Figure 1. Simplified Schema for Geoscience database implementation. Map visualization represented by default cartographic and classfication object assignment to SpatialObjects

Stephen M. Richard
June 12, 2001
Arizona Geological Survey

Relationship Entities shown in pink Color

A **CartographicObject** defines links to tables that define definitions of graphical objects and colors. The *DataSetID* for the linked tables serves to indicate what sort of graphical element is being specified. Multiple elements may be used for a single CartographicObject, superimposed in the order specified by sequence.

Geologic Structure symbols present a special problem, because a standard strike-and-dip symbol is considered the same CartographicObject, irrespective of its orientation, and while the same symbol is used for each measurement location (SpatialObject), the symbol is rotated depending on a value (the azimuth) specific to that SpatialObject. SpatialObject must be joined with CartographicObject through an attributed relationship(s) that includes the rotation value as its attribute. The *CartObjType* may be used to determine if the symbolization depends on an AttributedRelationship.

A **MapLegend** is a relationship that links a ClassificationConcept with an implementation-independent CartographicObject to symbolize objects belonging to the class, and assigns a Name, Label, and Description for objects of that class. Different symbols may correspond to the same ClassificationConcept (e.g. symbols for horizontal, inclined, vertical, and overturned planar bedding).

A **ClassificationScheme** is an AttributedRelationship that links SpatialObjects with unique ClassificationConcepts. Each link is assigned a classification confidence (CConf) and classification basis (CBasis).

A **SymbolizationScheme** is a relationship linking SpatialObjects with CartographicObjects. The link may be a SimpleRelationship if symbolization is independent of any attribute of the spatial object (e.g. line symbolization), or may be an AttributedRelationship, in the case that the symbolization depends on an attribute(s) of the spatial object. Examples of the second case include display of structure symbols (bedding, foliation), which must be rotated appropriately to represent azimuth of the structure, and Map unit labels, in which case the CartographicObject defines the text formatting (through GraphicTextFormat), and the SpatialObject-CartoGraphicObject link has a StringValue that determines the text to display.

A **MapView** is a collection of SpatialObjects within a bounded area (the Extent), classified using a particular ClassificationScheme, and symbolized using a particular MapLegend. The MapView does not necessarily use all the items in the MapLegend, or all the SpatialObjects classified under the ClassificationScheme. Every ClassificationObject in the ClassificationScheme that is related to a SpatialObject displayed in the MapView must have a CartographicObject assigned by the MapLegend associated with the MapView.

 - **SimpleMapView:** All SpatialObjects symbolized in the MapView are entirely within the MapExtent, and the set of CartographicObjects in the MapLegend is the same as the set of CartographicObjects used to symbolize spatial objects in the view.
 - **GeneralMapView:** SpatialObjects may come from different DataSets that may have extents different from the MapView extent, and the MapLegend may include CartographicObjects not in used in the MapView. SpatialObjects symbolized in the MapView must be clipped to the MapExtent, and the MapLegend must be filtered to select only the items that appear in the MapView.

MapViewDefinition defines a Title, Description, Extent, Projection, DesignScale, MapHorizon, ClassificationScheme and MapLegend to use for a particular MapView.

The *ViewSchemeType* in the MapViewDefinition determines how the MapView is constructed. In addition to specifying if the view is a GeneralMapView or SimpleMapView, the ViewSchemeType also varies along a second dimension based on how the link between CartographicObjects and SpatialObjects is defined, as follows:

 - **DefaultMapView.** Represents a default visualization of a geologic data set. Default ClassificationObjects, CartographicObjects, necessary CartographicObject attributes (e.g. rotation for strike-and-dip symbols) and feature-linked annotation (polygon labels, dip values) are assigned using fields embedded in the SpatialObject Tables. SimpleRelationship aggregates the DataSets containing the SpatialObjects through a simple relationship of type *MapViewID*; sequence attribute establishes display order for DataSets. All DataSets contain data within same MapExtent. MapLegend can be produced through a Query –all unique ClassificationObject/CartographicObject Pairs included in the set of SpatialObjects represented. MapLegendID and ClassSchemeID are not required, but a predefined MapLegend is necessary to structure the MapLegend display, display order, and explanatory name, label and text (ClassName, ClassLabel, ClassDesc) for features; otherwise the default legend layout for the particular GIS implementation will be used.
 - **DirectMapView.** *ViewMapID* is a RelationshipType for a SymbolizationScheme Relationship linking SpatialObject with CartographicObject, and MapLegendID identifies the appropriate MapLegend objects. All CartographicObject used must be included in the MapLegend. The *ClassificationSchemeID* link in the MapViewDefinition identifies the classification scheme used as the basis for assigning symbols to spatial objects. The direct scheme is necessary for indivially varying symbolization (e.g. structure symbols), and also allows for map generalization in which an object classified in the same way may be symbolized differently.
 - **NADM43MapView.** SpatialObjects are linked with ClassificationObjects through a ClassificationScheme specified by the MapViewDefinition, and ClassificationObjects are linked with CartographicObjects through the MapLegend. Assignment of CartographicObjects to SpatialObject requires two joins, and ClassificationObjects are conceptually equivalent to CartographicObjects because in order to be symbolized an object differently, it must be classified differently. Thus in order to rotate structure symbols to the correct display azimuth, ClassificationObjects for each azimuth must be generated, or the azimuth attribute of the data to symbolize must be propagated from the structural measurement table, through the SpatialObject, ClassificationScheme link (SpatialObject-Classification), and MapLegend link (Classification-CartographicObject).

A map visualization is constructed in layers that determine which features may overlay or mask other features. In ArcView terminology, each of these layers is a theme, and features included in a particular layer must have a source in the same dataset. Conceptually, the features in a particular layer may come from multiple datasets.

The MapLegend must explain the graphical elements used to symbolize features that are specific to a particular geologic map. Some graphic elements may be considered standard, and may not be included in the legend display.

SimpleRelationship used for linking instances of any two entities in which no uncertainty is involved. These include aggregations of parts, and linking spatialObjects to CartographicObjects for symbolization

AttributedRelationship Used for linking instances of two entities in which each relationship instance is assigned one or more attributes. This table is constructed to allow up to 5 attributes: Confidence (CConf), Basis (CBasis), StringValue (any string), NumberValue (any number), or Attribute (a link to another object in the database). Relationship constraints on RelType specify which attributes may have values and the domains of those values. Examples of AttributedRelationships include geologic classification of spatialObjects, and various kinds of fractional analyses (e.g. chemical analysis, modal mineral analysis, grain size distribution).

A mechanism for representing constraints on relationship role fillers and cardinality is not yet modeled.

HierarchyRelationship table represents transitive relationships (parent-Child, IsA) by including records between each object and all other objects it is linked to via transitive links.

TrackingRecordID/DS link for each object is direct link to a TrackingRecord of *TrackingRecordType* 'OriginTracking' (a ClassificationConcept) that provides information on the source of the object. The TrackingRecord data structure includes links to an *Activity* (tuple of person, organization, project) responsible for origin, a ProcessingMethod that describes how feature was processed to get in digital form. Links to citations for publications relevant to the origin of the information are constructed through a MetaDataRelationship link (see Metadata Schema).

Log entries to document updates, comments to any instance are tracking records of *TrackingRecordType* 'LogEntry' (a ClassificationConcept) linked to the object using a MetaDataRelationship link (see Metadata Schema).
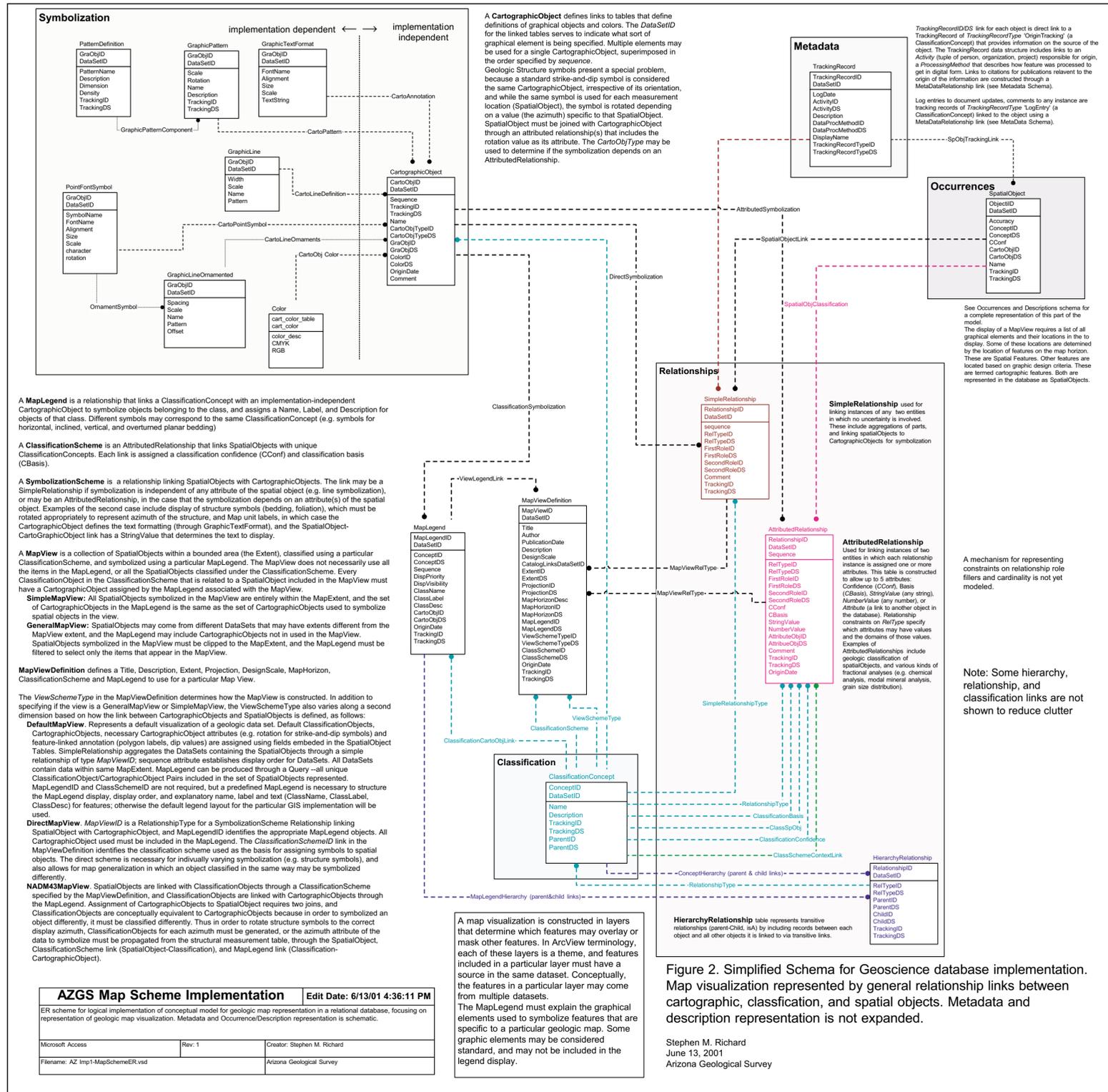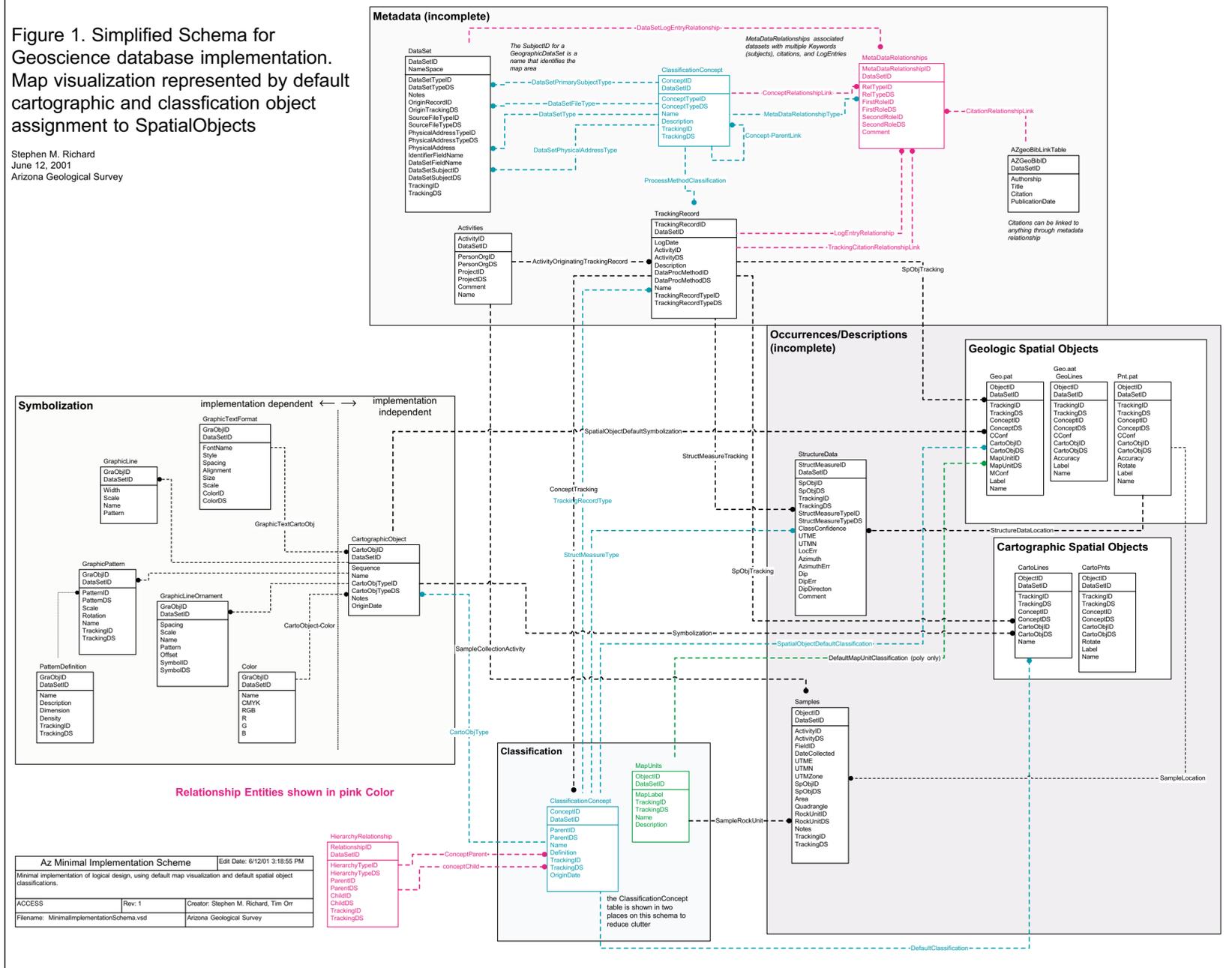
Figure 2. Simplified Schema for Geoscience database implementation. Map visualization represented by general relationship links between cartographic, classfication, and spatial objects. Metadata and description representation is not expanded.

Stephen M. Richard
June 13, 2001
Arizona Geological Survey

The SubjectID for a GeographicDataSet is a name that identifies the map area

MetaDataRelationships associated datasets with multiple Keywords (subjects), citations, and LogEntries

Citations can be linked to anything through metadata relationship

See Occurrences and Descriptions schema for a complete representation of this part of the model.

The display of a MapView requires a list of all graphical elements and their locations in the display. Some of these locations are determined by the location of features on the map horizon. These are Spatial Features. Other features are located based on graphic design criteria. These are termed cartographic features. Both are represented in the database as SpatialObjects.

Note: Some hierarchy, relationship, and classification links are not shown to reduce clutter

the ClassificationConcept table is shown in two places on this schema to reduce clutter

Az Minimal Implementation Scheme — Edit Date: 6/12/01 3:18:55 PM
Minimal implementation of logical design, using default map visualization and default spatial object classifications.

| ACCESS | Rev: 1 | Creator: Stephen M. Richard, Tim Orr |
| Filename: MinimalImplementationSchema.vsd | | Arizona Geological Survey |

AZGS Map Scheme Implementation — Edit Date: 6/13/01 4:36:11 PM
ER scheme for logical implementation of conceptual model for geologic map representation in a relational database, focusing on representation of geologic map visualization. Metadata and Occurrence/Description representation is schematic.

| Microsoft Access | Rev: 1 | Creator: Stephen M. Richard |
| Filename: AZ Imp1-MapSchemeER.vsd | | Arizona Geological Survey |