The MVZ Collections Information Model

Introduction

The collections information model represents the third element in the analysis and design phase of the MVZ's Collections Information System Re-Engineering Project. It was preceded by a high-level functional requirements analysis and a system requirements analysis, which are available at: http://www.mip.berkeley.edu/mvz/{funcreq.pdf | sysreq.pdf}.

Objectives and Scope of the Model

The objectives of the information modeling phase have been to develop a view of collections information that is both comprehensive and detailed. The scope of collections information was determined, through the requirements analyses, to encompass:

- Information currently captured in TAXIR;
- Additional information present on catalog cards, specimen labels and tags;
- Information in the catalogs of ancillary collections, most of which are not yet automated (e.g., field notes, sound tapes, photographs, films, etc.);
- Information about collections management activity (e.g., accessions, loans, permits, curation activity); and
- References to information or collection resources that will remain outside the system, such as references to past accession records, correspondence, etc.

Methodology

The information above is described with a modeling methodology that facilitates thorough data analysis. This level of detail is required to ensure that the resulting logical data structures do, in fact, correctly represent the Museum's requirements, and that they are appropriate for implementation in a relational database. The benefits of relational technology (powerful query capability, the enforcement of data integrity, stability of the underlying design in the face of new requirements, etc.) cannot be realized unless the logical design conforms to the basic constructs of the relational model.

The methodology we have used, Object Role Modeling (ORM), is for building conceptual models, which are less hampered by implementation issues than the logical models produced by Entity-Relationship (ER) techniques. A fully described conceptual model does, however, imply a logical model and can be used to design a database. In particular, ORM is better suited to developing a fully normalized logical model because the methodology requires the modeler to examine the relationships (i.e., functional dependencies) among data elements. (Functional dependency is the basis of data normalization; Kent, 1983; Date, 1990). Because the ORM methodology is not as widely understood as the ER methodology, a primer explaining ORM diagrams is appended to the model. A more detailed, but still brief, exposition of the methodology can be obtained from http://www.asymetrix.com/cstools/orm/ormview.htm. A full exposition of the methodology is presented in Halpin, 1995.

Information modelers commonly face a dilemma: whether to maximize the representation of business rules in a highly normalized design, or whether to anticipate database implementation by making "concessions" that improve performance of the most common tasks, but which also blur the representation of business rules. In general, we have chosen the normalized design that expresses business rules in data structures. In a few cases, however, we have made concessions to implementation, and discuss the rules that should be captured in application or server code to enforce data consistency.

Organization of the Model

The model is presented in both conceptual and logical forms. The conceptual model is presented first, as a series of 26 ORM diagrams. Each diagram is accompanied by a discussion of the illustrated concepts and, if relevant, a discussion of the alternative data structures that were considered. The 26 diagrams are grouped into topics as shown in the table of contents for the conceptual model.

The logical model is then presented in the form of table definitions. Table definitions include:

- a table description;
- designation of the primary key;
- a list of data fields with descriptions;
- secondary indices;
- incident foreign keys (i.e., references to this table found in other tables); and
- rules describing valid combinations of data values that apply to two or more fields in the same table.

Table and field descriptions are kept to a minimum to conserve space.

Groupings of tables in the logical model conform to the implementation priorities set forth in the Requirements Analysis. Tables for the main catalogs are presented first, followed by tables for the transaction management application, and tables for the ancillary collection catalogs. The main catalog tables are further grouped by subject categories: collection objects; taxonomy and publications; collecting event and locality; and people, organizations, and addresses. An index to the tables is included at the end of the logical model.

Diagrams treating the last subject category, people, organizations, and addresses, were included in the transaction management section of the conceptual model because these concepts have a stronger overall relevance to transactions. The tables derived from these concepts, however, are needed to develop the primary catalogs, and are therefore included in that section of logical model.

Conceptual Information Model

Conceptual Model Table of Contents

Collections and Collection Objects

Collection Object Subtype Hierarchy

The Collection_Object subtype hierarchy represents the core of this model. The explanation and discussion of this diagram is, therefore, more detailed than for any other conceptual area.

The departure point for our design was the "COLLECTION_UNIT" subtype hierarchy described in the ASC Information Model for Biological Collections (ASC, 1993). That hierarchy was created to accommodate the diversity of object types that comprise biological collections, including unsorted lots, single-taxon lots, (whole) specimens, specimen components (parts), and derived objects. The hierarchy efficiently accommodates the diversity of object types, without a proliferation of tables, data elements, and code, to handle the relationships between the various object types and other concepts in the model, most importantly taxonomic names, collecting events, and collection transactions. The ASC model is only a skeleton, however, and does not provide the complete range of data fields needed to describe real collections. More importantly, it does not resolve the fundamental conflict between catalog numbers, the traditional mechanism for identifying collection items, and the need for a more flexible mechanism, which would enable a wider variety of object types to be identified, including objects below the level of cataloged items. A conceptual extension was needed, therefore, to apply the subtyping strategy to real collections.

In addition to resolving this conflict, the MVZ and ASC models differ in addressing somewhat different sets of Collection_Object types. For example, the MVZ collections contain only one example of an UNSORTED_LOT (or Mixed_Species_Lot), a microscopic slide showing the tongue morphologies of two bat species, and relatively few examples of LOTs. On the other hand, the MVZ does contain a number of significant "ancillary collections", including tissues, photographs, and vocalization recordings, which need to be described in automated catalogs that are integrated with the main collection catalogs. The MVZ model therefore focuses on individual specimens, specimen parts, and related items.

The description of data concepts on this diagram begins with discussions of Collection_Object identity, the constraints and requirements that have influenced our strategy for handling object identity, and the straightforward aspects of these data structures. The discussion then continues with three more subtle and complicated aspects of the subtype hierarchy: 1) the relationships that record how some Collection_Objects can be derived from other Collection_Objects; 2) the strategy for dealing with information about unvouchered specimens; and 3) a brief introduction to determining which Collection_Object subtype is most appropriately related to other concepts in the model. Where relevant, we identify decisions that accommodate MVZ specific practices and might be too restrictive for other institutions.

Definition and Object Identity

A Collection_Object is a member of an MVZ collection that must be described and/or tracked separately from other objects. A Collection_Object may be a single physical object, a logical group of physical objects, or even a purely logical description of an event that has no single corresponding physical representation (e.g., a Vocalization_Recording). Generally, a Collection_Object is, or represents evidence of, a terrestrial vertebrate (tetrapod) collected from a particular place and time. Most Collection_Objects, therefore, are related, either directly or indirectly, to a Taxonomic_Name and a Collecting_Event.

The MVZ model resolves the following constraints and requirements concerning object identity and existing rules for assigning catalog numbers.

Collection Object Subtype Hierarchy ORM Diagram

1. *A modeling constraint*. All objects/entities in a super-subtype hierarchy must have the same primary key. In this case, the primary key values for all Collection_Objects, regardless of subtype, must be drawn from a single domain. The efficiency in managing relationships (mentioned above) cannot be achieved without gathering the various kinds of Collection_Objects under a unifying supertype (as was done in the ASC model).

2. *A constraint of convention*. Catalog numbers have already been assigned to nearly 600,000 specimens and are commonly understood to encompass all parts of a *single individual*, which may be separated by intention (see 3). Catalog numbers play an important role in maintaining the integrity of collection materials and information, but cannot be used as the primary key of a Collection_Object subtype hierarchy because the meaning of a catalog number is already defined and is too narrow to accommodate a diverse array of Collection_Object types, particularly Specimen_Parts.

3. *A requirement*: Several kinds of collection items do not bear unique catalog numbers, but must be tracked and described independently. For example, it is possible for the skull of a specimen to be lent, while the skin (or any other part) remains in the collection. For an object to be tracked and described separately from related objects, it must be described by a separate record, and that record must be permanently and consistently identifiable (e.g., by a number).

These requirements are reconciled by making Cataloged_Items a subset (subtype) of Collection_Objects. The Collection_Object supertype creates the data fields that are common to all Collection_Objects, whether they are Cataloged_Items or just associated objects, including Collection_Object_id, a unique, permanent, system-assigned, surrogate key (identifying number), which is typically not visible to users and identifies all Collection_Objects regardless of type. In addition to establishing object identity, the Collection_Object_id functions primarily to establish relationships between a Collection_Object and other concepts within the system. Catalog_Numbers will continue to function as the dominant access key to collection records, but because a Catalog_Number may not designate a unique object, users will be able to supplement Catalog_Number with object types, Part_Names, and in some cases “sub-numbers”, to designate a particular Collection_Object.

Other data fields created for every Collection_Object include the object’s current disposition, when the object-record was entered and last edited, by whom, and any other identifying numbers that may have been assigned to the object. Other identifying numbers are not generated by an MVZ-controlled process, and therefore an alpha-numeric field is used for these “numbers” to accommodate the widest variety of numbers or codes. An alpha-numeric field, however, does not allow these “numbers” to be sorted in numeric order (e.g., the number “1234” would sort before the number “20”).

Every Collection_Object must also have a Collection_Object_Type that indicates the subtype of the Collection_Object. The Collection_Object_Type data field is mandatory (must be entered for a Collection_Object instance to be valid), and is the only field for this purpose, indicating that Collection_Objects subtypes are mutually exclusive. For each Collection_Object instance (record), a corresponding instance must exist in one and only one of the *direct* subtypes of Collection_Object (i.e., Specimen_Part, Karyological_Slide, Histological_Slide_Series, Tissue_Sample, or Cataloged_Item).

Data fields provided by the Cataloged_Item subtype enable a subset of Collection_Objects to be designated as the “owners” of traditional Catalog_Numbers. Note that a combination of Institution_Acronym, Collection_cde, and Catalog_Number forms the complete specification of a traditional “catalog number”, thus a single Cataloged_Item might be designated as “MVZ-M-1234”. A unique index spans these data objects to guarantee that catalog numbers are assigned to one and only one Cataloged_Item within each collection.

The data field, Cataloged_Item_Type, indicates the subtype of a particular Cataloged_Item (subtypes here are also mutually exclusive). Note that Collection_cde and Cataloged_Item_Type are not synonymous. For example, a Cataloged_Item that is a Biological_Individual might be cataloged in any one of several collections, such as the Mammal, Hildebrand, or Teaching collections. Also note that a single Cataloged_Item (e.g., Biological_Individual) cannot be (currently) cataloged in more than one collection; e.g., the skin and skull of a mammal to be cataloged in the main collection, while a histological preparation is cataloged in the Hildebrand collection. Allowing a single item to be cataloged in more than one collection would require an equivalent number of copies of the same data (see discussion of object derivation relationships below).

Data elements associated with the subtypes of Cataloged_Items are described in subsequent diagrams.

The four Collection_Object subtypes that are not Cataloged_Items (Specimen_Part, Tissue_Sample, Histological_Slide_Series, and Karyological_Slide) will not bear unique catalog numbers. Instead they are (will be) labeled with the Collection_cde and Catalog_Number of the Biological_Individual from which they were derived. Catalog "sub-numbers" will be used to identify objects from the same individual that are otherwise indistinguishable (e.g., two tissue samples taken from the same individual might be labeled as H-1234.1 and H-1234.2).

Object Derivation Relationships

It is important to understand that the Collection_Object subtype hierarchy and the object derivation hierarchy (or parts decomposition) are not the same things. The subtype hierarchy, shown on the Collection Object ORM diagram, provides for the inheritance of data *structures*; e.g., enables a Biological_Individual to have a Catalog_Number, and a Tissue_Sample to be individually trackable by its own Collection_Object_id. The derivation hierarchy, on the other hand, records the information about which objects, parts, or preparations have been derived from a given individual. The derivation hierarchy is represented by the relationship "Collection_Object is derived from Biological_Individual", or more precisely, by the **values** held in the data fields (i.e., foreign keys) created by those relationships. A value would indicate, for example, that *this* Tissue_Sample was taken from *that* Biological_Individual.

Figure 1 shows the "real world" derivation relationships that are of interest to the MVZ. The cardinalities (how many derived objects) and existence dependencies of these relationships are discussed below.

> Tissue Sample – One or more Tissue_Samples must be derived from a Biological_Individual, but the individual might not be cataloged in an MVZ collection. (See discussion of unvouchered specimens, below.)
>
> Tissue Preparation – One or more Tissue_Preparations (e.g., an extract) may be derived from a Tissue_Sample or directly from a Biological_Individual. Again, a "voucher" might not be present for the individual. If the preparation is derived from a catalogued Tissue_Sample (vs. a Biological_Individual) it may be important to record from which Tissue_Sample the extract was derived.
>
> Specimen Part – A Specimen_Part must be derived from a Biological_Individual. The only exception is that in some cases a Specimen_Part can be derived from another part. We believe, however, that the derivation of parts is obvious and obviates the need to record explicit part-from-part derivations.
>
> Karyological Slide – A Karyological_Slide must be derived from a Biological_Individual. All karyological slides have been produced in conjunction with J. L. Patton's research program and are accompanied by "voucher" material (standard skin, skull, etc. preparations), except in cases where voucher material has been deposited in another museum.

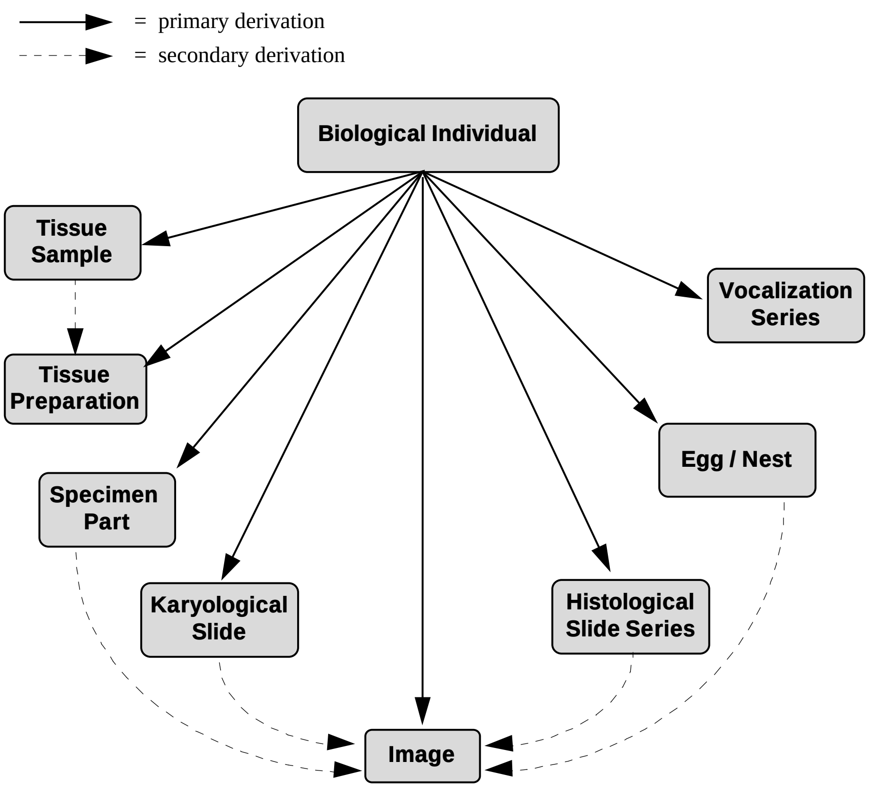

Figure 1. The object derivation hierarchy.

Image – An Image may be derived from virtually any other kind of collection object, or from none (e.g., a habitat photograph; see Images diagram). The subject of an Image may include the following kinds of collection objects: one or more Biological_Individuals (whole), a Specimen_Part, a Karyological_Slide, a Histological_Slide, or an Egg-Nest set. Zero to many Images can be produced from a single Collection_Object. An Image can also be derived from another Image (not shown, but discussed further under Images).

Egg_Nest – Material in an Egg_Nest set must be produced by at least two Biological_Individuals (birds). In some species, one or more individuals (helpers) in addition to the biological parents may participate in rearing nestlings. It is also possible that a nest parasite, such as a female cowbird, can contribute to the contents of an egg-set. Most of the 14,000 Egg_Nest collection were not collected along with the parents, nestlings, helpers, or nest parasites. In a few cases, the nest or egg-set is associated with one or both parents and/or nestlings in the collection. No cases exist in the collection in which a single bird is associated with more than one nest or egg-set. In an earlier version of the model the relationship between birds and egg-nests was described as many-to-many. It is now described in a simpler, but more restrictive, “many birds to one egg/nest” relationship.

Vocalization Series – In practice, a Vocalization_Series is always derived (recorded) from a single individual. It is also possible that several series can be recorded from a single individual. Again, the bird from which the recording was made is not always collected.

Derivation relationships shown on subsequent ORM diagrams are summarized in the figure below. Note that Tissue_Prep inherits a relationship to Biological_Individual by virtue of *being* a subtype of Tissue_Sample. Its relationship to Tissue_Sample is in addition to, not in lieu of, its relationship to Biological_Individual.

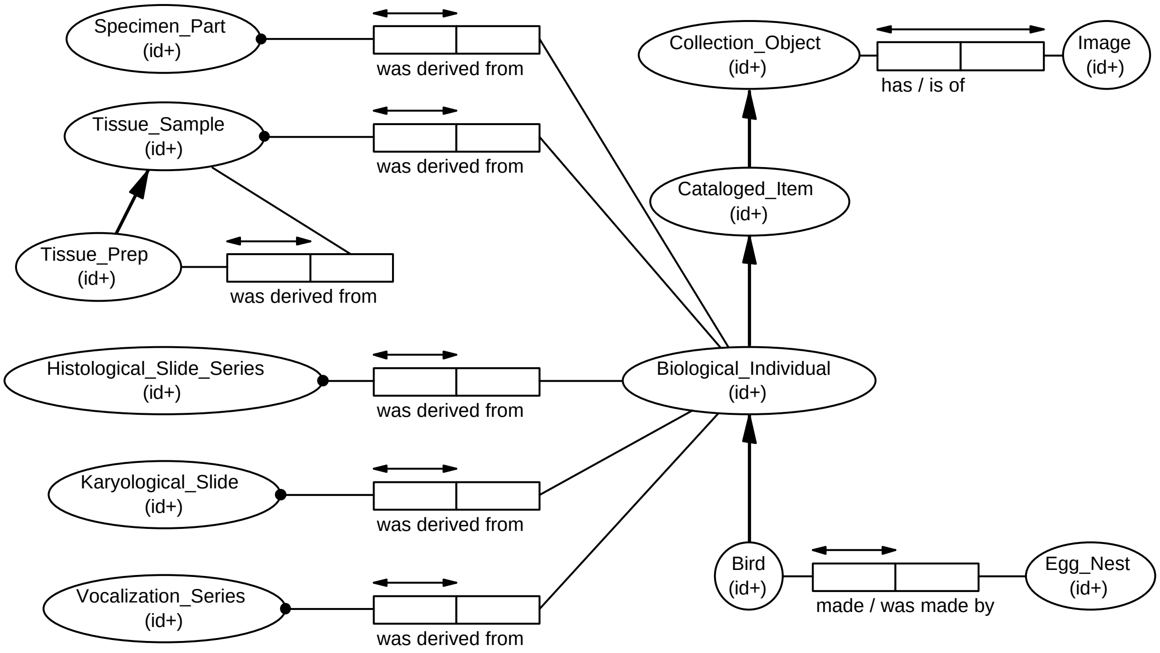

Figure 2

Objects on the left of the figure have equivalent relationships to the Biological_Individual object. We expect that one of the more common views of collections data will treat Biological_Individual as the "root" object, and will show all of the parts, tissues, etc. that have been derived from that individual. If the relationships on the left are aggregated into the Collection_Object supertype (as shown by the "is derived from relationship in this diagram), a simple one-table query can return the identifying information of all Collection_Objects derived from one or more Biological_Individuals. This would place the foreign key identifying the "derived from Biological_Individual" in the Collection_Object supertype. The problem with this data structure is that it accommodates non-sensical "derived from" relationships, such as a Bird being derived from a Herp (a type of Biological_Individual). Keeping each of these "derived from" relationships distinct (i.e., keeping the foreign key in each of the appropriate subtypes), ensures that only appropriate Collection_Object types (Specimen_Part, Tissue_Sample, Karyological_Slide, Histological_Slide_Series, and Vocalization_Series) can and must be recorded as derived from a Biological_Individual. The solution we have adopted is to use both data structures for their respective strengths. Foreign keys placed in the subtypes constrain the data to reflect business rules accurately, and a copy of the foreign key is replicated in the supertype to facilitate data retrieval. This approach requires the use of application and/or server code (triggers and stored procedures) to manage the redundant copy of the key value in the supertype. (The double asterisks on the "Collection_Object *is derived from* Biological_Individual" relationship indicates that this is a derived or denormalized fact.)

Unvouchered Individuals

Tradition in the MVZ (and many other institutions) holds that an individual can be cataloged only if it is represented by one or more "standard" preparations or parts, such as a study skin, a whole animal in alcohol, or at least some portion of a skeleton. Since this tradition was established, however, new study methods have produced a variety of non-standard preparations or collection objects. A problem arises because items such as blood samples, stomach contents, and vocalization recordings can be collected without collecting (or keeping) the entire individual, which causes non-standard collection objects to be present in an MVZ collection without the "standard" parts (i.e., the voucher) being present in one of the main collections. In addition, the sharing of materials among museums and researchers has resulted in non-standard objects being deposited in one institution, while the standard preparations are deposited in another.

The following examples illustrate practices that might result in Collection_Objects without standard “voucher” material:

Specimen_Part: Generally, one or more standard (anatomical) parts is sufficient to warrant assigning a catalog number. The exception among Specimen_Parts is stomach contents removed from an individual that was captured but not collected.

Tissue_Sample: A blood sample, toe clip or ear punch can be taken from an individual that is then released; a Tissue_Sample can be taken from individual that is then deposited in another institution; new tissue can be received in exchange for MVZ tissue or extract.

Histological_Slide_Series: A Histological_Slide_Series can consist of an entire individual, or part of an individual currently residing in another institution. In both cases, MVZ does not hold a traditional voucher.

Karyological_Slide: A Karyological_Slide can exist for an individual deposited in another institution. This is increasingly common for specimens collected in Latin America.

Vocalization_Series: A Vocalization_Series can be recorded from an individual that was not collected or is deposited elsewhere.

Egg_Nest: An egg set or nest can be collected without collecting the parent individuals (this applies to approximately 99% of Egg_Nest collection).

Image: A photograph or moving image can be taken of an uncollected individual.

The traditional rules for cataloging specimens are too restrictive to accommodate new kinds of materials in a consistent manner. The data structures that describe an individual, such as sex or age-class, are provided by the subtype Biological_Individual. The fact that Biological_Individual is a subtype of Cataloged_Item means that an individual must be cataloged (assigned a collection code and catalog number) in order to be described. The figures below illustrate the problem using a simplified entity-relationship-style diagram in which entities are represented by rounded rectangles and example data elements by ellipses. Consider the situation, in which a field worker makes a number of Vocalization_Recordings from a bird (Biological_Individual), collects and prepares the bird as one or more standard parts, and the bird is then cataloged according to the data structures shown in Figure 3. Note that Sex and Age_Class are properties of the Biological_Individual entity (i.e., the values are functionally determined by the particular Biological_Individual). Traditional cataloging rules do not pose a problem because the individual is collected along with the recordings.

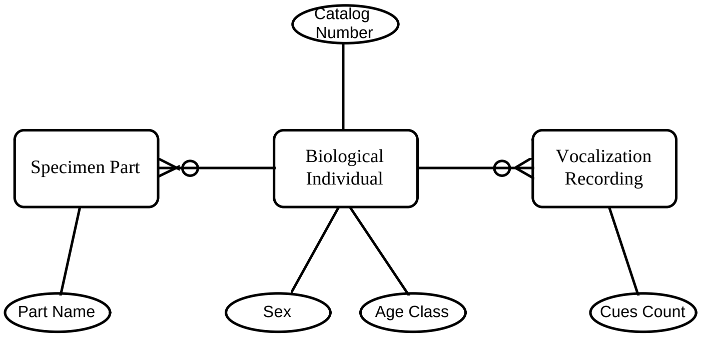

Figure 3

If the bird is not collected along with the recordings, however, the traditional cataloging rules “prohibit” creating a catalog record for the bird (Biological_Individual). In the absence of a “bird” record, the Vocalization_Recordings could be cataloged, but the Sex, Age_Class attributes could not be recorded, as shown in Figure 4.

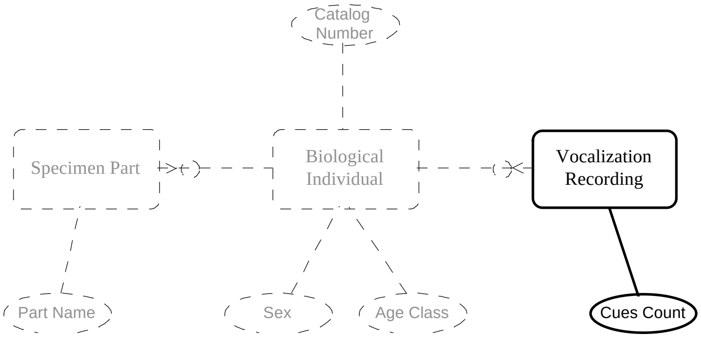

Figure 4

If the data fields for holding information about the individual were moved to, or replicated in, the entity describing the Vocalization_Recording, as shown in Figure 5, those same data would have to be duplicated if more than one Vocalization_Recording had been made. This alternative model would not be normalized, and additional application or server code would be required to guarantee consistency across multiple Vocalization_Recordings of the same individual. Note also that a particular Vocalization_Recording would have to be designated as the primary copy of data about the bird. Having done that, the application would not be robust to other combinations of Collection_Objects, such as a blood sample and a photograph.

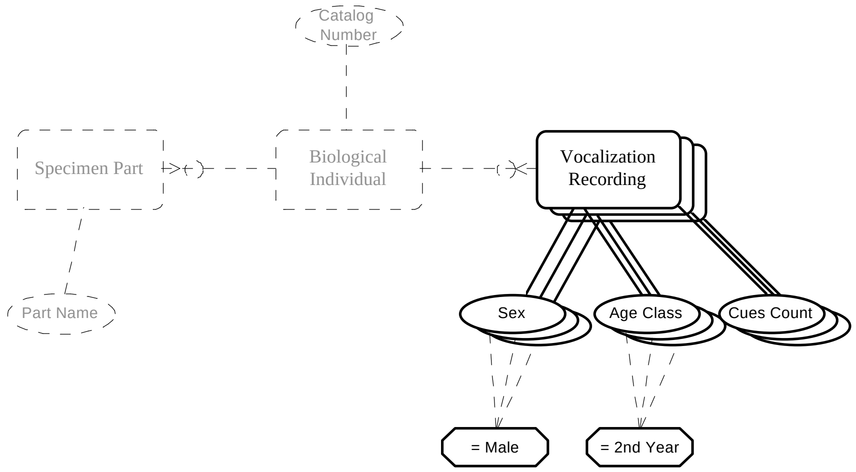

Figure 5

Our solution is to recognize the Biological_Individual data object as a logical concept rather than a physical one. The criterion for assigning a Catalog_Number to a Biological_Individual is expanded, and becomes simply the need to describe an individual, whether that need arises from the existence of standard voucher preparations, or ancillary collection items. As shown in Figure 6, the lack of standard Specimen_Parts does not preclude an "unvouchered" individual from being cataloged. (Also note that the unvouchered individual is not required to be cataloged in the same catalog as a normal vouchered individual.) The advantage of this solution is that it guarantees data consistency by making it impossible to have two versions of the same fact, such as the sex of a particular individual. From another perspective, the design requires the same fact to be stored in the same place -- using the same data structure -- regardless of the physical evidence that may or may not be present in the various collections.

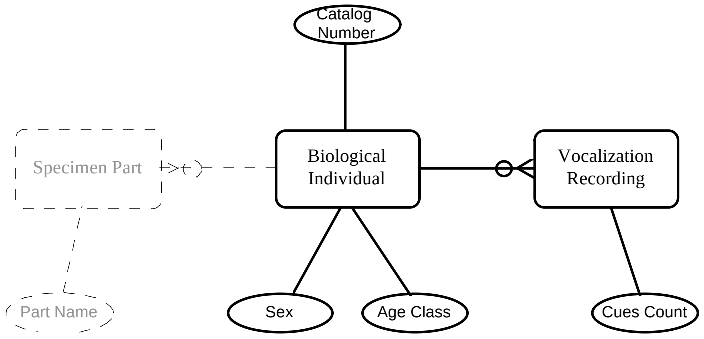

Figure 6

Object transformation

A Collection_Object can be transformed by analysis or use. Transformation may occur while on loan or while in the Museum. Derivation of new objects doesn't necessarily come at the expense (destruction) of the existing object:

simple derivation: A → A + B
or
destructive derivation: A → B

For example, a Tissue_Sample can be transformed into an extract (and a slightly smaller tissue sample), or a mandible can be disassembled into bone fragments (discarded) and a series of teeth mounted on SEM stubs.

A current disposition is recorded for each Collection_Object. When an object is completely consumed or destroyed in a usage event, it’s disposition will be marked as deaccessioned. When an object is lent (see Transaction Management section) this disposition will be set to “on loan”. The Loan_Item_Disposition field (see Loan_Item) indicates its status while on loan.

An object created through derivation from another object does not involve the transfer of title (see transaction management section) and therefore does not require a new accession transaction. For example, if an MVZ Tissue_Sample is lent, and the borrower returns an extract of it, the returned extract does not require a new accession number.

Whether or not a newly derived or transformed Collection_Object needs a new Collection_Object record (different than any existing records) is determined by whether or not the original object still exists and whether or not the new object type must be described with different data fields. For example, if a herp specimen is cleared and stained, a new Collection_Object may not be required because field values in the relevant Specimen_Part can simply be updated. An extract (Tissue_Preparation), however, requires fields not present for a Tissue_Sample, and thus requires a new Collection_Object record, regardless of whether or not the source Tissue_Sample remains.

Biological Individual

We have chosen to use the term Biological_Individual rather than Specimen because the latter is variously understood to mean a whole individual, or one or more standard preparations derived from it. In this model the Biological_Individual object represents a logical concept rather than the physical individual. This strategy is appropriate for the mammal and bird collections because physical individuals are typically prepared into a variety of Specimen_Parts. While this strategy doesn't necessarily seem appropriate for herpetology collections, in which individuals are typically preserved and stored whole, it can be used for individuals that are treated this way, and also accommodates the less common cases in which more elaborate preparations are made. Moreover, the amount of data that a user has to enter under a system that accommodates multiple parts per individual should be the nearly the same as in a system that treats an individual as a physical item. In both cases data are entered to describe the individual's Taxonomic Identification, Sex, Age_Class, etc., as well as the physical representation of the individual. This design differs in being able to accommodate multiple physical representations as separately trackable objects, while ensuring that data about the individual are consistent.

The Biological Individual diagram shows the simple data objects that are common to all Biological_Individuals, as well as those that are “discipline specific”; i.e., used to describe individuals in one of the three MVZ disciplines (as shown by the subtypes Mammal, Bird, and Herp). The data object, Biological_Individual_Type, is used as the classification variable, and indicates the kind (subtype) of individual: M (mammal), B (bird), H (herp), and F (fish). The Teaching and Hildebrand collections contain whole or preparations of fishes, but because catalog descriptions of these individuals don’t require any additional data elements, the “F” subtype isn’t shown as an explicit subtype in the ORM diagram. Note also that Biological_Individual_Type and Collection_cde are not redundant. For example, an individual may be of the Biological_Individual_Type “M”, but may be cataloged in the Teaching Collection (Collection_cde, “T”).

The “ate / was eaten by” relationship is included for those rare cases in which an important specimen is found to be the “stomach contents” of another (e.g., a rare kangaroo rat found in the stomach of a rattlesnake).

The “is parent of / is child of” relationship is used for cases in which one (or both) parent(s) is (are) collected with its (their) progeny. A frequency constraint (0-2) is placed on the “is child of” role because an individual may be present in the child role no more than twice; i.e., can be related to at most two parents.

The values of Age_Class are to some degree discipline-specific, and Age_Determination_method is recorded only for Birds. Therefore, it may be more appropriate to create separate Age_Class fields for each of the different types of Biological_Individuals..

Biological Individual ORM Diagram

Specimen Part

In Mammalogy and Ornithology in particular, Biological_Individuals are routinely separated into a suite of standard preparations or parts. The Specimen_Part object is used to describe the physical anatomical structures/preparations that represent the Biological_Individual. A controlled vocabulary is used to describe the nature of each part. A combination of Part_Name and Part_Modifier is unique for any given individual. This is shown by the interpredicate uniqueness constraint in the diagram. While an individual has many paired structures, the modifiers “left” and “right” can be used to make the description of almost any single part unique (e.g., humerus, left). In the logical model, the interpredicate uniqueness constraint is implemented as a unique index over the fields, Derived_from_Biol_Indiv, Part_Name, and Part_Modifier.

A whole individual is considered to be a kind of Specimen_Part. This means that even in the herpetology collection, in which nearly every specimen is represented by a whole animal in alcohol, a Specimen_Part (i.e., whole animal) must be entered for the Biological_Individual.

The relationship “Specimen_Part*is derived from*Biological_Individual” shown on this diagram represents the “master copy”, which is replicated (denormalized to facilitate data retrieval) in the Collection_Object supertype (see the discussion of Derivation Relationships under the Collection_Object ORM diagram).

Specimen Part ORM Diagram

Tissue Sample

Each Tissue_Sample is described by a controlled vocabulary of tissue types. More than one kind of tissue may be stored in the same vial, thus a list of tissue types can be attached to a tissue sample.

A Tissue_Sample may be further categorized (subtyped) as being a Tissue_Preparation. Because there is only one subtype, a simple logical field indicates whether a particular sample is a Tissue_Preparation or not. A Tissue_Preparation can be made without a tissue sample being cataloged in the system. Hence, the “was derived from Tissue_Sample” relationship is not mandatory for Tissue_Preparations. Both Tissue_Samples and Tissue_Preparations, however, must be related to the Biological_Individual from which they were taken. This relationship (the foreign key value identifying the Biological_Individual) is replicated to the Collection_Object supertype and is shown the Collection_Object diagram.

If a Tissue_Preparation is made from a cataloged Tissue_Sample, the tissue type(s) listed for the preparation must be one of those listed for the source sample. If preparation is made from an uncataloged sample, all tissue types should be available for its description.

Tissue Sample ORM Diagram

Slide Preparations

Two types of slide preparations are generated in sufficient numbers, through the research programs of MVZ curators, to warrant the establishment of new collection catalogs. These are the karyological slides (chromosome preparations) and histological slides (embedded and sectioned material). Karyological slides will be visibly identified by the catalog number of the individual from which they were prepared followed by a sub-number (Karyo_Slide_number). The Histological slides will be group-catalogued because the requirements do not call for tracking or describing individual slides within a series.

Slide Preparations ORM Diagram

Egg_Nest

The egg and nest collection represents, for the most part, a series of privately assembled collections that were acquired by the Museum. Original data for these items is consequently somewhat variable. Most items are composed of a clutch of eggs, but some have an accompanying nest, and some nests have no eggs. The tradition in oology is to catalog eggs by clutch, rather than individually.

The most difficult data structure problem posed by the egg and nest collection is the association of a nest or egg-set with a Taxon_Name.

- A nest gets its taxonomic identification from its constructors/users; i.e., if a nest is built / used by a member of taxon-x, then the nest “is of” taxon-x. The true relationship is: nest is built or used by individuals of taxon-x.
- Identification of a nest could be complicated by hybrid pairs. In such a case, the nest would be, in some sense, a hybrid, even though neither of the parents is a hybrid.
- An egg set contains one to several eggs, and they may be of different species (i.e., one or more eggs may be nest parasites). The collection actually contains several parasitized egg-sets. The parasitizing species can be unknown, suspected, or known. (No cases are present in which a nest is parasitized by more than one species.)

These facts make taxonomic identification of a nest or egg set a more complex series of statements than for a typical Biological_Individual or Cataloged_Item. Within the framework of the ASC model, an Egg_Nest set could potentially be an UNSORTED_LOT (mixed species lot), but differs from that concept in that a maximum of only two taxa can be present in an egg-set, host and parasite. We have provisionally accommodated the requirement for two current taxonomic identifications by adding another relationship between the Egg_Nest object and the Taxon_Name object. This solution, however, is not entirely satisfactory because data structures used to hold the parasite identification are different and simpler than those that are used to hold all other identifications (see Taxonomic_Identification ORM diagram). Deficiencies include that a different kind of query must be used to retrieve Egg_Nest records by parasitic species, the date and person making the identification cannot be recorded, and an identification history cannot be accumulated for parasitic eggs.

An alternative strategy would be to move the identification relationship up the Collection_Object hierarchy, from Cataloged_Item to the root supertype, and to partition (decompose) Egg_Nest sets into their specific components, such as a Nest and zero to two Egg_Sets (nest only, “host” eggs only, or host and parasite eggs), each of which would be represented by a distinct Collection_Object (somewhat like Biological_Individuals are decomposed into constituent Specimen_Parts). Separate identification records could then be created for each component of the Egg_Nest set. A potential drawback of this strategy is that the client application, rather than the structure of the database, would have to ensure that all other kinds of Collection_Objects “receive their identifications” through the Cataloged_Item (Biological_Individual) from which they were derived.

Recorded descriptions of egg sets sometimes indicate that not all eggs present in the nest at collection time are now present in the set. Hence, fields are provided for egg counts in the field and egg counts present in the set.

Egg_Nest ORM Diagram

Vocalizations & Audio Tapes

The MVZ holds a moderately large collection of animal vocalizations recorded on audio tape. A recording may originate through either field or laboratory work. Because many copies can be created from a single recording, two separate catalogs are used to manage this collection: one is for logical representations of the vocalization series themselves, i.e., the occurrence of an event (vocalization); and the other for the tapes on which these recordings reside.

A Vocalization_Series is a continuous recording of some duration. Many songs are typically recorded in a single Vocalization_Series, with an audible “cue” (verbal count) typically given by the field-worker after each song. A Vocalization_Series is similar to other kinds of Collection_Objects because each is “collected” in a single Collecting_Event (place-time), and each represents evidence of a single Biological_Individual, which belongs to a taxon. Catalog number, Collectors_Number, Collecting_Event, and Taxonomic_Identification data are all handled with data structures present in, or related to, the supertypes (see Collection_Object Diagram). A list of contained Vocalization_Types and total number of songs (Cues_count) are attributes of each series.

Once recorded on a Master_Tape, a Vocalization_Series can then be transferred to one or more tapes. Typically, they are “sorted” by copying “related” series onto Species_Tapes. The existence and position of a Vocalization_Series on a Tape thus becomes a property of a Vocalization_Recording.

One or more Spectrograms and/or digitizations (Digital File) can be generated from a particular Vocalization_Recording.

A pseudo-foreign key to Master Tape has been added to the Bird object (see Biological_Individual Diagram). This is a pseudo-foreign key because it will hold the catalog numbers of the Master tapes as they appear in TAXIR records, before the Vocalization and Tape catalogs have been automated.

Vocalizations and Audio Tapes ORM Diagram

Images

The MVZ holds collections of several kinds of visual resources, including photographs, slide transparencies (of various formats), technical illustrations, fine art illustrations, films and videos. As with recordings of vocalizations, all visual resources can exist in multiple copies. Simple reproduction results in copies that have the same logical content or subject matter as the original, although physical medium can change from one reproduction to the next. Describing image content is perhaps the most important and time consuming aspect of cataloging visual materials. It is therefore desirable to make the same descriptions of logical contents “available” to the physical objects that give them substance.

Our treatment of images thus divides naturally into two sections: aspects of subject matter or content, which are properties of “Image_Content”; and aspects of physical representation and reproduction, which are properties of the physical “Image”.

We don’t believe that the potential for reproduction processes to alter image content detracts significantly from the utility of this basic dichotomy. Determining that reproduction has altered image content sufficiently to warrant distinguishing subsequent images from the original will always be a judgment call. The model accommodates these “sufficiently different reproductions” by allowing the user to create a separate Image_Content description for the new physical Image, while still recording that the modified Image was derived from the original Image. In other words, members of a physical reproduction lineage do not all have to represent the same logical content.

The treatment of moving images (i.e., Films and Film_Clips) is analogous to the treatment of Vocalization_Series and Tapes.

Images ORM Diagram

Taxonomic Information, Citations, and References

Taxonomic Authority File & Classification

The primary uses of taxonomic information in collections are to label collection objects with taxonomic names, and then to retrieve information by those names, either directly or indirectly through a classification. The fact that classification, taxonomic nomenclature, and identifications are all dynamic turns these simple requirements into a difficult data management problem. Our present model of taxonomic information provides only those capabilities that are, in our judgment, essential to the proper management of taxonomic names, classification, and the application of names to collection objects. The model does not recognize taxa separately from the names that designate them (a fact that is not widely appreciated), nor does it accommodate multiple classifications. It does provide: vocabulary control over taxonomic names (through an authority file of names linked to appropriate citations of the relevant literature); a single hierarchical classification with the capability to hold a variable number of "interpolated" ranks (e.g., tribe, subtribe, etc.), a many-to-many mapping of synonyms to valid names; the capability to sort taxa in collection (i.e., "phylogenetic") as well as alphabetic order; and the capability to re-identify collection objects without discarding previous identifications. Generally, the model of taxonomic nomenclature could be characterized as permissive. It can accept data from a wide variety of sources (with heterogeneous structures) and places the responsibility for assuring nomenclatural and classification integrity on the data administrator, rather than on restrictive data structures. For example, the model does not enforce the rule that a valid species must be a member of the genus that contributes the generic portion of its name (e.g., the data administrator must ensure that the species *Thomomys bottae* is a member of the genus *Thomomys*).

A Taxon_Name can be unomial (for taxa above the rank of species), binomial (species), trinomial (subspecies), or a hybrid name (see Hybrids, below). A Taxon_Name is identified by a surrogate key. This enables a data entry error to be corrected without requiring identifications to be changed. It can be argued that either the rank and/or the publication (including author and date) are sufficient to differentiate all cases in which the same string of characters has been used to designate two or more different taxa (taxonomic concepts). We have elected not to enforce this rule with an index because the original publication will not be recorded initially for many names. The data administrator may find it beneficial to create this index once the work of compiling community-wide taxonomic authority files (including publication data) has progressed sufficiently.

The model allows the Full_Taxon_Name to be parsed into Taxon_Name_Elements, and allows full names to be indexed by the elements they contain. (The index anticipates a potentially needed performance enhancement and should not be understood to implement a taxonomic "business rule".)

The model allows the original author(s), year, and publication to be recorded for every name. Again, none of these attributes is required because the MVZ's taxonomic authority file will be derived from several sources (e.g., Mammal Species of the World, by Wilson and Reeder), with different structures and degrees of completeness. Note, the author(s) of a taxonomic name may not be the same as the author(s) of the publication.

A subtype series is used to partition all taxonomic names into two nested subsets: those that are the correct name of a valid (i.e., accepted) taxon; and those that are arranged "phylogenetically" in the collections rather than alphabetically.

Taxonomic Name Authority File and Classification ORM Diagram

A Taxon_Name may be entered into the database and designated as private before it has been published. A Taxon_Name that has been designated as private should not be visible to the "public" (i.e., users not affiliated with the MVZ), and cannot also be designated as valid. (A name should be considered valid only after it has been published.)

A Taxon_Name that is not valid must be a synonym of a Valid_Taxon_Name. The exclusion constraint between the roles Taxon_Name "is valid" and Taxon_Name "is synonym of" prevents a Valid_Taxon_Name from being a synonym of a Valid_Taxon_Name. A Valid_Taxon_Name may have a rank (but is not required to) and may be classified in a single hierarchy. (The ring constraint specifies that the recursive relationship is acyclic.) A Taxon_Name that is not valid may be the synonym of more than one Valid_Taxon_Name. This situation can arise when the Taxon_Name is typified by a mixed syntype series, or in cases when the type specimen cannot be unambiguously assigned to a particular valid taxon.

Some taxa have been accorded a special "protection status" by a regulatory body, which then determines which permits are needed for collecting and shipping these taxa.

Taxa, Names, and Ranks

At present, the model does not require that every taxonomic name be associated with a Linnaean rank. Allowing Rank to be null enables the database to hold classifications in which some taxa have not been assigned ranks, as is being done now in some cladistic classifications. Rank is an important attribute, however, in classifications containing genera that have been subdivided into subgenera. In each subdivided genus, there are always two taxa (the genus and subgenus) that have the same name (e.g., genus *Thomomys*, and subgenus *Thomomys*) and are distinguishable only by rank.

Hybrids

In reality, hybrids are not taxa, but individuals that have been identified as the progeny of a mating between parents that are/were members of two different taxa, or even a backcross between a hybrid and a "normal" parent.

Examples:

Chen caerulescens hutchinsi x *Branta canadensis*
Thomomys bottae x *Thomomys umbrinus*
Thomomys bottae bottae x *Thomomys bottae alexanderi*
Thomomys bottae bottae x *Thomomys umbrinus*
[Thomomys bottae x *Thomomys umbrina]* x *Thomomys bottae*

Rather than accommodating hybrid identifications as special identifications, in which two taxa are associated with an individual simultaneously, we have chosen to accommodate hybrids in the taxonomic authority file as if they were real taxa.

Implications for other aspects of the model:

- Author / date for hybrid names: Hybrid names are not "published" and therefore do not have authors and dates of original publication.
- Decomposition of name into name elements: Taxon_Name_Elements at a given rank can appear more than once in a hybrid name. The purpose of this Taxon_Name_Element decomposition is only to allow indexing on each element, and not to enable a reconstruction of the name from its elements. The model requires that each combination of Taxon_Name_id + Taxon_Name_Element + Rank_name be unique. The application code that parses full names into name elements and inserts them into the database must not cause an error by inserting redundant combinations into the database.

- Linking hybrid names into the classification hierarchy: The classification data structure (is_member_of) allows only a single parent taxon to be designated for each taxon name. The model does not allow both parental taxa to be recorded for a hybrid taxon. We have considered creating a special subtype, Hybrid_Name, under Valid_Taxon_Name, that would support two parent relationships, but doing so would require additional special code to navigate the classification tree. Hybrids are not very common, and thus the following "work-around" was deemed sufficient. For hybrid names, a single parent taxon (is_member_of) will be designated, and that taxon will be the first parent in common to both parental taxa.
- Hybrids and synonyms: Hybrids create another kind of "unsupported" classification/nomenclatural integrity rule on taxonomic names. (The first category includes the rule that species and subspecies names must be formed in accordance with the genus or species, in which they're classified.) If an authority file contains hybrids, re-classifying or re-naming species that form hybrids will require that all related hybrids be re-named and re-classified.

Collection Order

The requirement to report taxa in collection-order is met by data fields created on this diagram. The normalized form of a single list is represented by a data structure called a “linked list”. A linked list simply provides a “this taxon is followed by that taxon” type of relationship. No Ordered_Taxon may occur twice in the same role (i.e., a single taxon cannot be followed directly by more than one taxon), and every Ordered_Taxon must occur in at least one role. (Taxa on either end of the list are special cases because they occur in only one role, whereas taxa within the list occur in both roles.)

A linked list, however, is an inefficient structure for some retrieval operations, thus a simple integer value, Collection Order, is attached to each Ordered_Taxon to speed performance. A single rearrangement of collection order, however, requires that the Collection_Order values be changed for half of the list, on average.

The three main collections differ in which rank is arranged in “phylogenetic” order and which are in alphabetical order, and in some situations there is variation within a collection.

Collection Order ORM Diagram

Taxonomic Identification

An Identification is viewed in this model as an event in which a Person applies a Taxon_Name to a Cataloged_Item at a particular time (date). More than one identification can be made on a given Cataloged_Item, and thus an "identification history" can be accumulated for the item. Maintaining an identification history requires a data object that implements a many-to-many relationship between the Taxon_Name and Cataloged_Item objects. Only one Identification at a time, however, "is accepted", as shown in the second relationship between Cataloged_Item and Identification. This relationship indicates which Taxon_Name should appear on most labels and reports (i.e., those not concerned with identification histories). The effect of this structure in the logical model is to create a foreign key in the Cataloged_Item table that identifies the accepted Identification. The business rule "only one accepted identification at a time" is therefore represented in data structure rather than application logic. A necessary additional constraint, however, is not represented explicitly in the model: the foreign key in the Cataloged_Item table (Accepted_Identification_id) must point to an Identification record that in fact concerns that Cataloged_Item. (InfoModeler Version 2 will support interpredicate constraints between sets of roles, rather than just between individual roles. The rule in question here is a subset constraint that requires that the pair of roles in the lower relationship -- a specific pair of Taxonomic_Identification_id and Cataloged_Item_id values -- must first exist as a pair in the upper relationship.)

If application performance needs to be improved, a three table join can be eliminated by denormalizing -- copying -- the accepted Taxonomic_Name across the Identification relationship, into the Cataloged_Item table.

An Identification can be supplemented by Remarks and a coded Identification_Modifier (e.g., "cf.", or "?").

We expect that one of the more important uses of collections information will be to determine the geographic ranges of taxa based on collections records. It is also true, however, that in some cases the finest level of taxonomic identification (e.g., subspecies) can be based on the collecting locality of the specimen in question, rather than on characteristics of the specimen itself. We provide a simple mechanism (a logical flag) that will enable a user to filter out any specimen for which the identification was based on geography.

This strategy is not entirely satisfactory, however, because excluding all identifications based on geography would exclude cases in which subspecific identification was based on geography even though specific identification was based on characters. A preferable alternative might be to replace the unary predicate (logical field) with a field that allows the user to indicate the highest rank that is determined by geography. Another alternative would be to use the logical flag, but also to enter a separate identification, to the higher rank taxon (based on characters) as part of the identification history.

Primary Key of the Identification Object

In the ASC model (1993), a functionally comparable entity (Determination) was specified that had a primary key composed of four data elements, their equivalents in this model being Collection Object, Taxon Name, Person, and Date. The important capacity that the ASC structure enables is to accommodate the recording of corroborating identifications – i.e., it allows the same specimen to be identified as the same taxon, as long as two or more records are distinguished by different identifiers (Person), or different dates. The shortcoming of the ASC Determination entity is a formal one; the relational model stipulates that primary key attributes may not be null. The ASC model, therefore, would not accommodate many real cases in which either or both the Person and Date may not be known (i.e., will be null). The MVZ model uses a surrogate key to distinguish Identification records, and thus accommodates both corroborating identifications and cases where Person and Date information are missing. Only three data elements are required: the surrogate key for the identification (Identification_id), the foreign key to the Taxon_Name (Taxon_Name_id), and the foreign key to the Cataloged_Item (Cataloged_Item_id).

Scope of the Identified Object

All parts or evidence of a single Biological_Individual must have the same taxonomic identification; i.e., the skin, skull, tissue samples, photographs, vocalization recordings, etc. of the same individual should not be allowed to have different taxonomic identifications. It can be argued that membership in a taxonomic group is therefore a property of the Biological_Individual, and not a property of any other kind of Collection_Object. For example, a skull is not a member of species except by virtue of the fact that it was part of an *individual* which was a member of that species. Cases do exist, however, in which another kind of Cataloged_Item (e.g., a photograph or clutch of eggs) may not be accompanied by standard voucher material of the individual; i.e., there is no Biological_Individual record to hold a taxonomic identification, even though an identification does exist for the photograph or clutch of eggs. Requiring an identification to apply to only Biological_Individuals would preclude recording an identification for these other kinds of Cataloged Items. For this reason, we have decided to link the Taxonomic_Identification object to the Cataloged_Item object rather than to the Biological_Individual object.

Taxonomic Identification ORM Diagram

Specimen Citations and Type Status

A special status is conferred upon a specimen when it is the subject of a publication. This is undeniably true for any specimen designated as a name-bearing type in a publication, but a broader view of “special status” might also encompass a specimen that is figured, measured, or otherwise described in a publication. This model adopts the broader view, and supports the general ability to link a specimen (or Cataloged_Item) with a Publication.

A Citation is simply an association between a Cataloged_Item and a Publication. The model provides two fields, Page_Number and Remarks, to hold the particulars of a Citation. (Additional fields may need to be added to the Citation object, such as figure and plate numbers, if the system is used to produce a type catalog.)

Type status has traditionally been recorded in collection databases as a single-valued attribute of a specimen. This approach has a number of shortcomings, including the inability to specify which publication imparted the type status (or which type status – see below), as well as the inability to record which taxonomic name(s) is (are) fixed by the type. The designation of a specimen as a name-bearing type must be made in a publication. Our view, therefore, is that a statement about a specimen’s type status must involve explicit specification of the specimen (Biological_Individual), the Publication, and the Taxonomic_Name, in addition to any information about the type status itself.

A specimen’s type status, however, is not necessarily fixed and permanent. For example, a specimen may be designated as a syntype (or cotype; one of a series) in the original publication, and then selected as the lectotype in a subsequent publication. A specimen’s type status can be modified only through a published work. A specimen can have only one current type status for a given Taxonomic_Name, but cases do occur in which a single specimen has been designated as the name-bearing type for more than one Taxonomic_Name. The model allows multiple Publications to be linked to a single Cataloged_Item, and thus handles changes in type status for a given name, as well as cases in which a single specimen serves as a type for more than one taxonomic name.

While type designation and identification are clearly distinct concepts, a potential disadvantage of using separate data structures to represent them is that each structure creates a separate relationship between Taxon_Name and Biological_Individual. Retrieving specimens by taxonomic names therefore requires searching in more than one place, Specimen_Citation and Taxonomic_Identification. Alternatively, a rule could be imposed on top of these data structures that would make searching by taxa more simple. This rule would require that a comparable Identification record be entered for any specimen that is a name-bearing type. The taxon author could be entered as the “identifier” (Person), and the date of publication could be entered as the identification date. Note that only the year of publication may be known, whereas the database software may require the Identification_date to be a complete date.

In some cases, a specimen may bear some indication that it is a type even though the publication details are unknown. A “work-around” for this problem might be to create a record for “unknown” publications, and to link such specimens to this record temporarily.

Specimen Citations and Type Status ORM Diagram

Publications

The model of publications used here is a simplified version of the CitDB 6.0 schema, a bibliographic database being developed for the Genome Database (Dr. Ken Fassman, pers. comm., 1994). This model is simpler in that it recognizes a narrower range of publication types (i.e., it does not recognize theses, conference papers, and personal communications as distinct types of references or publications), and contains only the data elements commonly found in simple bibliographies.

Publications are divided into three subtypes: Journal_Articles, Books, and Book_Sections (e.g., a preface, forward, or chapter). Note that Journals are not treated as a subtype of Publication because they are not cited in total. Instead, Journal_Articles are cited, and are contained within Journals.

The Agent object is used in the "author" role, because some publications are written by an organization rather than a person. The Agent object accommodates both cases.

Publications ORM Diagram

Collecting Event and Locality Information

Collecting_Event

The Collecting_Event concept in this model has equivalents in the ASC and SMASCH models. It represents the conjunction of time, place, and people (Agent) in an action that produces physical material and/or data. The Collecting_Event object documents the origin of one or more Cataloged_Items; i.e., the circumstances of their removal from nature.

The common practice in collecting higher vertebrates is that, regardless of the number of people in a collecting party, only one collector assigns a collector's number to the collected item and records the collecting information under that collector's number in his/her field notes. If multiple collectors are involved, they are usually listed in the notes.

A Collecting_Event must be recorded for a specimen if any of the relevant information is known; i.e., locality, collection date, or collector. It is being left as an implementation decision whether to represent "no collecting information" (e.g., for laboratory-reared animals) as a specific Collecting_Event or to leave the foreign key to Collecting_Event null.

A "verbatim locality" will be recorded to match, as closely as possible, the way "locality" was expressed by the collector. This data object is attached to the Collecting_Event rather than the Locality because a Verbatim_Locality is typically recorded by the collector in field notes for each visit to the Locality. Consequently, the Verbatim_Locality value is a property of (functionally determined by) the Collecting_Event. The Verbatim_Locality will not be modified except to correct a data entry error. All other fields in the Collecting_Event and Locality tables can be interpreted and updated to achieve greater precision and standardization.

The concept of an "Expedition" has not been represented, at least not explicitly, in MVZ collection catalogs or TAXIR databanks. A large proportion of MVZ specimens, however, were collected on expeditions, defined here as a "long" event, in which one or more researchers visit a series of Localities between a range of dates. We believe this concept can be useful as another high-level focal point for a diverse array of collection materials and in reconstructing the history of MVZ research activities.

Collecting Date

Our strategy for recording the date of collection has been devised to accommodate two requirements: retrieval of Collecting_Events (and Collection_Objects) according to a date range; and the recording (and reporting) of dates exactly as expressed by the collector. Generally, a date specifies a range on a time-line. This is true even when only a single date, such as "June 12, 1965", is reported because a single date can be interpreted as encompassing the time interval 00:00:00 and 23:59:59 on that date. Some collecting dates have been expressed with less precision, such as "June 11-13, 1965", or "Summer, 1934". Almost every expression, however, can be interpreted as implying a begin date and an end date. Queries by date typically use a date-range; i.e., after date X and before date Y. To accommodate both vague expressions of dates and this most common form of date query, we will require that every date be converted to an implied date range (or left blank if interpretation is impossible). This will allow a pre-defined query to operate on the maximum number of Collecting_Events.

The Verbatim_Date field is provided to record and report the date expression exactly as it was given by the collector.

Collecting Event ORM Diagram

Other Legacy Fields

The MVZ's legacy system (TAXIR) has a flat-file structure, and thus all Collecting_Event and Locality data were "repeated" for each specimen collected at the same place, time, etc. The repetition, however, was not always automated, and variation exists among data that are supposed to describe the same event. Data clean-up cannot be completed before the new system is deployed, so two legacy fields have been added to the Cataloged_Item object to accommodate legacy data. Taxir_Specific_Locality will hold data about collecting date and locality, while Taxir_Coll_Name will hold the value(s) recorded for the collector name(s).

Collecting_Event and the Collection_Object Subtype Hierarchy

In most cases, a complete Biological_Individual is collected in a single Collecting_Event (one place, one time). The individual is captured and subsequently prepared into its component parts. A potential problem arises when objects*usually* collected along with the Biological_Individual are obtained significantly before the individual is captured in a "terminal event". We note two kinds of exceptions:

1. An individual is captured and evidence taken (e.g., gut contents or blood sample); the individual is marked, released, and then recaptured days, weeks, or months later. It is therefore possible for related samples or evidence to be collected at one or more times prior to the final collection of the individual.

2. A collector can make a vocalization recording of a bird, and fail in the immediate attempt to collect the individual. Some time later (hours or the next day), the collector may return to the site and be successful in collecting the individual.

The issue that must be carefully considered is: "Which Collection_Object subtype is the appropriate one to 'relate' to the Collecting_Event?" Alternatives and their implications are:

A. Attach Collecting_Event to the top of the Collection_Object hierarchy so that separate Collecting_Events can be recorded for any kind of Collection_Object. The drawback to this alternative is that most Specimen_Parts and Tissue_Samples can only be collected only in conjunction with the individual. An individual cannot survive without its vital parts. (Gut contents and a few minor parts being the only real exceptions.) Keeping information consistent while allowing the possibility of recording separate Collecting_Events, will require: i) user knowledge, ii) navigation-based application control (i.e., the user has no opportunity to enter a separate Collecting_Event_id for each Collection_Object unless certain conditions are met), and/or iii) extensive server programming.

B. Attach Collecting_Event to only those Collection_Object subtypes that can be collected separately from the individual. In this case, the data structure would admit separate Collecting_Events only where they are possible. The problem with this solution is that a subtype may contain mixed cases (e.g., Specimen_Part and Tissue_Sample; an individual could survive without a few feathers, but could not without its skull; an individual could survive without some fraction of its blood, but could not without its liver).

C. Attach Collecting_Event to Cataloged_Item and exclude the possibility of recording multiple capture/observation events for a single catalog number (i.e., Biological_Individual). Biological_Individual is not the appropriate object, however, because other kinds of Cataloged_Items can be collected (e.g., an Egg_Nest, Image, or Vocalization_Series) even though they are not treated as Biological_Individuals in this model.

Alternative C has been used in this model because cases requiring separate Collecting_Event information are *extremely* rare in practice and may not even exist within the Museum's collections. All Collection_Objects derived from a single Biological_Individual are typically collected simultaneously. The motivation for allowing different Collecting_Events to be recorded for related Collection_Objects would be to accommodate research that extends the scope of a typical systematics-oriented collection (e.g., capture-mark-and-release studies).

Collectors' "Numbers"

A Collectors_Number is an identifying number or string applied to a collected item in the field (typically written on a paper tag tied to the specimen). Its function is to establish a linkage between an item and descriptive data written in field notebooks. Note that in Mammalogy, Ornithology, and Herpetology the tradition is to use collectors' numbers as identifying numbers on individual specimens, whereas in other disciplines another concept (e.g., "field number") is used to identify all material, potentially hundreds of individuals, produced from a single Collecting_Event. The two concepts are not synonymous.

Our treatment of Collectors_Number has been influenced by a number of factors. Most important among these is that different collectors have and continue to use very different "numbering systems". A single collector may change his/her system one or more times within his/her career, or may even use more than one system simultaneously. Occasionally, a collector may erroneously assign the same number to two different specimens. Some collectors correct this error by adding a suffix, such as "123-a".

Additional requirements for accommodating collectors' numbers (= field numbers) include:

- Data entered into the system should accurately reflect what is on paper, but the data are not necessarily generated by MVZ personnel and may not conform to an MVZ format standard. This calls for an alphanumeric field. (Some older collectors used characters of the Greek alphabet as "numbers", which have been translated to/interpreted as the full spelling; e.g., "gamma").

- Collectors_Number should sort as numbers rather than strings (i.e., 2 comes before 12).

- Some Collectors_Number have no integer portion; some have more than one. If more than one, it can be difficult to determine which represents the incremental portion.

- Adding Person_id or collector's initials would reduce ambiguity, but would not guarantee uniqueness of a Collectors_Number.

- Tradition within the MVZ is that collector initials are not printed on tags/labels because they would be redundant with collector's name. (In some cases, though, the prefix on a Collectors_Number may not be the collector's initials; e.g., ZP refers to the Zona Protectora project, but in field notes of DBW.)

- Different collectors may have the same name (first, middle, last; most commonly father and son).

We have considered using a simple alphanumeric string to hold Collectors_Number, but rejected this alternative because it wouldn't provide the desired sorting capability. We have instead opted to use a series of three fields, Coll_Num_Prefix, Collectors_Number, and Coll_Num_Suffix. Note, Collectors_Number has the data type "real", rather than integer, to accommodate decimal numbers within the numeric portion of the field group. This will spare data entry staff from having to decide which portion of the decimal to treat as the number versus prefix or suffix. The Coll_Num_Prefix and Coll_Num_Suffix fields are intended to include the separator characters adjacent to the numeric portion of the number, such that a simple string concatenation will produce the collector’s number as written in the field notes (or on the specimen tag). Because many software packages strip trailing and/or leading spaces from string values, the only departure from this rule will be to substitute a dash (-) for any trailing space on the prefix and leading space on the suffix. For example, “ZP 1234” would be entered as:

Coll_Num_Prefix:	ZP-
Collectors_Number:	1234
Coll_Num_Suffix:	

An explicit foreign key is used to indicate the Person who assigned the Collectors_Number. If the primary collector’s initials are not listed as a prefix in their field catalog, they should not be entered as the Coll_Num_Prefix.

Collectors and Collecting Events

Many collectors may participate in a Collecting_Event, and the system must have the capability to store and report multiple collectors in the order given, rather than alphabetically.

Note that the Agent supertype is used in the collector role because it is not uncommon for an Organization to be the collector of record.

The Agent type, Group, is used only to accommodate legacy data where a large team of collectors may have been recorded as “*Person-name* et al”; e.g., “Grinnell et al”. At present, we do not intend to create a Group record (a.k.a. Team, Committee) for every case in which multiple collectors participate in a Collecting_Event. Each collector will simply be linked directly to the Collecting_Event. In other systems, the motivation for creating such a Group record is to enable its re-use, which is clearly efficient when the institution’s data show that many large groups have conducted several Collecting_Events. We do not expect that MVZ contains enough large collector groups to warrant this type of strategy, but it could be adopted should the need become evident.

Locality

A Locality is a place from which a Collection_Object originated. The scope of a Locality is determined by the collector, particularly the way he/she chooses to record information in field notes. Typically, a Locality encompasses an area that a person can walk around within a few hours, and is described in relation to a named place or landmark (geographic feature). It is not uncommon, however, for specimens to lack locality information completely, or for a Locality to be specified only as a large geographic area (e.g., a county, state, country, or region).

Elevation is recorded as a set of bounding values. If the collector specifies only a single value (i.e., does not specify a range, or a point value with a deviation), the upper and lower bounds are set to be equal. Elevation values will be stored in meters to allow the creation of a meaningful index, but the original elevation units will be recorded in a separate field.

For name-based retrieval, each Locality is linked to a record in the Geographic Authority File. The authority file is flat and contains only the political and natural geography fields commonly found in collections databases. This system is simple and has proved sufficient for most retrieval needs, but represents a less powerful approach to name-based retrieval. An earlier version of this model specified a recursive data structure that permitted a named place to have more than one parent (e.g., the United States is contained in both North America and the Pacific Ocean). This would have enabled a mixture of natural and political classifications, as well as synonyms and preferred names, but was deemed too complex for the first implementation.

Fields for Township-Range-Section data are also provided. Interpredicate constraints are given such that:

- Township, Township_Dir, Range, and Range_Dir must all be populated or all left blank
- Section can be entered only if Township etc. have been entered
- Section_Part can be entered only if Section has been entered.

Locality ORM Diagram

Latitude Longitude

The model allows multiple latitude longitude points to be specified for a single locality. This will accommodate multiple determination methods for a given locality, including multiple GPS readings, and determination from a gazetteer or map. Only one point, however, will be allowed as the preferred determination for a given locality and will be presented on standard views and reports.

If the latitude longitude expression is given by the collector as degrees-minutes-seconds, it should be recorded as given, but also converted to decimal degrees. If the original point determination is given in UTM coordinates, these will be recorded, but also translated into the decimal degrees format of latitude-longitude.

No standard has yet been adopted for expressing the precision of a latitude longitude determination. However, we believe the strategy of recording the implied limits of a latitude-longitude expression to hold great promise. This strategy entails recording the implied minimum and maximum for latitude, as well as longitude. If the collector reported +37.52 degrees latitude and +121.45 degrees longitude, these would be recorded as:

min. latitude:	+37.515
max. latitude:	+37.525
min. longitude:	+121.445
max. longitude:	+121.455

This “bounding box” strategy is simple, flexible enough to accommodate any degree of accuracy (from thousands of kilometers to meters), and does not rely on a fixed number of arbitrarily defined classes of accuracy. It is similar to the strategy of recording an x-y point and a radius of certainty, but differs in that it can accommodate long geographic features that would be bounded more accurately by a rectangle than a circle or square.

Latitude Longitude ORM Diagram

Field Notes

The practice of recording extensive field notes was strongly encouraged by the Museum's founder, Joseph Grinnell, and is followed to this day by his academic descendants. The MVZ Library contains approximately 670 volumes of original collectors' field notes that document the activities of researchers and the origins of Museum specimens. A volume of field notes is typically divided into sections, each representing one of the three "standard" sections (journal, catalog, and species accounts) for a collector's activity in a period of time. Some volumes contain the notes of multiple authors, but any one section is always authored by a single person (collector). Page numbers are not always written in the notes, thus the position/page where information for a particular specimen has been entered must be indicated by other data; e.g., author (Person_id), Collection_Date (start date if a range is used), and Collector's number, which should appear only once within a single catalog section. The staff are currently cataloging these volumes, and ultimately intend to link every specimen to the field notes describing the circumstances of its collection. This is actually practical because "Collector", "Collector's Number", and "Collection Date" have already been recorded in TAXIR and constitute a relatively consistent reference to sections of field notebooks (see below).

The field notes of collectors that are still active are not bound in volumes or deposited in the library. In some cases, the Museum only possesses copies of field notes, often not in standard format nor on archival paper in indelible ink. These are present in folders in file cabinets.

Additional facts about field notes and notebooks:

> "Grinnellian style/format" includes separate sections for species accounts, journal, and catalog. (An equivalent of the catalog section is always present, but species accounts and journals may be absent.)
>
> Volume labels (written on the spine of the volume) are not unique. For example, two volumes are labeled "Zink, R.M. 1978-1982", one contains the journal, the other the catalog.
>
> Some volumes are labeled for the vertebrate class they contain (e.g., Birds, Mammals).
>
> Numbering systems within field catalogs can vary widely, even for a single author. Eric Johnson reported that A. M. Alexander started with separate Collectors_Number series for each vertebrate class, then switched to a unified series (continuing from the last of her existing Bird numbers), continued this for a while, then went to annual series (starting with 1 every year), and then continued the last series over several years.
>
> Years on spines don't always indicate what's inside (e.g., Hafner's, San Bernadino Nat'l Forest: spine and blue page say 1974, but catalog begins in 1973.
>
> Contents are not always chronological; field notes may be transcribed from original notes, which can introduce transcription errors (including chronological transpositions).
>
> There are occasional duplicate assignments of the same field number (i.e., same number assigned by the same person to two different specimens).

Field Notes ORM Diagram

Transaction Management

High-Level Transaction Relationships

This first diagram of the transaction management section shows the high-level relationships between Transactions, Shipments, Permits, and Agents.

A Transaction is defined as any action that changes title to or custody of one or more Collection_Objects. Every Transaction must be of one and only one subtype. A generic supertype is used because all Transactions share some data elements and because most Transactions participate in the same relationships with other entities; e.g., Agent_Address, Shipment, Permit, Correspondence, and other Transactions (although the rules for participation may vary among Transaction subtypes).

The MVZ records information about outgoing Shipments only. Relevant information about incoming Shipments (e.g., date of arrival) is recorded in the Transaction record. A single Transaction can be shipped in many Shipments. While it is also possible for a single Shipment to contain more than one Transaction, in practice, there is no requirement for this capability. For simplicity, a one-to-many relationship was described between Transactions and Shipments.

Permits are required for both Shipments and Transactions. The model allows Shipment and Transaction activity to be reported by Permit.

All Transaction types may be conducted with one or more Persons or Organizations (Agents). An object similar to a supertype, the Agent_Address object, was created to represent either a Person or an Organization at an Address. An Agent_Address that conducts a Transaction plays a particular role in the Transaction. For example, a Person receiving a Loan might be the Borrower. The Organization transferring title to material might be the Donor. For any Transaction type, only a limited suite of roles are valid. The valid roles for a given Transaction type are recorded by the “Transaction_Type has valid Agent_Role” relationship, and is stored in an authority file. The relationship between Agent_Address and Transaction is many-to-many, but not mandatory from either perspective. Agent_Role, however, must be recorded for every occurrence of an Agent_Address participating in a Transaction.

The following discussion clarifies the treatment of potentially ambiguous collection transaction activities; i.e., which Transaction_Types should be used in which situations.

Outgoing tissue (for analysis): These cases are to be treated as Loans.

> Policy states that specimens/tissues are to be provided to external researchers for molecular analysis only under the condition that the researcher agrees to return either: 1) a DNA extract of the lent samples; or 2) new tissue and data from a vouchered specimen (not necessarily deposited in MVZ).
>
> Implications/Interpretations:
>
> - The Museum is not relinquishing title to the tissue; the external researcher is borrowing and processing it.
> - A returned DNA extract is not strictly the same object (tissue), but a derived object, related to the original. A new catalog record will be created for the extract (Tissue_Preparation), but not an accession record (title was never relinquished). The date an extract was returned (received) is recorded as a reconciliation of the corresponding Loan_Item (Reconciled_date).

- The researcher should return information describing how the extract was prepared (particularly preparation method and date), which will then be recorded in the catalog, much as if it were prepared at an MVZ lab. (This promotes consistent representation of the same information.)
- If new tissue (or extract) is returned instead of an extract of the "Lent" tissue, a Deaccession transaction will be created for the lent material, and the new material accessioned. The Deaccession and Accession constitute an Exchange. An accession record (as well as new catalog records) will be made for new tissues or extracts ("new" meaning "not derived from MVZ specimens").

Exchange:

> An Exchange implies a "this for that" pair of Transactions, an Accession for the incoming material, and a Deaccession for the outgoing material.

Modeling conventions require that all entities in a supertype-subtypes series be identified by the same primary key; i.e., a single number series -- Transaction_id. Thus, Transaction_id 1001 may be a Loan, while 1002 may be an Accession. This is a significant departure from the existing numbering system in which Loans and Accessions have parallel number series. The solution is to use a separate Loan_Number, Accession_Number, etc., for each transaction type. This filing number is visible to the user, while Transaction_id is hidden.

The existing series of Accession numbers **must** be accommodated because they are already recorded for some 600,000 specimens. Note: although the Accession_Numbers exist in TAXIR data, they will not point to Accession records that exist in the system (referential integrity must be relaxed between Collection_Object and Accession).

High-level Transaction Relationships

Agent

It commonly occurs in collections data that a particular function or role can be performed by either a Person, an Organization, or a Group of people. Adding to this complexity are cases in which: 1) a Group is expressed as an explicit list of people, 2) a Group is indicated only as "*person_name* et al", and 3) a Person or Group is indicated by something other than a proper name – e.g., "Italian Fishermen", or "Trapper", or "Indian". To accommodate all these cases we have created an abstract supertype, Agent, and a series of four subtypes: Person, Organization, Group, Ambiguous_Agent. The "Ambiguous_Agent" subtype is used to record Agents known only by something other than a proper name.

The MVZ maintains a donor-card file, and the presence of a card for a particular agent (either person or organization) is indicated by a logical (yes/no) variable.

Agent ORM Diagram

Agent-Address

The Agent_Address object was created to accommodate the conjunction of a Person, an Organization, and an Address. (Technical note: although the Agent_Address object functions like a supertype, it is not a true supertype because the primary keys of the Agent_Address and Agent entities (Person or Organization), are different; i.e., not drawn from the same domain.)

The model allows an Agent_Address to be valid without an address, but every Agent_Address must represent at least a Person or an Organization (one of the two must be populated to create a valid Agent_Address). An Address must be present, however, for an Agent_Address to participate in a transaction. Note that the Agent_Address object creates a many-to-many relationship between Addresses and the objects that use them, Persons and Organizations. It also records a many-to-many relationship between Persons and Organizations; i.e., multiple affiliations.

Agent_Address records are never deleted. When the need arises to create a new Address for a Person, the previous Agent_Address record is marked as no longer current, and a new Agent_Address record is created for the Person. (A Person or Organization may be represented by many “current” Agent_Address records.) The motivation for preserving old Agent_Address records is that Transaction records are never deleted; they form a permanent on-line archive of collection documentation. To keep an old Transaction record accurate and complete, the Agent_Address records attached to it must also be maintained as they were when the Transaction was active or processed.

Phone numbers are attached to the Agent_Address object, rather than directly to the Person and Organization objects because it is conventionally understood that phones are connected to physical places (even though they are used by people). While cellular phones and call-forwarding invalidate this assumption, we have kept this structure for its capability to capture place-specific phone numbers. Place-independent phone numbers can be accommodated by copying them to all Agent_Address records that represent a single person.

The dashed arrow extending from the role “Agent_Address has Job_Title” to the role “Agent_Address represents Organization” indicates that a Job_Title can only be assigned for those Agent_Addresses that represent an Organization. (It would be inappropriate to have a Job_Title associated with a Person using a personal address.)

In an earlier version, a similar dashed arrow (subset constraint) also extended to the role “Agent_Address represents Person”, indicating that Job_Title is an attribute that is valid only for a Person associated with an Organization. This arrow was removed to allow a Job_Title to be recorded for an Agent_Address in which the actual Person is unknown, but the Job_Title can facilitate proper delivery of correspondence or of a shipment; e.g., Collection Manager or Department Chair.

Agent Address ORM Diagram

Person & Organization

Basic attributes of people and organizations are depicted in this diagram. In addition to the standard (Anglo) suite of names recorded for people (first, last, and middle), the model accommodates prefixes (e.g., Dr. or Professor), suffixes (e.g., Jr., or III) and other names (e.g., nickname, married name, etc.). Date of birth, death, and Remarks can be recorded to help distinguish among people with the same names.

An Organization may represent an entire institution (e.g., The American Museum of Natural History), or a department or office within an institution (e.g., Department of Mammalogy, American Museum of Natural History). Institution name is required as part of Organization_Name because a department name by itself would generally be insufficient to unambiguously identify an organization to a user.

Unlike phone numbers, e-mail addresses often provide a functional means of communication with a Person regardless of where they are physically located (i.e., can be accessed by that person from a variety of locations). E-mail addresses, therefore, are placed in direct relationships to Person and Organization.

Person Organization ORM Diagram

Transaction, Permit, & Correspondence

Permits authorize Shipments and Transactions, and at least one of the MVZ's existing permits is an authorization to operate a facility containing dead animals. Managing information about Permits therefore concerns not just their relationships to Transactions and Shipments, but also the monitoring of expiration dates, renewals, etc.

The data objects common to all Transactions are shown in this diagram. Relating these objects to the Transaction supertype enables these data objects to be inherited to all Transaction types.

All business-related Correspondence is filed in folders labeled with the external correspondent's name (in a *large* series of file cabinets). This includes Correspondence sent and received via fax, e-mail, and even notes of phone conversations. Signed Loan invoices are also treated as Correspondence. The model currently allows each item of Correspondence to be described briefly, and for the external correspondent and filing location to be identified specifically. This structure is tentative, and may be altered to accommodate a more efficient work-flow, which may become apparent once the staff adjust to performing day-to-day operations with the assistance of the transaction management application.

Transaction, Permit, and Correspondence ORM Diagram

Shipment

The model accommodates a basic description of outgoing Shipments. A valid Shipment record must designate either the carrier (and a shipment method), or the individual that is "hand-carrying" the Shipment to its destination. Not all Shipments contain Transactions. For example, large printouts of collections information are sent as Shipments, rather than via "regular" mail. A "contents" field is provided to make the records for such Shipments intelligible to subsequent users.

A Shipment can be flagged if its contents include hazardous materials ("hazmat"). This indicates to front- office personnel that appropriate labels must be affixed to the Shipment.

Shipment ORM Diagram

Accession & Deaccession

An Accession documents how the Museum gained title to one or more Cataloged_Items. In addition to the fields inherited from the supertype (any relationship attaching to the Transaction object), an Accession can be described by the data objects shown on this diagram. Two levels of description are allowed. Gross description is provided by the Nature_of_Material object (see Transaction supertype), and is required to create a valid Accession record. Item-level description is provided by linking Accessions to Cataloged_Items. This is accomplished in the cataloging process.

A Deaccession documents (provides an explanation for) how or why one or more Collection_Objects is/are no longer part of the permanent collections. The catalog record of a deaccessioned item is not deleted because it may still constitute valuable information. Instead, the object's disposition is marked as deaccessioned and an explanation is entered in a Deaccession. (A single explanation may apply to many Collection_Objects, thus there is utility in creating a Deaccession Transaction.)

As noted in the discussion of Transaction types, a Deaccession may constitute part of an "Exchange".

Accession Deaccession ORM Diagram

Loan

Lending a Collection_Object can be viewed as an activity that places the object at risk, at least to some degree, because custody of the object is transferred to an external party. Accurate records about loans are kept to mitigate this risk. The processing of Loans constitutes one of the largest burdens in collections management, and thus the benefit of good record keeping must be balanced against the cost of record keeping. We believe the concepts outlined here strike an appropriate balance.

The model allows a Collection_Object to participate in multiple loans over its life span (from Accession to Deaccession). “Closed” Loans are never deleted, and thus become part of the permanent archive of collections documentation.

Most material that goes out on Loan is cataloged, and thus can be linked to a loan by its Collection_Object_id. Cases do exist, however, in which material is not cataloged before going out on loan; i.e., it is unknown whether such material will ultimately be accessioned. Collectors numbers (Collectors_Number_cde) are used in these cases to track the object. (Note, this version of a collectors number **--** named with the suffix “cde”**--** is not the same as the collectors number recorded for cataloged material; see Collecting_Event diagram.) Either a Collection_Object_id or a Collectors_Number_cde must be present for a Loan_Item to be valid.

The model supports description of an object’s condition, special handling or usage instructions, as well as a general description. Private remarks, which do not appear on the Loan invoice, may also be entered.

Several kinds of events may constitute the end of an object’s participation in a Loan; it may be returned, lost in transit, deaccessioned, etc. (Some of these events require that a corresponding Deaccession record be created.) The term we use for closure is reconciliation. The date of reconciliation and the Person making the reconciliation are recorded for each item.

Policy specifies that all Loans are initially made for a period of six months. It is common, however, for the borrower to request one or more extensions. New due dates may be entered, but do not overwrite any existing due dates.

Loan ORM Diagram

Borrow

System requirements for describing and monitoring borrows are less stringent than they are for other Transaction types. System information about Borrow Transactions is supplemented by photocopies of the lender's invoice and/or relevant Correspondence.

The essential pieces of information are either shown on this diagram, or are available by inheritance from the Transaction supertype (e.g., both the lender and borrower are recorded as an Agent_Address playing a particular Agent_Role in the Transaction).

Borrow ORM Diagram