SEQUOIA 2000 METADATA SCHEMA FOR SATELLITE IMAGES

Jean T. Anderson and Michael Stonebraker
EECS Department
University of California, Berkeley
Berkeley, California

Abstract

Sequoia 2000 schema development is based on emerging geospatial standards to accelerate development and facilitate data exchange. This paper focuses on the metadata schema for digital satellite images. We examine how satellite metadata are defined, used, and maintained. We discuss the geospatial standards we are using, and describe a SQL prototype that is based on the Spatial Archive and Interchange Format (SAIF) standard and implemented in the Illustra object-relational database.

1 Introduction

The Sequoia 2000 project is developing a computing environment to support global change research, involving large volumes of complex data in many formats [1,2].[1] The design goals for the database schema that manages these data include:

- Develop a database schema library composed of user-defined types and SQL templates that support earth science data. A researcher uses these building blocks to create a database schema for a specific application.
- Include enough metadata so files can be generated in widely-used scientific formats, such as netCDF and HDF.

This paper focuses on the metadata schema being developed to manage satellite images. Section 2 looks at satellite data and the metadata associated with a satellite image. Section 3 describes an approach for basing a database schema on emerging geospatial standards to accelerate development of the schema library and support interoperability. Section 4 describes a prototype implementation, which is based on a geospatial standard and uses the Illustra object-relational database to manage the metadata and satellite images. Section 5 addresses metadata maintenance.

[1]This research was sponsored by the University of California and Digital Equipment Corporation. Other government and industrial sponsors include the California Department of Water Resources, U.S. Geological Survey, Construction Engineering Research Laboratory (CERL) of the U.S. Army Corp of Engineers, the National Aeronautics and Space Administration (NASA), Epoch systems, Inc., Hewlett-Packard Corp., Hughes Aircraft Company, MCI, Metrum Corporation, PictureTel Corporation, Research Systems, Inc., Science Applications International Corporation, Siemens Inc., and TRW Space and Electronics.

2 Digital Satellite Data

To set the stage for our discussion of satellite metadata, this section describes data from the Advanced Very High Resolution Radiometer (AVHRR) sensor on the NOAA satellites. First we introduce some satellite basics and describe a satellite image. The satellite images described in this paper are end products produced by the EROS Data Center[3]. Next we identify the metadata elements specific to a satellite image as well as some broader categories of metadata associated with a data set.

2.1 Satellite Image Basics

The AVHRR sensor has five channels that collect measurements in different wavelengths. The first two channels collect visible and near-infrared measurements, and the last three record thermal measurements in different spectral resolutions. The point directly below the satellite has a spatial resolution of about 1 kilometer and becomes increasingly distorted further away from the orbit path because of the wide viewing angle and the curvature of the earth.

After data are transmitted to ground stations, the measurements are calibrated and corrections are applied. Data are corrected geometrically to reduce distortion, and location information is added. The data are then released for further processing [3,4].

These satellite images are perceived slightly differently by computer and earth scientists. The computer scientist focuses on structure and treats each image as a three-dimensional raster (a regular grid). The dimensions for this data set are 2889 rows x 4587 columns x 10 deep. The earth scientist, however, focuses on the processes that created the image. From his or her point of view, the image has 10 *bands* or *layers*, each created by a different process. Each band is composed of 2889 lines and 4587 samples. The database metadata schema should support both viewpoints.

2.2 Satellite Image Metadata

How researchers access the data largely drives which metadata are stored. Dozier [2] provides two examples of how researchers would like to query raster data:

- Select all daytime AVHRR images for the Southern Sierra Nevada between October 1991 and June 1992 and sort chronologically.
- Find the Landsat Thematic Mapper image closest to April 1, 1992, for the Bubbs Creek area that is cloud-free and has snow. Map the snow-covered area and snow grain-size index.

Both queries need date information (to retrieve images that match the specified date) and geographic information (to restrict the spatial search to a specific region). The second query additionally requires access to information in the bands. In other words, the metadata supporting these queries must include temporal and spatial information as well as the structure of the raster. Table 1, below, summarizes metadata for a single satellite image from [3], organized into three sections described in greater detail: **Grid**, **Spatial Registration**, and **Temporal Information**.

The **grid** for the sample AVHRR data set was introduced briefly in the previous section; *i.e.,* it is made up of 10 bands, and each band has 2889 lines and 4587 samples. The first five bands contain the calibrated measurements from the five channels on the satellite sensor. The rest of the bands contain information added by the EROS Data Center. Band 6 contains the normalized difference vegetation index (NDVI), which is calculated from the first two bands and indicates the degree of "green-ness" of a pixel. Bands 7-9 contain satellite geometry information. Finally, these images are *composites* that were constructed from images collected over a 14-day period to produce cloud-free images. Each pixel in band 6 (NDVI) contains the "best" NDVI measurement for that pixel from all images. Band 10 tracks the source image that was used for each pixel.

The image must be **spatially registered** to a geographic coordinate system using a standard projection before many operations can be performed, such as overlaying an image with state or county boundaries or transforming it to a different map projection.

Table 1: AVHRR Metadata

Grid		
No. lines	2889	
No. samples	4587	
No. bands	10	
bands	AVHRR channels 1-5, NDVI, satellite zenith, solar zenith, relative azimuth, pixel source date	
units of measure	meters	
pixel size	1000 (meters)	
Spatial Registration		
map projection	Lambert Azimuthal	
lat/lon of center	100 West	45 North
false easting/northing	0	0
nw corner	-2050000	752000
Bounding box:		
lower left lat/lon	-119.9722899	23.5837576
upper left lat/lon	-128.5300591	48.4030555
upper right lat/lon	-65.3946489	46.7048989
lower right lat/lon	-75.4163527	22.4793919
Temporal Information		
Begin date	4-Jan-1991	
End date	17-Jan-1991	

This data set used the Lambert Azimuthal Equal Area projection, which is optimal for covering large regions. The central meridian for the projection was located at latitude 100 West, longitude 45 North (Gettysburg, South Dakota). The upper north west corner of the image is registered at -2050000, 752000, provided in *eastings* and *northings*, which are the number of meters from the projection center (negative numbers are west of the central meridian). The latitude/longitude coordinates for each corner of the image form a bounding box used in spatial searches.

Temporal information allows extracting subsets of the data. A satellite image may have a single date and time the image was taken. Composite images, such as the one referenced in Table 1, have a date range spanning the dates of all images used to create the composite.

2.3 Satellite Data Set Metadata

An image is stored as part of a larger data set, and a site may have many data sets. Additional metadata describe the data holdings for a site so researchers can find out what is available. Although this paper primarily focuses on raster metadata, we define the following broad categories of metadata for a satellite image data set:

- **Raster:** As discussed above, raster metadata include grid structure, and spatial and temporal information.

- **Data Lineage:** Lineage metadata include the processing history, algorithms and parameters that were used to produce an image. Better algorithms may be developed for processing images, or anomalies may be discovered in images years after their creation. For example, a better NDVI algorithm might be developed or the raw data may have been calibrated incorrectly. A researcher analyzing an image wants to know the processing steps that created an image as well as detailed information about the algorithms. He or she also might want to identify all rasters that were processed with a given algorithm; perhaps to reprocess them using a different algorithm.

- **Data Set Description:** Researchers want to know what data sets are available at a particular site as well as detailed information describing each data set. High-level site catalogs and data set descriptions are critical for data browsing tools such as TIOGA [5] and TECATE [6]. Metadata typically include geographic region, dates the data set covers, search keywords, data availability, and contact information.

- **Object Descriptions:** Researchers often need to know the structure of a database table or the specific properties of an attribute; for example, *longitude* is a single-precision float that must be between -180 and +180. Schema information is also needed to generate transfer files in a specific format, such as netCDF or HDF.

3 Geospatial Data Standards

Sequoia schema development is drawing from emerging geospatial standards to accelerate development and support interoperability. So far we have been working with the Spatial Archive and Interchange Format (SAIF) [7] and the Content Standards for Digital Geospatial Metadata (FGDC) [8], which are among the contributors to the Open Geodata Interoperability Specification (OGIS) [9].

SAIF was accepted as a draft standard by the Canadian General Standards Board in 1991. It is an extensible, object-oriented data modeling and archival/transfer standard that targets both geospatial and non-geospatial data.

The Federal Geographic Data Committee (FGDC) effort to specify a digital geospatial metadata standard began in 1992, resulting in a final draft published in 1994. It adopts a simpler data model organized into a hierarchy of data and compound elements needed to document a geospatial data set.

The FGDC gave the Open GIS Foundation the task of defining an interoperability specification for geospatial database interfaces and geoprocessing capabilities. Work on OGIS began in 1993.

Current Sequoia schema development is using the SAIF and FGDC metadata standards as starting points. SAIF is highly extensible and committed to integration with other standards. The FGDC metadata standard is particularly useful for high-level metadata and appears to be gaining popularity in the geospatial user community (for example, OGIS is proposing to use it for its high-level data set and catalog descriptions). While OGIS is developing a comprehensive internal data model, as of this writing the specification is still a moving target. Once OGIS is complete and services are implemented, it should not matter whether a Sequoia researcher implements a schema based on SAIF or the FGDC metadata standard, or any other supported standard, since interoperability will be handled by OGIS services.

It is important to appreciate that the data models underlying different standards do not map exactly to each other. Furthermore, we realize from our initial experience with the FGDC metadata and SAIF standards that we would probably need to extend any standard for Sequoia earth science data. Reliable methods are needed to transform between data models and describe user-defined extensions without losing information. Finally, we need the ability to extend the underlying data model that provides a transformation service. Designing a schema that encompasses all Sequoia data has been challenging and we do not expect any standard to anticipate every real-world object we want to share. Rather, transformation services should support an evolving schema and should provide a mechanism for extending the underlying model.

Transformation between data models, or schemas, is an important research area. Rosenthal [10] describes an approach that uses a unified underlying data model comprised of the union of all features, noting that the complexity of a unified model is increased by any extensions made to it. Bright [11] uses a global data structure, generated from an online system taxonomy, that contains concise descriptions of data available in all input schemas. The resulting global structure (the 'summary schema') is smaller and easier to modify than a union of all schemas. Sciore [12] describes an approach that is independent of any data model. Instead, the metadata (the 'context') is attached to the value and conversion functions handle transformations between source and target contexts. These approaches are applied to the relational model; each would need to be extended to handle object-oriented capabilities.

Realistically, we do not expect to see tools that effortlessly provide interoperability services for some time. Adhering to standards today gives us a common framework and a less ambitious degree of interoperability using tools the standards community provides or tools we develop ourselves.

4 Prototype Schema Implementation

We implemented a prototype schema based on SAIF using the Illustra object relational database,[2] which supports SQL3 object-oriented features such as inheritance, polymorphism, and user-defined types. The Illustra spatial extension supplies the spatial data types, operators, and access methods needed to manage and query geospatial data. We also use its large object data types to manage the satellite images.

First, we briefly introduce the SAIF data model, then show the AVHRR metadata from Table 1 loaded into a SAIF view in Table 2. Next we describe the SQL implementation in Illustra, also discussing the SAIF types we modified and a few weaknesses in our implementation.

The SAIF standard includes many components, but we have focused primarily on its data model, which we organize into three layers:

- *Building blocks* are base objects that include enumerations and primitive types. An enumeration is a list of named values. For example, the *UnitMeaning* enumeration identifies the real-world units stored in another variable. Its domain of values is large and include *metres, miles, degrees, radians, years,* and *seconds.* Primitive types correspond to the types used in a programming language, such as *real, integer,* and *string.*

- *Generic constructs* use the building blocks to define real-world objects. For example, we used the SAIF **Grid** and **StructuredData** types in our AVHRR raster prototype.

- The *SAIF Standard Schema* defines over 300 classes to form the base standard. SAIF recognizes that no one will implement the entire schema and provides rules and a syntax for defining a subset of the standard, called a *profile.*

[2] Many thanks to the following individuals for early brainstorming on what a SAIF SQL raster might look like: Kenn Gardels of U.C. Berkeley CEDR-REGIS, Sam Gustman of the U.S. Army Corp of Engineers CRREL, and Mark Sondheim of the British Columbia Surveys and Resource Mapping Branch.

Table 2, below, summarizes a SAIF view of our sample AVHRR metadata, divided into the following parts that are described in detail: **Grid**, **StructuredData**, **GeneralLocation**, **TimeObject**, and **SpatialReferencing**. SAIF types are listed in the left column; compound types are italicized, and attributes are in regular type. The sample AVHRR metadata are listed in the right column.

Table 2: SAIF Raster

Grid	
positionOfOrigin	-2050000, 752000
GridReference:	
ordering	rowOrder
originCorner	nw
cellPositionAtCorner	false
originCoordinates	0, 0
GridFramework:	
gridDimensions	2889, 4587
gridSpacing	1000, 1000
units	meters, meters
OffsetStructure	none
StructuredData	
contentValues	actual image
ContentStructure:	
indexingScheme	sequentialArray
attributeNames	AVHRR channels 1-5, NDVI, satellite zenith, solar zenith, relative azimuth, pixel source date
GeneralLocation	
BoundingBox:	
minPoint	-119.9722899, 23.5837576
maxPoint	-65.3946489, 46.7048989
TimeObject	
Duration:	
startTime	4-jan-1991
lengthOfTime	14
SpatialReferencing	
Projection	LambertEqualArea
centralMeridian	-100, 45
falseEasting	0
falseNorthing	0

Grid describes the geometry of the raster. SAIF adds many metadata elements that generalize a grid, allowing it to be defined in any number of dimensions and in a variety of shapes. While this is more functionality than an AVHRR raster needs, it allows describing other types of gridded data, such as global and regional climate models. Grid attributes include:

- *positionOfOrigin*, which stores the coordinates that map a corner of the image to a location on the ground.

- *GridReference*, a compound type that orients *positionOfOrigin. ordering* specifies how data are ordered. *originCorner* indicates which corner of the image *positionOfOrigin* references. *cellPositionAtCorner* indicates if the origin of a cell is positioned at the corner ('true') or in the center ('false'). *originCoordinates* specifies the grid cell coordinates of *positionOrigin.*

- *GridFramework*, a compound type that describes the grid structure. The *gridDimensions* array stores the size of each dimension. The *gridSpacing* array stores the pixel size for each dimension in real-world units; *i.e.,* **1000** meters. *units* stores the real-world unit of measure for *gridSpacing* in each dimension; *i.e.,* 1000 **meters**. *OffsetStructure* indicates if an offset structure applies, such as would be used for a triangular or hexagonal grid. We list it as 'none' for a regular grid.

StructuredData stores the data and includes:

- *contentValues*, which contains the actual data.

- *contentStructure*, a compound type that further describes the *contentValues* structure. *indexingScheme* specifies if data have been ordered physically in a specific format, such as run-length encoded. *attributeNames* is an array of names.

GeneralLocation contains the bounding box coordinates for the image, and **TimeObject** contains temporal referencing. Finally, **SpatialReferencing** contains projection information, which we include in Table 2 for completeness. However, projection handling has not been implemented yet and is part of continuing design efforts.

We implemented the SAIF type hierarchies in Illustra, defining enumerations as new base types and compound types as composite types and tables. Figure 1 shows the resulting SQL. (Note that *GridOrder, OriginReference, OffsetStructure, UnitMeaning, IndexScheme,* and *SatelliteBands* are enumerations. For brevity, we omit the steps that create them as base types.)

We added the *SatelliteBands* enumeration and, to take advantage of Illustra SQL features, replaced the following SAIF types with Illustra types:

- The Illustra *Pnt* type replaced the SAIF *Point* in *positionOfOrigin. Point*, intended for single points, carries ancillary information for each coordinate that would be redundant for the large amounts of point data in some Sequoia data sets. The Illustra *Pnt* is composed of the pair of coordinate values and supports spatial searches.

- The Illustra *Box* replaced *minPoint* and *maxPoint* in *BoundingBox* to take advantage of Illustra spatial search capabilities.

- The Illustra *image* type replaced the primitive type in *contentValues. image* is a fully typed large object with supporting functions that allow manipulating data directly from SQL.

Figure 1: SQL for SAIF Types

```
create type GridReference
(   ordering                  GridOrder,
    originCorner              OriginReference,
    cellPostitionAtCorner     boolean,
    originCoordinates         arrayof(integer) );
create type GridFramework
(   gridDimensions            arrayof(integer),
    gridSpacing               arrayof(double precision),
    offsetStructure           OffsetStructure,
    unit                      arrayof(UnitMeaning) );
create table Grid
(   positionOfOrigin          Pnt,
    orderAndOrigin            GridReference,
    spacing                   GridFramework );
create type ContentStructure
(   indexingScheme            IndexScheme,
    attributeNames            arrayof(SatelliteBands));
create table StructuredData
(   contentStructure          ContentStructure,
    contentValues             image );
create type BoundingBox under Box;
create type GeneralLocation
(   range                     BoundingBox);
create table Duration
(   startTime                 timestamp,
    lengthOfTime              integer);
```

Figure 2, on the next page, shows the SQL statements that use the building blocks created in Figure 1 to create a sample raster table and insert one row based on the sample AVHRR data in Table 2.

The following query finds all AVHRR images between October 1991 and June 1992 for the Southern Sierra Nevada, ordering the results chronologically:

```
select    startTime, contentValues from Raster
where     sensor = 'avhrr'
and       startTime between
          '1991-10-01 00:00:00' and '1992-06-01 00:00:00'
and       Contains(range, '(-122, 35, -117, 38)')
order by startTime;
```

With a constraint on time to eliminate nighttime overpasses, this would perform the first Dozier query in Section 2.2. This query could be simplified by replacing the *Contains* expression, which requires the researcher to know the bounding box coordinates, with a user-defined SQL function that accepts a placename as a parameter and looks up the coordinates automatically. The next query uses such a function (*FindLocation*) and introduces additional functions that would map the snow data (*MapSnow*) and find the image closest to a given date (*NearestDate*). The second Dozier query would look like this:

```
select   MapSnow(contentValues) from Raster
where    sensor = 'landsat'
and      NearestDate(startTime,'1992-04-02 00:00:00')
and      FindLocation(range, 'Bubbs Creek');
```

Figure 2: SQL for Raster Create and Insert

```
create table Raster
(   sensor varchar(20),
    satellite varchar(20),
    bands integer)
Grid, StructuredData, GeneralLocation, Duration;

insert into Raster
(   sensor, satellite, bands,
    positionOfOrigin,
    orderAndOrigin,
    spacing,
    contentStructure,
    contentValues,
    range,
    startTime,
    lengthOfTime) values
(   'avhrr', 'noaa-11', 10,
    '(-2050000, 7520000),'
    GridReference('rowOrder', 'ne', 'f', '[0,0]'),
    GridFramework ('[2889,4587]', '[1000,1000]',
        'none', '[metres,metres]'),
    ContentStructure('sequentialArray',
        '[chan1, chan2, chan3, chan4, chan5, ndvi,
        satellite_zenith, solar_zenith,
        relative_azimuth, pixel_source_date]'),
    '/data/f9101046',
    '(-119.9722899, 23.5837576,
        -65.3946489, 46.7048989)',
    '1991-01-04 00:00:00',
    14);
```

Finally, we note a few flaws in our prototype:

- More band information is available than the name we stored in *attributeNames*. While band information for a given sensor is static, new bands may be added by researchers. We should create a new SAIF type that completely describes each band.
- We need to implement the SAIF *SpatialReferencing* type to manage projections.
- We broke some rules by ignoring the supertypes the SAIF *Raster* inherits. The next iteration should incorporate high-level SAIF constructs.

5 Metadata Maintenance

This section returns to the metadata categories described in Section 2, discussing how each is maintained and to what degree maintenance can be automated.

5.1 Raster

We can automate raster metadata extraction and insertion into the database easily if the raster is in a standard format, such as netCDF. UNIX scripts extract the header information and form the SQL insert statement. If a raster is derived from another raster, for example a region is clipped and produces a new raster with different dimensions, the clip function can register the new grid information.

5.2 Data Lineage

We insert and maintain lineage information at the raster level with Illustra database rules that capture processing information and insert it into lineage tables. For example, the SQL statement that inserts a raster also triggers an insert into the lineage table to describe the creation. If a calculation is performed, information about any source raster(s) along with the function and parameters passed to the function must be tracked. Versions of the function (or program) that was applied must also be tracked.

Lineage must also be tracked at the sub-raster level and we are considering strategies for storing lineage for each pixel or for groups of pixels. One option is to simply add another band to the image like the AVHRR sample data set did. Section 2.2 described how a composite image was produced by storing the "best" pixel from a series of images, with band 10 storing the source of each pixel. The most inefficient option would be to store lineage information for each pixel as a separate row in the table.

5.3 Data Set Description

Entry and maintenance of high-level metadata is labor-intensive and challenging to automate. Currently, data catalog "inventories" require manual intervention to evaluate and insert the information into the database and define a set of keywords for searching. But most researchers do not have sufficient time to provide comprehensive details for a data set. A hierarchical approach to metadata management is important, requiring a small set of information at the top that gives researchers an entry point from which they can query for optional details the data provider is able to supply.

Text-based systems will become increasingly important to automate metadata entry of data set descriptions. For example, the GIPSY [13] prototype automatically extracts geospatial information from text documents.

5.4 Object Descriptions

Entry and maintenance of object descriptions can be partially automated, first by taking advantage of database system catalogs that maintain base information about all user objects. An ancillary object description table stores additional detailed information. Wherever an object is adopted from a standard, it is possible to load its description from electronic documents provided by the custodian of the standard. Of course, any user-defined objects depend on the user to provide descriptions.

We are developing database schema tools that merge information from the system catalogs and object description table, and output object information in a variety of formats. Initially we have focused on extracting the information to output create statements for an object or set of objects. However, the same information could be output in a different format, such as an FGDC metadata description file.

6 Conclusion

The design goals of the Sequoia 2000 schema are to provide researchers with schema building blocks to create application-specific schemas and a metadata infrastructure that supports generating data files in a variety of scientific formats.

We described a strategy for developing a database schema based on geospatial data standards to support these goals. Sequoia can accelerate development by drawing what it can from the available standards and extending them as needed to satisfy earth science requirements. The key challenge we foresee is transferring user-defined extensions between different data models without losing information and, in particular, extending the underlying data model that provides the transformation services.

Finally, we described how metadata are used and maintained. We are able to automate maintenance for all metadata categories to a degree, except for high-level data catalogs. Since they will be broadly accessed, they need more automated entry.

While this paper focused on management of satellite images, Sequoia schema development addresses the broader needs of earth science data. Many of the metadata elements discussed in this paper also apply to other gridded, georeferenced data.

References

[1] Stonebraker, M., "An Overview of the Sequoia 2000 Project," *Proc. 37th IEEE Computer Society International Conference: COMPCON-92,* San Francisco, CA, Feb. 1992, pp. 383-388.

[2] Dozier, J., "How Sequoia 2000 Addresses Issues in Data and Information Systems for Global Change," Sequoia Technical Report 92/14, 1992. Additional information provided by J. Dozier via e-mail.

[3] U.S. Geological Survey, "The 1991 Conterminous U.S. AVHRR Biweekly Composites," README file from the CD-ROM, EROS Data Center, U.S. Geological Survey, Sioux Falls, SD.

[4] Lillesand, Thomas M., and Ralph W. Kiefer, *Remote Sensing and Image Interpretation,* John Wiley & Sons, Inc.: New York, 1987.

[5] Stonebraker, M., J. Chen, N. Nathan, C. Paxson, and J. Wu, "Tioga: Providing Data Management Support for Scientific Visualization Applications," Sequoia Technical Report 93/29, 1993.

[6] Kochevar, P., Z. Ahmed, and L. Wanger, "An Intelligent Assistant for Creating Data Flow Visualization Networks," Sequoia Technical Report 94/52, March 1994.

[7] Surveys and Resource Mapping Branch, "Spatial Archive and Interchange Format: Formal Definition Release 3.1," Province of British Columbia, Ministry of Environment, Lands and Parks, April 1994.

[8] "Content Standards for Digital Geospatial Metadata," Federal Geographic Data Committee, Washington, D.C., June 8, 1994.

[9] Buehler, Kurt, ed., "The Open Geodata Interoperability Specification," Version 1, October 1994, The Open GIS Foundation, 1 Kendall Square, Suite 2200, Cambridge, MA.

[10] Rosenthal, Arnon, and David Reiner, "Tools and Transformations–Rigorous and Otherwise–for Practical Database Design," *ACM Transactions of Database Systems,* Vol. 19, No. 2, June 1994, pp. 167-211.

[11] Bright, M.W., A.R. Hurson, and S. Pakzad, "Automated Resolution of Semantic Heterogeneity in Multidatabases," *ACM Transactions of Database Systems,* Vol. 19, No. 2, June 1994, pp. 212-253.

[12] Sciore, Edward, Michael Siegel, and Arnon Rosenthal, "Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems", *ACM Transactions of Database Systems,* Vol. 19, No. 2, June 1994, pp. 254-290.

[13] Woodruff, Allison Gyle, and Christian Plaunt, "GIPSY: Georeferenced Information Processing System," Sequoia Technical Report 94/41, March 1994.