Arizona Geological Survey ER data model

Stephen M. Richard

June 14, 2001

INTENTION:

Originally was to be a NADM 4.3 implementation. In the course of implementing a database that would allow inclusion of information in existing AZGS databases and would allow for a more complete representation of geologic information the 4.3 model was deemed insufficient. Focus then shifted to the Cordlink variant model as a starting point. Various aspects of this model were found insufficient or unsatisfying. The logical model presented here was evolved to reduce the number of tables and allow greater flexibility and logical consistency. The final implementation resembles the NADM 4.3 model only in very general terms.

Two schemata are included.

Figure 1 is a simplified schema showing representation of a 'Default Visualization', which uses the geologic classification and symbolization of the original map author, which are included in the native GIS data tables (AAT and PAT in ESRI lingo). This schema includes some representation of description—spatial objects, sample locations and structural measurements are included. It does not included the correlation tables necessary for building general relationships between objects. The metadata representation is very schematic (only one table shown).

Figure 2 is a simplified schema showing the general relationship structure, a more in depth (but not complete) view of the feature-level metadata implementation, and the MapView and MapLegend tables that define different visualizations based on the same data. This schema includes some explanatory text.

DESCRIPTION:

The model builds on the design philosophy laid out in Richard (unpublished, 1998, url: http://www.azgs.state.az.us/GeoData_model.pdf), and Richard [1999]. The components of the model are:

1. Classification concepts table(s). At the core of the model is a table or group of tables with similar structure that define terminology. The essential elements of these *ClassificationConcept* tables are a unique identifier, a name, and a definition/description. The unique identifier follows the global unique identifier scheme described below. The name is a string that allows human identification of the concept (e.g. 'basalt'), and the definition/description is a free text field that defines the term or describes its meaning precisely.
2. Relationship tables. These are tables that link data instances. The meaning of the link is defined by a relationship type attribute. Three sorts of relationship tables are included with different structure and application. *Hierarchy Relationship* tables define parent-child relationships in hierarchies. *Simple relationship* tables link data instances in simple assignment, which may have a sequence (e.g. formations in a group). The most complex relationships are *attributed relation-*

ships, which allow an attribute value to be associated with the link, along with a sequencing index, and classification confidence and classification basis attributes.

3. Description tables. These are tables tailored to particular kinds of descriptions. The core model includes tables for structural measurements, text, geochronologic ages, chemical substances, lithologic description, stratigraphic time, spatial objects, and measured quantity. Some of these are linked to *ClassificationConcepts* directly through the sharing of a unique identifier, and provide a structured description to characterize the *ClassificationConcept.* Others provide descriptions of 'real world' instances (a particular rock sample, a particular contact, a particular fault….).
4. Map Visualization tables. These are a set of tables used to define map visualizations. This group includes three core tables: 1) map view definition table, that specifies a title for the map, author, design scale, map extent, symbolization scheme and classification scheme used for the map; 2) map legend, that relates each symbol used in the map visualization to a classification concept; and 3) Cartographic object table, that defines the symbols used on the map in implementation-independent terms. Three modes of defining assignment of symbols to spatial objects represented on the map are used. First, in this database, all spatial objects have a default classification attribute and a default cartographic object attribute. This default classification/symbolization corresponds to that assigned by the original author of the map visualization. Second, symbols may be associated with spatial objects through the map legend, (symbol – classification link) and a spatial classification attributed relationship (spatial object – classification link). This approach corresponds to the NADM 4.3 and Cordlink Variant approach. Finally, spatial objects may be linked to symbols through an attributed relationship link whose type is the identifier for the map view definition. This final approach corresponds most closely to how map visualizations are actually generated from spatial data. The relationship attribute is the rotation to apply to structure measurement symbols, or the text string to display in the case of purely cartographic annotation symbols.

Identification Scheme:

Unique identification of data instances in an internationally distributed data warehouse is achieved by partitioning responsibility for maintenance of unique identifiers. At the top level, each organization providing data to the system must be assigned a NameSpace by the overall system manager. Note that a NameSpace is a *ClassificationConcept*, but its name string must be globally unique. Within each namespace, every data file must have a unique identifier. The system manager must assign a unique identifier number to each data table, geographic data set (coverage, shape file, etc.), image, text file, etc. that will be used by the system. Information about each data file (called a *DataSet* here) is stored in a central *DataSet Table* maintained within each NameSpace. This information must include a physical address (url) for each *DataSet* so that it can be located automatically when accessed. Within each *DataSet*, every data instance has a unique identifier number, generally in the form DataSetName + "ID".

Metadata:

Feature level metadata is implemented by linking every data instance with an origin TrackingRecord, either as an attribute of the instance, or by inheriting origin tracking from the *DataSet* that contains the instance. The TrackingRecord defines a person, organization, and project (an 'activity) that generated the data instance, along with a link to a data processing description for how the information was obtained and

introduced to the database. Each tracking record may be linked (through a *SimpleRelationship*) to one or more bibliographic citations. The metadata scheme is described further in a separate document.

Table naming conventions:

Tables and fields in this model are named following the conventions used by international standards efforts such as UML [OMG, 1999] and the Open GIS Consortium. Names are strings with no spaces. The first letter of separate words in the name is capitalized, and no underscore separates words in the name. Typing underscores is error-prone, and under many display conditions, the underscores may be difficult to see.

SPECIFIC COMMENTS AND DIFFERENCES WITH NADM 4.3

The following table is based on the NADM 4.3 scheme extracted from the BD4.3 Microsoft Access database distributed with GeoMatter II.

NADM 4.3 Table	Comments
Source	Source is more complicated than this construct allows. A Source may be associated with a Project that produced the work. A Source may be unpublished mapping. The processing history used to convert and analog source to digital form should be recorded in the source tracking. A source might be associated with a spatial object that exactly locates the study area (more useful than just lat/long box) What if a source has multiple authors? Shouldn't author-source relationship be many to many? What if a source is a compilation with multiple related citations? If sources are other than published documents, a source should be related to an 'activity' that identifies a particular person (group), working for a particular organization, under the auspices of a particular project, to be the 'author' in a source record. Extent will be an attribute only for geographic data sources. Extent should be a separate entity that may optionally be associated with a source. In the CordLink model, SOA's have similarly defined extents, and could use the same entity? If a SourceType attribute is added, the Type attribute could be used to determine when an Extent would be required for a source. Extent description should have Name (map area name), min/max Lat & Long, and link to extent polygon in geographic coordinates. How does model deal with same data set in different projections? I'd suggest that the Extent description object should be separate from the projection object related to a source. Several sources could have the same extent, but be represented using different projections. Likewise several sources with the same extent could have different subjects. If the source is digital data, what does scale mean? Resolution seems to provide the necessary information. Scale is an attribute of a citation to a published map. Why does a Source only relate to a single classification scheme? In the AZ implementation, this table is combined with Classification_name to become the table named ‘MapViewDefinition’. Much of the metadata role of this Source table is transferred to the ‘TrackingRecord’ links.

Organization	Contact information would be useful to include in Organization entity. What happens if an organization changes its name??
RelatedSource	For more complicated source histories that will evolve as digital data sets are developed, edited, and combined, a source-source relationship would be complicated to used to capture the tracking history. The basic concept here is of sources being related to sources. What needs to be tracked is the relationship between sources and spatial or data objects (descriptions) Source-Source relationship needs to be ordered to establish sequence of source development events. In the AZ implementation, this relationship is implemented as a kind of SimpleRelationship.
Cartographic_ Object	What about annotation on the map? In order to record the full cartographic layout for a geologic map, many cartographic objects must be located and recorded. For example, the location of bar-and-ball symbols on faults, labels for structures, rock unit label locations, etc. This requires a class of spatial objects that are purely cartographic--that is their location is not geologically significant, but needs to be recorded to represent the data visualization. These must be linked to annotation objects that record the text or symbolization information necessary. The direct relationship between a Cartographic_Object and Classification_Object requires that if I want to represent the same thing (aggregation of COA's, 1..M) with different symbols in two different map schemes, I have to create two ClassificationObjects for the same thing. Thus, the ClassificationObject is a convolution of semantics and symbolization. This does not appeal to me. In the AZ implementation, Cartographic_objects have their own identifiers, and represent implementation-independent graphical elements used to symbolize features in a map view.
Classification_Scheme	What if a derivative map is defined with an extent that is different from all the sources the data visualization draws upon? Needs a description/definition to record the reason for the scheme's existance. A particular classificationObject in a particular Classification_Scheme is associated with a unique CartographicObject; the link to the CartographicObject should be explicit in the Classification_Scheme table. In Az implementation, role of this table is combined with Classification_object to become 'MapLegend' (see note in Classification_object section).
Classification_Object	A ClassificationObject is a Classification of COA's. The name and label (and description?) assigned here are particular to the ClassificationScheme/sourceID that the ClassificationObject is linked to. If the DataClassification Relationship is thought of as a kind of COARelation, then ClassificationObject is logically equivalent to COA, and the label, name and ?description? particular to the legend in a particular Classification Scheme would be attributes of the Classification_Scheme relationship. Classification Group is equivalent to a classification of classification objects. To order classification groups on a legend, the groups must be ordered. The Classification_Tree for objects in the Classification_scheme aggregation allows hierarchical organization of the legend, which in combination with the class_seq attribute of the Classification_scheme could also be used to attain the same effect. For a Classification_Object that is simply used as a heading in a particular map leg-

	end, there may be no associated COA, thus the Classification_Object-COA link should be optional. In Az implementation, Classification_object and Classification_scheme are combined into one table named 'MapLegend', which is an ordered aggregation of symbols (cartographic_objects). Each symbol is linked with a classification object (analogous to COA in NADM4.3). More than one symbol may be linked with a single classification object (e.g. inclined bedding and horizontal bedding (symbols) are both bedding (the COA or classification)).
Data_Classification	DataClassification is a kind of COA relationship that has sequence, percent, quality, basis(?), and source attributes. The percent and quality attributes are optional. I think a more flexible relationship mechanism through a construct like the COARelation entity in this model is a better solution; the semantics of the relationship are indicated by the relationshipType, which also dictates the attributes included, and is used in a RelationshipConstraint relationship to determine the types of things that may be linked through a particular relationship type. In AZ implementation, this relationship is implemented as a kind of AttributedRelationship.
COA_Relation	Since COA_Relations will often involve varying degrees of interpretation/confidence, this relationship needs some sort of Quality/Confidence attribute. Also, some indication of the Basis for inclusion in the relationship would be warranted in some situations (X overlies Y, based on drilling logs, based on field observation, based on air photo interpretation...) In AZ implementation, this relationship is implemented as a kind of AttributedRelationship or SimpleRelationship, depending on the nature of the relationship.
Rock_Unit	This table is essentially a rock unit thickness table. Thickness is only relevant to stratified units, and thus is an optional attribute of a Rock_Unit. The rationale for linking RockUnitThickness directly to the COA in this fashion is unclear. The CordLink concept of descriptions seems to me to more clearly capture the meaning of RockUnitThickness. I don't see any way to record that the rock unit thickness is associated with a spatial object representing the location of the thickness determination. Also a given unit may have different thicknesses in different places. Rock_rank is a classification of classifiers, and could equally be represented as a COA_relation between a Rock_Unit_Rank COA and the rock units of that rank. In AZ implementation, Rock_Unit and Rock_Rank are ClassificationConcepts. Rock_Rank and RockUnitThickness are attributes of a RockUnit description.
Stratigraphic_age	Stratigraphic age should apply to COA in general, so stratigraphic ages could be applied to metamorphism, FormalUnit, or Structure. What if an age assignment is compound (Jurassic and Early Proterozoic; Cambrian, Devonian and Mississippian), or involves uncertainty (Miocene or Cretaceous). How can stratigraphic completeness or incompleteness be represented? In AZ Implementation, a StratigraphicTime scale is represented as a collection of named links between upper and lower bounding time-picks. Different time scales are represented using different datasets. Compound stratigraphic ages are represented as AttributedRelationships.
Structure	This entity is a relationship between Classifiers, with attributes of Confidence and RelationshipType, and perhaps sequence. It appears to allow aggregating StructuralTypes into

	a single classification object. If StructuralType is considered a COA (see comments on StructuralType), then this is another COARelationship. LocationAccuracy is a property of a spatial object, not of a link between classifiers. In the AZ implementation, Structures in the apparent sense of this table are ClassificationConcepts.
Formal_unit	A link to a spatial object would allow linking the Formal unit with its type section location. A source for definition of a formal unit is required. In the AZ implementation, a Formal Unit is a ClassificationConcept with an associated description.
Stratigraphic_rank	StratigraphicRank is a classification of Classifications--could be modeled a kind of COA. ClassificationConcept in Az implementation.
Structural_Type	Modifier is like group or rank--it is a classifier of classifiers, and could be represented using a StructuralType hierarchy. StructuralType then becomes a COA. ClassificationConcept in Az implementation.
Lithology	Lith_level is a classification of Classifications, and could be modeled as a COA ClassificationConcept in Az implementation.
Rock_composition	The RockComposition Entity is a description of a rock unit component. The rock components are COA's. The component descriptions are then combined in fractional parts (vol%) to describe a rock unit. This aggregation of descriptive elements into a classification object is analogous to the function of the Structure entity in this model. The model needs to represent the equivalent construct that describes a particular rock sample associated with a particular location--which is done by the Spatial_Object_rock_composition relationship in this model. The aggregation of rock descriptions into a typical description for a COA requires a similar component-unit correlation table, which does not appear in this model. Volume% and Vol_quality are duplicated in Spatial_object_rock_composition link to SpatialObject. In the Az implementation, the rock_composition entity is represented by a 'LithologyDescription' and the fractional aggregation of lithologic components (described by LithologyDescriptions) to form a composite lithologic entity (identified as a kind of ClassificationConcept) is represented as a kind of AttributedRelationship in which the attribute is vol% and classificationConfidence represents vol_quality. The aggregate description represented by a set of AttributedRelationship links may be associated with a spatial object for 'singular object' description, or with a ClassificationConcept to provide a RockUnit description.
Geochronologic_ Age	An IsotopicAge needs to be related to a spatial object (point) that locates the dated rock sample. This table should be thought of as a link into a separate geochronology database that includes complete information on sample description, sampleID, laboratory procedures, analytical data, etc...
Spatial_ Classification	It would make sense to make the SpatialClassification include the ClassificationScheme that it applies to, so that the same SpatialObject could be classified differently in different schemes. Classification should have confidence and basis of classification as attributes.

	Classification should include a confidence attribute. In the Az Implementation, a ClassificationScheme is a ClassificationConcept that identifies a set of Spatial Object-ClassificationConcept links as AttributedRelationships. SpatialObjects may be linked directly to CartographicObjects for symbolization through a set of SimpleRelationship links identified by the MapViewDefinitionID.
Source_Dataset	Disp_priority and disp_visibility appear to duplicate the disp_priority and disp_visibility in the ClassificationScheme. How are these two semantically equivalent attributes related? Need some sort of clarification for the difference between the attributes in sourceDataset and ClassificationScheme. Can a DataSet have more than one source? In Az implementation, this relationship between an information source and a data file are represented by the DataSet Origin TrackingRecord.
DataSet	This entity added for completeness in GeoMatter II template database. Semantics not clearly defined. Similarly named entity in Az implementation plays similar rolc, but includes some of the attributes here associated with Source.
SOA	SOA and SOAValidDesc are added by Geomatter II template to link with descriptions in the CordLink 5.2 sense. See CordLink 5.2 model. The Singular Object concept in NADM4.3 is quite restrictive in that in only allows descriptions to be associated with unique spatial objects. The SOA is an extension of this that uses a correlation table to allow many-to-many association of descriptions with spatial objects. The Desc_type attribute in the GeoMatter II SOA table is analogous to the DataSetID used in Az implementation. Because the use of string or numeric ID values in primary keys for different tables creates serious problems in defining a general unique identifier scheme, in the Az Implementation, a numeric DataSetID replaces Desc_type. The relationship Constraints expressed by SOA_valid_desc must be expressed using DataSetType or explicit identification of DataSets in Az Implementation.
Spatial_Object	Points and lines should have a location uncertainty attribute expressed as a length, providing a scale-invariant means of determining the line type to use for symbolization. What about lines whose existence is uncertain, like a suspected fault? Is an existential confidence attribute also necessary? The semantics of DataSetID are not clearly enough defined
Spatial_Object _Name	This should include a description attribute to define what the name is supposed to mean. It thus becomes equivalent to a COA with a direct link to Spatial objects. This Name mechanism is the only way available in the model to represent the connectivity of a collection of arcs in this schema. In Az Implementation, this concept is represented as a ClassificationConcept.