteria and Statutes of the International Committee on Systematic Bacteri-

ology and Statutes of the Bacteriolog: Bacteriological Code, Revised edition. American Society for Microbiology. 189 pp.

Le Renard, J. 2000. TAXIS: a taxonomic information system for managing large biological collections. *In*: Abstracts. TDWG 2000: Digitizing Biological Collections. p. 18. Taxonomic Databases Working Group, 16th Annual Meeting, Frankfurt, 10-12 November 2000.

Lee, M.S.Y. 2000. A worrying systematic decline. *Trends Ecol. & Evol.* 15(8): 346.

Merriam-Webster. 1993. Webster's Ninth New Collegiate Dictionary. Merriam-Webster, Inc., Springfield. 1557 pp.

Mims, C. 2003. Endangered species, endangered science. *Zoogoer*. 32(4): 18-28.

Minelli, A. 2003. The status of taxonomic literature. *Trends Ecol. & Evol.* 18(2): 75-76.

Minnigerode, M.D. 1998. Tracy: a Herbarium Management System. http://www.csdl.tamu.edu/FLORA/input/inputsys.html (August 1998).

Moretzsohn, F. 2002. TaxonBank, proposal for a new online database for taxonomic research on type specimens. Proceedings of 2nd International Workshop of Species 2000 (1999), National Institute for Environmental Studies, Japan. pp. 142-147.

Morris, P.J. 1998. A data model for paleontological species level specimen based information. http://www.athro.com/general/ip/datamodel.html (16 Apr 1998).

Murphy, F.A., C.M. Fauquet, M.A. Mayo, A.W. Jarvis, S.A. Ghabriel, M.D. Summers, G.P. Martelli and D.H.L. Bishop. 1995 Sixth Report on the International Committee on Taxonomy of Viruses. Springer Verlag, Wien & New York. 586 pp.

Naskrecki, P. 2003. MANTIS: A manager of taxonomic information and specimens. http://140.247.119.145/Mantis/ (26 August 2003).

NCBI [National Centre for Biodiversity Informatics]. 2002. SAMPADA. http://www.ncbi.org.in/sampada/index.html (1 May 2002).

Nishida, G.M. (Ed.) 2002. Hawaiian Terrestrial Arthropod Checklist, Fourth Edition. Bishop Museum Technical Report No. 22. Hawaii Biological Survey, Bishop Museum, Honolulu, HI. 313 pp.

Simpson, J.A. and E.S. Weiner (eds.). 1989. The Oxford English Dictionary, Second Edition. Oxford University Press. 22,000 pp.

Pando, F. 2001. A Primer for BIBMASTER: A database application for nomenclature, literature and specimen management, Version 1.0. Real Jardín Botánico – CSIC, Madrid. 24 pp.

Pando, F. and Anonymous 2003. Herbar: una aplicación en MS-Access para la gestión de herbarios. http://www.rjb.csic.es/herbario/herbar.htm (19 September 2003).

Pullan, M.R., M.F. Watson, J.B. Kennedy, C. Raguenaud and R. Hyam. 2000. The Prometheus Taxonomic Model: a practical approach to representing multiple classifications. *Taxon* 49: 55-75.

Raguenaud, C. 2002. Managing complex taxonomic data in an object-oriented database. [Unpublished manuscript representing partial PhD thesis, Napier University.] http://www.soc.napier.ac.uk/publication/op/getpublication/publicationid/1845313.

Saarenmaa, H., S. Leppäjärvi, J. Perttunen and J. Saarikko. 1995. Object-oriented taxonomic biodiversity databases on the World Wide Web, Internet Applications and Electronic Information Resources in Forestry and Environmental Sciences, Workshop at the European Forest Institute, EFI Proceedings 3, Joensuu, Finland, pp 121-128.

Shattuck, S.O. and N.J. Fitzsimmons. 2000. BioLink, the biodiversity information management system. CSIRO Publishing, Collingwood, Australia. (http://www.biolink.csiro.au/)

Taswell, S. and R. Peet. 2000. FGDC Biological Data Working Group Biological Nomenclature/Taxonomy Meeting Summary. Smithsonian National Museum of Natural History, 2-3 November 2000. http://biology.usgs.gov/fgdc.bio/FGDC_TaxNom.doc (19 September 2003).

Trehane, P., C.D. Brickell, B.R. Baum, W.L.A. Hetterscheid, A.C. Leslie, J. McNeill, S.A. Spongberg And F. Vrugtman. 1996. International Code of Nomenclature for cultivated plants. *Regnum Vegetabile Ser.* Vol. 133. Balogh Scientific Books, Champaign, IL. 175 pp.

van Regenmortel, M.C. Fauquet, D. Bishop, E. Carsten, M. Estes, S. Lemon, J. Maniloff, M. Mayo, D. McGeoch, C. Pringle and R. Wickner. 2000. Virus taxonomy: classification and nomenclature of viruses: seventh report of the International Committee on Taxonomy of Viruses, 1st edition. Academic Press, San Diego, London. 1162 pp.

Vernon Systems. 2003. Vernon: User Manual, Version 4.0. Vernon Systems Ltd., Aukland. 378 pp. http://www.vernonsystems.com/downloads/VernonManual.pdf (24 March 2003).

Winston, J.E. 1999. Describing species: practical taxonomic procedure for biologists. Columbia University Press, New York. 518 pp.

Ytow, N., D.R. Morse and D.M. Roberts. 2001. Nomencurator: a nomenclatural history model to handle multiple taxonomic views. *Biol. J. Linnean Soc.* 73(1): 81-98.

Ytow, N., D.M. Roberts and D.R. Morse. 2002. A data structure shared by databases supporting multiple taxonomic views developed for different taxa: a comparative analysis. http://www.nomencurator.org/TDWG.html (19 September 2003).

Zhong, Y., S. Jung, S. Pramanik and J.H. Beaman. 1996. Data model and comparison and query methods for interacting classifications in a taxonomic database. *Taxon* 45: 223-241.

Zhong, Y., Y. Luo, S. Pramanik and J.H. Beaman. 1999. HICLAS: a taxonomic database system for displaying and comparing biological classification and phylogenetic trees. *Bioinformatics* 15(2): 149-156.