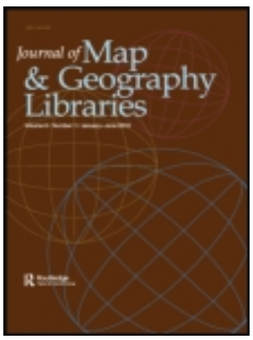

Routledge
Taylor & Francis Group

Journal of Map & Geography Libraries

Advances in Geospatial Information, Collections & Archives

ISSN: 1542-0353 (Print) 1542-0361 (Online) Journal homepage: http://www.tandfonline.com/loi/wmgl20

Embracing the Open-Source Movement for Managing Spatial Data: A Case Study of African Trypanosomiasis in Kenya

Shaun A. Langley & Joseph P. Messina

To cite this article: Shaun A. Langley & Joseph P. Messina (2011) Embracing the Open-Source Movement for Managing Spatial Data: A Case Study of African Trypanosomiasis in Kenya, Journal of Map & Geography Libraries, 7:1, 87-113, DOI: 10.1080/15420353.2011.534693

To link to this article: https://doi.org/10.1080/15420353.2011.534693

Journal of Map And Geography Libraries, 7:87–113, 2011

ISSN: 1542-0353 print / 1542-0361 online
DOI: 10.1080/15420353.2011.534693

Routledge
Taylor & Francis Group

Embracing the Open-Source Movement for Managing Spatial Data: A Case Study of African Trypanosomiasis in Kenya

SHAUN A. LANGLEY and JOSEPH P. MESSINA
Michigan State University, East Lansing, Michigan, USA

The past decade has seen an explosion in the availability of spatial data, not only for researchers, but the public as well. As the quantity of data increases, the ability to effectively navigate and understand the data becomes more challenging. Here we detail a conceptual model for a spatially explicit database management system that addresses the issues raised with the growing data management problem. We demonstrate utility with a case study in disease ecology: to develop a multiscale predictive model of African trypanosomiasis in Kenya. International collaborations and varying technical expertise necessitate a modular open-source software solution. Finally, we address three recurring problems with data management: scalability, reliability, and security.

KEYWORDS disease ecology, open-source, database management, tsetse, African trypanosomiasis

1. INTRODUCTION

The transdisciplinary nature of modern research in disease ecology often requires and generates vast quantities of data varying thematically and in structure. The data management challenge is considerable and often prohibitively costly. Data arise from a multitude of sources and occur in spatially explicit or aspatial forms with concomitant structures. Rarely are these data accompanied by the ontologically coherent metadata necessary to facilitate cooperation and collaboration. Recent discourse in studies of infectious disease ecology have suggested a need to emphasize the role of the changing

This research was supported by The National Institutes of Health, Office of the Director, Roadmap Initiative, and NIGMS: Award RGM084704A.

Address correspondence to Shaun A. Langley, Department of Geography, Michigan State University, East Lansing, MI 48824, USA. E-mail: langleys@msu.edu

dynamics of space and land cover in describing the interactions of diseases with environmental processes (Ostfeld, Keesing, and Eviner 2008). Therefore, effective engagement in disease ecology research requires the ability to correctly access, interpret, and integrate these highly diverse data (Shekhar and Chawla 2003; Watson 2004; Longley 2005).

In the early 1990s there was a great deal of research concerning the expansion of traditional database management systems (DBMSs) to incorporate functionality for handling spatially explicit data types (Stonebraker and Moore 1996; Shekhar and Chawla 2003). Traditional DBMSs are self-describing in that the definitions of data are stored in a catalog, along with the raw data, without the need to store separate descriptive files. This is an extremely efficient means for managing and accessing large quantities of data. Databases improve our abilities to interact with data through the construction and querying of indexes, which serve as a data road map stored on the physical media. Early database frameworks rarely interacted directly with spatial data (Egenhofer 1994; Stonebraker and Moore 1996). Yet with technological advances, particularly the development of satellite platforms for remote sensing, great quantities of data were being generated, necessitating the development of DBMS capabilities specifically to facilitate handling of spatial data.

SQL, standard query language, was formally adopted in 1986 by the American National Standards Institute as SQL-86, and it is the most widely used database-querying language (Lans 2007). The SQL framework provided a logical structure for querying information stored in DBMSs. The first versions of SQL, however, could not explicitly handle spatial data structures. Several extensions of SQL were proposed including GEOQL, SAND, GEO-Kernel, and PSQL; however, the extension that proved most influential—and eventually adopted—was Egenhofer's Spatial SQL (Egenhofer 1994; Adam and Gangopadhyay 1997). Spatial SQL extended the domain to include spatial operators and attributes. Egenhofer further defined the graphical presentation language (GPL), a set of tools in which the results of a spatial query could be manipulated. The Open GIS Consortium (OGIS) promoted spatial functionality by recommending a set of critical reforms in SQL, mainly the adoption of Egenhofer's spatial abstraction model, which introduced Geometry as a base class for spatial objects (Egenhofer 1994; OGIS 1999; Shekhar and Chawla 2003).These recommendations were fully adopted in 1999 with the release of SQL3 (OGIS 1999; Lans 2007).

Data sharing is a critical consideration for our research group, as we collaborate with institutions and researchers throughout much of the United States and East Africa; yet the biggest problem we face as a group is the ability to share data and analysis. We require a new medium that facilitates this flow of information without the need to physically carry the data between institutions. Internet-based GIS and data servers are one solution we employ to allow for simultaneous interaction and analysis of the data. Internet-based GIS emerged with the release of two projects, GeoChange

(Drew and Ying 1996) and the Alexandria Digital Library (Smith and Frew 1995). GeoChange, in particular, set the standard in terms of functionality and usability with interface-supporting, batch-processing, import/export, automatic metadata generation capabilities (Adam and Gangopadhyay 1997). The Alexandria Data Library, though spare in relation to GeoChange, became widely adopted by the University of California system for in-house sharing of proprietary spatial data. Today, despite improvements in functionality, most spatial databases still lack the ability to efficiently handle the raster data structure. Two ongoing projects promise to bring this functionality into spatial DBMSs in 2010. Of particular interest to us is the WKTRaster project, a community supported, open-source project by the Open Source Geospatial Foundation (OSGEO). WKTRaster extends the functionality of the PostGIS library to implement the Raster-type class in the same way Geometry was done in order to support spatially explicit handling of vector data (Stonebraker and Moore 1996). WKTRaster, when fully implemented, will facilitate consistency in data handling of all spatial and nonspatial data types, similar to the manner in which spatial data are managed with the GeoRaster loader included in the Oracle Spatial 11g package.

Although advances in scientific understanding and new technologies have helped to curb the spread of infectious diseases in the developed world, the same cannot be said about the developing world. As the incidence of infectious diseases decreased in the developed world, fewer resources and less attention was devoted to combating those diseases in the developing world, contributing to an increase in the prevalence of many illnesses there (Cohen 2000). In order to develop better strategies for combating disease, we need to enhance our understanding of the underlying ecological conditions that contribute to the emergence of diseases and deliver solutions practicable in developing world contexts (Ostfeld et al. 2008). Disease ecology is generally not considered a discipline in itself but it rather seeks to understand the relationship between disease epidemiology and the landscape (climate, physical, and human) (Sutherst 2004; Keesing et al. 2006; Tatem et al. 2006; Johnson and Thieltges 2010). The transdisciplinary nature of the field creates a unique set of problems, many of which pertain to the use of data while maintaining the rigorous standards mandated by institutional review boards, Health Insurance Portability and Accountability Act (HIPAA), and international research and privacy standards. Very little has been published that directly explores these types of management problems within the disease ecology literature. Routinely, issues of scalability, reliability, and security emerge that hinder the effective dissemination of federally funded data and models. Storage of large quantities of data must at a minimum facilitate the range of applications necessitated by the questions posed in disease ecology. Data *scalability* speaks to the ability of researchers to address questions at multiple scales of spatial or temporal resolution, depending upon the question being asked; the storage of such data must facilitate the rapid, concurrent access and integration of the data across varying

resolutions (Shekhar and Chawla 2003). *Reliability* requires mechanisms to ensure that data mismatches or inappropriate analytical methods are identified or prevented (Shi et al. 2002; Devillers and Jeansoulin 2006). Furthermore, data reliability, particularly with concurrent usage and modification of the data, necessitates mechanisms for ensuring the integrity of the underlying data over time (Shi et al. 2002; Olson 2003). Finally, *security* issues arise when interacting with individually identifiable human data or sensitive community data stored or generated within the DBMS (Olson 2003). Though institutional guidelines and privacy laws may restrict access of the data to preapproved users, the limitations should not preclude nonprivileged users from asking broader questions that might interact with the underlying data when aggregated to remove identifiable data or other information that may be restricted by institutional guidelines or laws (e.g., ethnic identity at low densities in census block group data). Finally, privacy restrictions should be scalable, changing dynamically with the user and scale of resolution requested.

In collaboration with the International Livestock Research Institute in Kenya (ILRI), we have accumulated an extraordinary volume of data for that country. Irrespective of theme, international collaborations often present unique problems in terms of the management, sharing, and dissemination of data necessary to carry out analyses. Our framework for a data management system is a novel solution for spatial modeling in disease ecology, and the use of open-source software makes this an inclusive cost-effective solution for sharing with international collaborators and organizations with limited budgets. The entire suite of data and models is designed to be packaged electronically or on a portable drive to facilitate electronic transfer and physical transportation.

A framework for data management geared to these issues and flexible in the use of restrictions is an ideal solution for working with data types characteristic of research in disease ecology. As part of the National Institutes of Health (NIH) Roadmap program and with the NIH General Medical Sciences support, we are developing a multiscale predictive model that defines the relationship between climate change, land use and land cover change, social systems, and the distribution of tsetse flies and sleeping sickness across Kenya (Messina et al. 2007). Here we present a case study for the implementation of a generalizable disease ecology DBMS framework that provides scalability, reliability, and security to optimize interactions between users and data.

1.1. Case Study: A Model for African Trypanosomiasis in Kenya

African trypanosomiasis (AT), or sleeping sickness, is a major threat to human health across Africa, particularly among impoverished peoples (Brun et al. 2010; Gyapong et al. 2010). Typically considered a disease of the past, its prevalence has increased in recent years, particularly in East Africa, due to

the declining emphasis on trapping and control, climate, and anthropogenic factors (Bauer et al. 1992; WHO 2005; Batchelor et al. 2009). Although monitoring has improved, the extent to which AT impacts East Africa is largely unknown. Recent contributions by foreign countries and aid organizations directed toward addressing AT have declined dramatically in contrast to the increased attention toward AIDS, malaria, and other diseases (WHO 2001; Siringi 2003). The World Health Organization (WHO) has responded by designating AT a neglected tropical disease (Kennedy 2005; WHO 2006; Brun et al. 2010). Not fully understanding the ecological processes that contribute to the spread of AT may result in the inefficient application of control regimes and misallocation of resources, thus retarding the efforts of the African Union to combat and control AT (Cox 2004). As trypanosomiasis has increased in prevalence, the impact on human and animal populations has been considerable, resulting in severe economic hardship for rural families throughout East Africa (Campbell et al. 2000; Campbell et al. 2004).

Tsetse flies (*Glossinidae* family) are the primary vectors for the cyclical transmission of AT. The general distribution of tsetse has been demonstrated in terms of the biophysical extent and the presence of suitable hosts (KETRI 1996; Cecchi et al. 2008). However, the precise limits, historical and contemporary, have not been formalized experimentally (Wint 2001). Furthermore, the current distribution belts reflect outdated data and methodologies (FAO 1979; Wint 2001; Muriuki et al. 2005). Through prior studies, our research group has been able to describe the inaccuracies of these distribution limits, particularly in terms of seasonal changes (DeVisser and Messina 2009; Moore and Messina 2010). Furthermore, global climate change is shifting tsetse habitats, though the degree to which this is occurring is unknown (Sutherst 2004).

To adequately understand the mechanisms behind the increasing incidence of AT, it is important to consider a diverse range of inputs from social, physical, climatic, and even political dimensions. The range of scientific disciplines and methodologies required necessitates an extensive volume of data to be created, collected, and maintained. The management of the volumes and types of data, physically and logistically, has proven to be a significant challenge and the one which we address in this paper. Thus we present a conceptual model for a comprehensive open-source, computing environment that promotes efficient organization, storage, and retrieval of disparate data. Furthermore, we extend the discussion of spatial databases by presenting a model framework for a spatial DBMS that rigorously and consistently manages both spatial and nonspatial data.

1.2. Data Holdings and Acquisitions

We have collected all publicly available data and a significant portion of the known privately held relevant AT disease ecology spatial data for Kenya.

TABLE 1 A Summary of Our Data Library Grouped by Major Theme

Biophysical	Social	Geographical
Precipitation totals	Population density	Land use/Land cover
Monthly temperatures	Population predictions	Land use projections
Evapotranspiration	Historical population	Satellite imagery
Temperature scenarios	Town locations	Lakes
Precipitation scenarios	District census data	Rivers
Historical climate data	Livestock	River basins
Agro-climatic zones	Wildlife	Roads
Agro-ecological zones	Agricultural production	National parks
Lithology	Poverty	Railroads
Soils	Various cadastral data	National boundaries
Landforms	Entomological	Administrative districts
Forest range	Tsetse distribution	Elevation
Wetlands	Tsetse habitat suitability	Topographic maps (countrywide)

The data fall into a classification scheme, defined by topography, soils, vegetation, climate, ecological diversity, water resources, and anthropogenic factors known to control or influence the ecological processes driving tsetse distributions over time and space (see Table 1 for a summary). Nonspatial data consist primarily of governmental and intra-agency reports obtained through private libraries in Kenya. The reports collected focus on policies and governmental and community control and eradication programs. Currently, these reports are neither catalogued nor indexed, limiting efficient use of any information they may contain.

Remotely sensed image data comprise the majority by physical file size of our data collection. We possess most of the known publically available aerial imagery for Kenya from the Landsat, MODIS, PALSAR, and ASTER platforms; including a number of image products summarizing land use classification (MODIS types 1–5 for 2001 to 2005 and Landsat MSS– derived 1 km for 1980), land surface temperature (MODIS LST), precipitation (WorldClim 30-y average at 1 km), and vegetation indexes[1] (NDVI for 2001 to 2008 every 16 d, 250-m resolution). Additional sources of land use and land cover information are provided with Africover as vector or raster data types, GLC2000, CLIP cover,[1] and UMD Global Land Cover. We have elevation data from ASTER, SRTM, and digitized topographic maps (30-m, 90-m, and 250-m spatial resolutions, respectively). Regarding the distribution of tsetse, we possess vector and raster digitized distributions of fly belts for Kenya for 1967, 1973, 1996, and 2000.[2] Finally, there are rasterized estimates of livestock densities (#/km^2) for 2007 (ILRI[3]). Utilizing these data, DeVisser et al. (2010) developed the TED model to predict the distribution of tsetse in Kenya (see Figure 1). To effectively and efficiently recover and maintain the value of the data, we require a solution that not only provides for efficient storage and retrieval of the data, but that also allows for automated metadata generation.

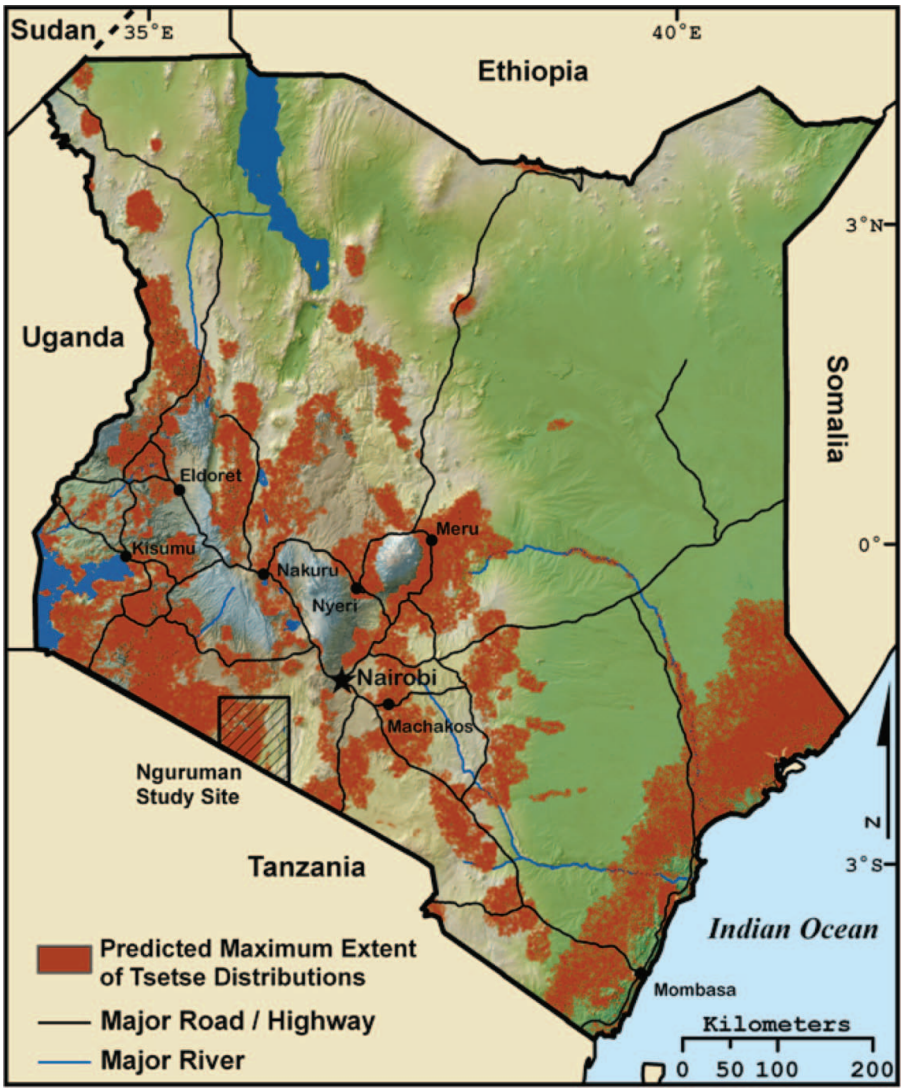

FIGURE 1 Maximum extent of tsetse distribution predicted by the TED Model between the beginning of 2002 and the end of 2009. As described by DeVisser et al. (2010), the TED Model uses five scenes of MODIS 1km annual land cover, 207 scenes of MODIS 250 m Normalized Difference Vegetation Index (NDVI), and 207 scenes of MODIS 1km day/night Land Surface Temperature (LST) products to predict the fundamental niche of tsetse in Kenya, and a fly movement model to predict tsetse distributions or realized niche of the tsetse species of interest. The TED Model is written in Python scripting language and is run within ArcGIS 9.2 (or later versions of ArcGIS). Other data sets used to construct the map include shapefiles of Kenyan major roads, highways, cities, rivers, Kenyan water bodies, lakes, and African country political boundaries. To construct the background topographic relief map, the Shuttle RADAR Topographic Mission (SRTM) 90 m Digital Elevation Model (DEM) was used to create a gridded hillshade product, and a dry season NDVI scene was combined with the SRTM DEM to create an elevation/vegetation color scheme.

We have 103 spatial data sets of socioeconomic and demographic assessments of Kenya between 1971 and 2008. A portion of these data were provided by the Integrated Public Use Microdata Series (IPUMS) International data set for Kenya,[4] and they originate from the Kenya 1989 and 1999 census collected by the Kenya National Bureau of Statistics. The IPUMS data are a systematic sample of every twentieth household; these data represent a sampling fraction of 5 percent and expansion factor of 20. A long-form questionnaire was implemented to survey individuals within households. Location data for each individual are limited to the respondent's province and district, first and second administrative levels respectively, of five possible levels each at an increasingly finer spatial scale of resolution. Data at finer spatial scales are not available as part of the Kenya Bureau of Statistics' effort to maintain privacy. A further complication was a change in the number of districts in Kenya from forty-two in 1989 to sixty-nine in 1999; however, this change was made by subdividing existing districts allowing rough comparisons between associated regions. The IPUMS sample provides ninety-seven household and individual variables, most importantly geographic information (urban/rural status, province, district), utility (electricity, water supply, sewage type, and type of cooking fuel), and dwelling (number of rooms, toilet type, floor, wall, and roof material). Other relevant data include individual variables describing household position, demographic characteristics, education, employment, migration, and disability (see Table 2 for a sample). There are a total of 1,074,048 individual entries for the 1989 census and 1,407,597 for the 1999 census samples.

Decisions on data management often conflict during collaborative research, resulting in the lack of a cohesive strategy for data management and the inability to share such data effectively. Although our colleagues in Kenya have the technological skills to work with spatial data, the telecommunication network is insufficient to provide the necessary bandwidth or reliability to acquire the data via direct transfer over FTP or other similar protocol. To overcome this problem in the short term, it is necessary for us to carry the data into the country on physical media and to have it accompanied by a data management system that can facilitate interaction with the large quantity of data. Thus, efficiency and portability are significant concerns.

2. DEVELOPMENT OF A SPATIAL DATABASE SYSTEM

Our solution for a spatial DBMS involves bridging a variety of software packages following the basic framework as described by Câmara et al (1996) for the development of the TerraLib GIS Library and the integration considerations posed for the MurMur project (Parent, Spaccapietra, and Zimányi 2006). First, we outline the conceptual framework, and second, we implement the design. Third, we describe the development of routines, batch, or other

TABLE 2 A Sample of the Subset of Kenya Census Data we Hold from the 1990 National Census

	cntry	year	sample	serial	persons	wthh	subsamp	gq	unrel	ubran	provke	distke	ownrshpd	electrc
1	404	1989	4041	1000	4	20.0000	26	10	0	2	1	1010	216	2
2	404	1989	4041	1000	4	20.0000	26	10	0	2	1	1010	216	2
3	404	1989	4041	1000	4	20.0000	26	10	0	2	1	1010	216	2
4	404	1989	4041	1000	4	20.0000	26	10	0	2	1	1010	216	2
5	404	1989	4041	2000	1	20.0000	76	10	0	2	1	1010	216	1
6	404	1989	4041	3000	4	20.0000	2	10	0	2	1	1010	216	1
7	404	1989	4041	3000	4	20.0000	2	10	0	2	1	1010	216	1
8	404	1989	4041	3000	4	20.0000	2	10	0	2	1	1010	216	1
9	404	1989	4041	3000	4	20.0000	2	10	0	2	1	1010	216	1
10	404	1989	4041	4000	1	20.0000	92	10	0	2	1	1010	140	2
11	404	1989	4041	5000	1	20.0000	81	10	0	2	1	1010	216	1
12	404	1989	4041	6000	12	20.0000	5	10	0	2	1	1010	216	1
13	404	1989	4041	6000	12	20.0000	5	10	0	2	1	1010	216	1
14	404	1989	4041	6000	12	20.0000	5	10	0	2	1	1010	216	1

preconfigured shell scripts that can be selected to run either from the command line or through an interactive GUI prompt whereby a user can add and recall raw data files or query the database to return a mash-up of spatial data files or metadata. Finally, we develop a set of SQL triggers (code set to run when activated by a defined action) to enforce data integrity.

2.1. Conceptual Model

Figure 2 demonstrates our conceptual model for a spatial DBMS. In contrast to previous implementations of MySQL, Postgres, and other common spatial databases, modern DMBS models facilitate raw data and metadata to be stored together, embedded in the database (Elmasri and Navathe 2004; Watson 2004). The proposed spatial computing environment uses open-source, community-supported software and standards, providing a solution to the data-management problem that is temporally extensible. The database is portable in that we can copy the database and software binaries to a portable drive that can be carried to Kenya. Of critical concern to us in selecting software components was the interoperability of the system, and the ability of components to interface and work together. Our selected DBMS is PostgreSQL[5] (Postgres), an advanced, readily available, open-source, object-relational database management system. Using standard SQL syntax, Postgres allows for complex query capabilities, including spatial queries, and facilitates strict rule and primary key enforcement. Postgres is also extensible, allowing for the addition of new functionality (Stonebraker and Rowe 1986; Stonebraker and Kemnitz 1991).

PostGIS[6] is an extension to the Postgres language that adds functionality for the storage and retrieval of spatial data files. PostGIS is, at its core, a suite of tools that serves as the back end for spatial functionality in Postgres (Ramsey 2005). Of particular interest is the WKTRaster[7] (Beta 0.1.6) project, which extends the ability of a Postgres database to store and index raster data, a first of its kind. The project mirrors the inherent vector-based functions (of Geometry type) for raster data. The result is a single set of SQL functions that handle both spatial data types. This extension has the potential to greatly enhance and facilitate the utilization of raster data by end users. Figure 3 presents the conceptual model for the user interface to the Postgres DBMS. We incorporate a variety of software packages, explained later, each of which provides the user with statistical, visual, and geoprocessing capabilities; the user can interact with these packages through a GUI or through a command line interface.

GRASS (geographic resources analysis support system)[8] is one of the few open-source options for handling raster GIS data. Developed by the Open Source Geospatial Foundation, GRASS is an increasingly popular solution for the use and analysis of spatial data in academic research. Like the many other

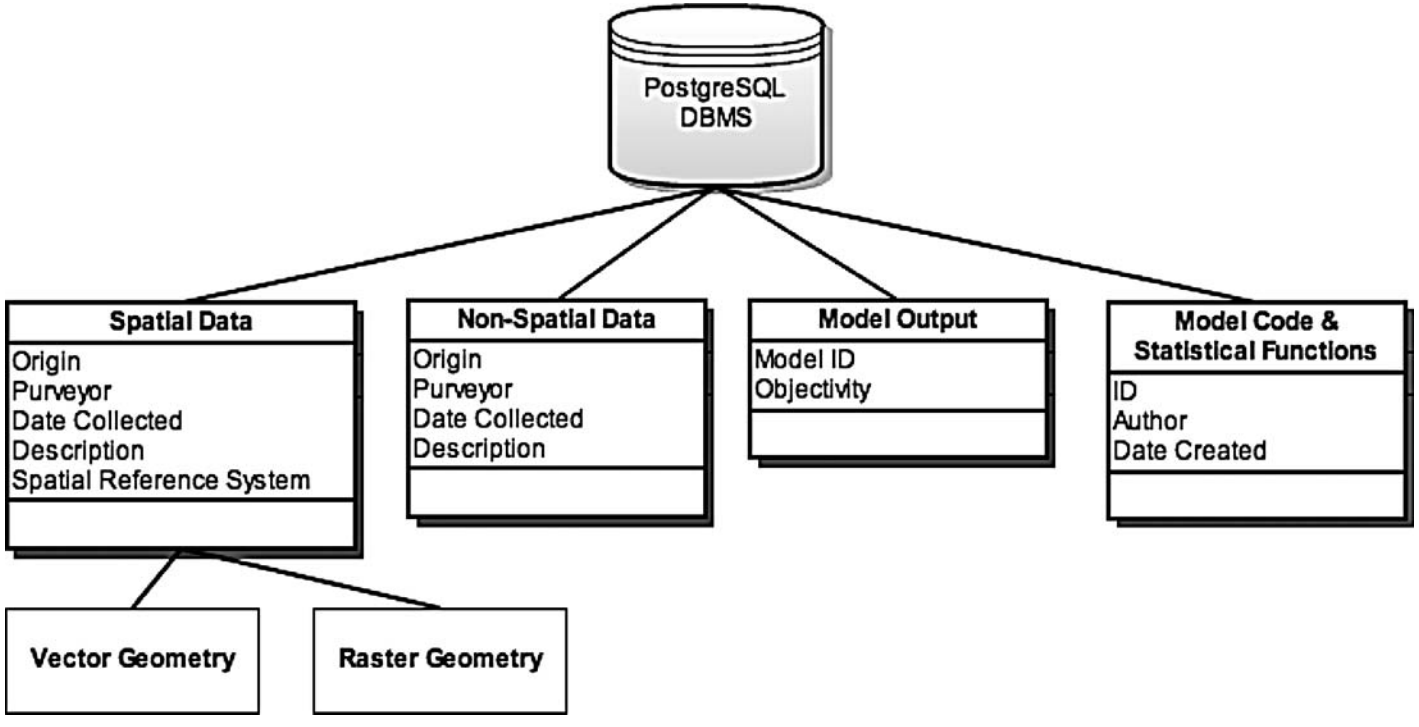

FIGURE 2 A conceptual model framework for the spatial DBMS.

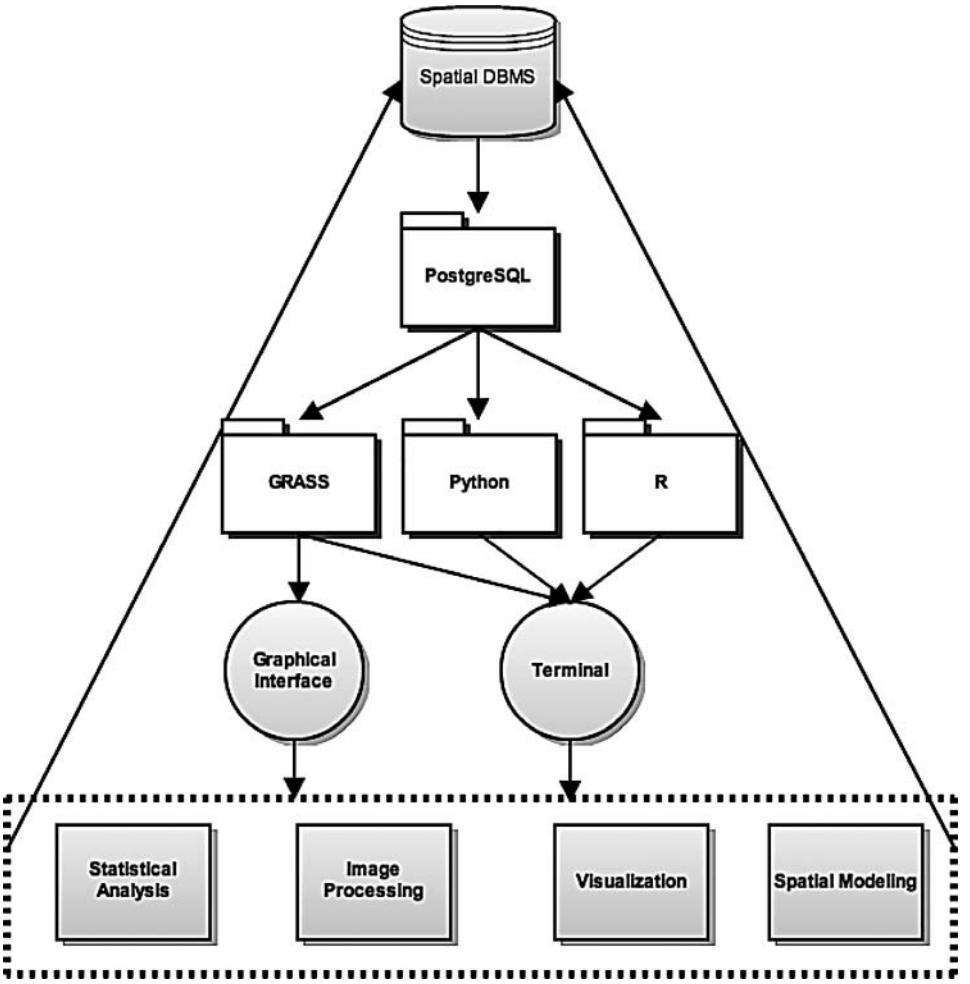

FIGURE 3 Flow of data through the model implementation as users interact with the system. It should be noted that flows are one-directional, meaning that although users can interact with the data to generate analysis, the results must be stored as a separate entity in the database. This ensures that the underlying data cannot be changed.

components selected for the system presented here, GRASS is interoperable with Python and Postgres, and is extensible, allowing users to create and add new functionality.

Although there is a plethora of statistical packages available, R is our preferred statistical package,[9] in large part because it interfaces easily with GRASS and Postgres. We can do so without needing to install any additional software components. R is an open-source solution, developed by an international community of users, to create an alternative to the often expensive and restrictive programs offered by statistical companies. Although R lacks a graphical interface, it uses far less computer memory than most other

comparable statistical packages. This allows us to perform complex analyses with fewer hardware demands, an important consideration for maintaining portability of the project.

Finally, Python is an object-oriented programming language developed by Guido van Rossum in 1991 and maintained, in large part, by the open-source community (Python Software Foundation 2010). Python is an efficient scripting language that facilitates interoperability between our database (Postgres), statistical analysis software (R), and GIS (GRASS) (Neteler and Mitasova 2008). Python is a highly intuitive, object-oriented programming language, easy enough to learn and use that it makes for a good solution to bridge our project components. Furthermore, the open-source nature of the language fits well with the other components, enabling us to incorporate additional extensions written by the community.

The most critical considerations for long-term data storage are persistence and security (Elmasri and Navathe 2004; Watson 2004). Digital data are inherently ephemeral in that over time physical storage media will fail or degrade, thus requiring continuous rewriting on digital media to ensure persistence. Though drive technology has progressed significantly in the past decade, drive failure is not uncommon; the volume of data and frequency with which the data are accessed puts immense stress on the physical mechanisms. Therefore, it is necessary to employ a strategy that ensures the long-term viability of the data. To this extent, we employ a redundant array of independent disks (RAID) as our secondary storage medium, mirroring data between groups of drives. This enables corrupted data resulting from disk failure to be recovered in real time, without the need to create tertiary backup regimes. Furthermore, we enforce constraints to access of the data in making the data read-only, reducing the chance a write error will occur or values will be inadvertently changed. Finally, all files will have MD5 checksums (a cryptographic hash code generated from a file's binary data) included as an attribute of the file allowing us to verify the integrity of any data file (Rivest 1992). The strategies employed here will also enable us to protect against accidental user error, which may result in inadvertent modification or deletion of data. Security restrictions, though a good first line of defense, are frequently circumventable. The ability to roll back changes within Postgres, as well as being able to restore data from the RAID, will ensure the long-term integrity of the data library.

2.2. Database Standards

Generic frameworks for database development, such as the one we outline here, should use established ontologies in their component descriptions, which we satisfy by conforming to the formal ontology described by the Open Geospatial Consortium (OGC).[10] The OGC is an independent group

tasked with the purpose of developing and maintaining a set of standards for the management of spatial data to promote both consistency and interoperability across GIS platforms. The Geographic Data Abstraction Library[11] (GDAL) stands as a single standard for interoperability of raster data within the GIS community. As a library, it facilitates the conversion between data products. However, as is the case when working with proprietary data, conversion between formats requires a commonly understood intermediate. The GDAL standards and abstraction libraries used in our project facilitate this conversion between data formats by providing a commonly understood intermediate.[12] Custom GDAL libraries are used largely in our implementation of GRASS as a mechanism to add support for a range of occasionally unsupported data formats.

There are an inordinate number of disparate standards for metadata generation and inclusion, none of which are remotely universal. Regardless, it is necessary for our own data management that we accept and enforce a single standard for metadata management. The fact that our DBMS facilitates the storage of raw data within the system precludes the need to store metadata separately as this information is included within the database as discrete variables (the raw data are technically also included as an attribute value). However, it is foreseeable that we may need, on occasion, to transfer data outside the realm of the database. Under this scenario, it is necessary to have a mechanism to recreate the metadata files discarded earlier. Thus, we incorporate custom scripts that can be called to assemble the necessary metadata precursor information from the data table. The resulting metadata report meets the formatting and content criteria specified by the OGC and ISO19115 specifications.[13]

2.3. SQL-Rule-Based Interactions and Scripting

In an effort to ensure the integrity of the raw data and prevent accidental deletion or modification, all users are restricted in the ways they can interact with the DBMS. In most cases, raw data are restricted to read-only access by users at the database level. One way we achieve this is by restricting Update and Delete privileges to the database administrator account only. This ensures that raw data cannot be changed or deleted through unverified scripting. It further facilitates the simultaneous access and use of raw data by multiple processes. Figure 3 symbolizes the flow of data from the DBMS to the user. Even though the raw data are read-only, it is useful to permit users to save the output of models to the database, with the option to link to the source data. Indexing these data sets together is useful when new users explore the data. We will discuss this functionality further in the section “A User Perspective: Interfacing with the DBMS.” Metadata for these model outputs will vary, but will at a minimum include the user identity and

model code, as well as a summary of the model objective. A set of predefined rules or triggers is loaded into the DBMS, providing enforcement of the desired constraints (e.g., spatial/temporal mismatch encountered during scripted analysis). These rules serve to provide a minimal standard of validity and consistency for model output and statistical analysis (Shi, Goodchild, and Fisher, 2002; Devillers and Jeansoulin 2006).

The storage of raw data within the DBMS precludes the need to regularly interact with different data formats. Nevertheless, it may be useful to incorporate a mechanism whereby we can convert data among data formats or export data into a range of formats. Although most of these tasks can be accomplished within the GRASS interface, we include in our library a set of Python scripts to extend our ability to move among data formats (particularly useful for the range of raster imagery available). For example, our "data bank" includes raster imagery (*.img, .tif, .hdf, among others), vector data (*.shp and others), text documents (*.doc, .docx, .txt, .pdf, and others), metadata files corresponding to acquired imagery (*.xml, .pdf, .txt, and others), as well as an assortment of other types not specifically mentioned here.

2.4. DBMS Interface—A Manager Perspective

Implicit in the design of our database are SQL rules (and triggers) that constrain the ways users can work with data in an effort to prevent common mistakes. Because the database holds data at varying resolutions and extents, we constructed rules to check for common errors committed by users in selecting data layers. In the event that data layers are deemed incompatible, the user is alerted to the mismatch and encouraged, though not required, to restate their request. These rule sets operate at the point of data retrieval and storage. When importing data, we check and request that the user define the relationship between spatial data and metadata. If metadata are not available, the user is prompted to input known characteristics of the data file in an attempt to force this paired relation. Common examples of prompted information include the time stamp of data acquisition or generation, solar or view angle (for satellite-derived imagery), or a definition of codes that may occur within the data table (often the case with census data). Some of these data can be retrieved from the raw data, such as extent, summary statistics, or file type. Enforcing these relationships at the time of storage will greatly reduce the number of unpaired spatial data files and corresponding descriptors.

These rule sets also operate on a variety of demographic or other similar data types stored in our database. In our specific implementation of the database, we have sample volumes of census data collected by the Kenyan government and acquired from IPUMS.[14] As data are input into the database,

users are prompted as to the type of data being input and plain English definitions of the variables (a long form description of the purpose of the data) included in the data set. We developed a means to query the data file for variables and return this list in memory to the rule set. In the event the data file does not return the correct list of variables, the user is prompted to specify the range of variables or create them.

2.5. DBMS Interface—A User Perspective

Though many incantations of interfaces are possible, one means by which users can interact with data is through a Web browser, which connects to the database via an application developed for Mac OS. Through the application, users have access to all the tools needed to query, utilize, and analyze data from the DBMS. From a set of menus, users are prompted to select a subset of data. Next, they are given the opportunity of selecting from either a set of precompiled models or statistical scripts that can analyze the data, or the data can be brought forward and immediately visualized in the browser window. Finally, the users are provided with command line tools that can augment their interaction with the data. This approach to data interaction lowers the learning curve for new users and gets them instantly connected with the data.

2.6. Statistical and Analytical Analysis

We extend the functionality of our DBMS to assist users in browsing the library of data by using the R package to automatically calculate descriptive statistics. These summaries are potentially most valuable to users not involved with the production of the data, or who may not be familiar with a region of interest. Furthermore, providing a means to compute descriptive statistics automatically enforces consistency between files that is often a symptom of user error when files are independently managed.

Perhaps the most common challenge users face in interacting with databases is retrieving data both correctly formatted and appropriate for use in a particular analysis (Longley 2005). The scripted analysis tools help bridge the gap in understanding for users not familiar with the R statistical package. With a simple menu prompt, the user can specify which data files and statistical methods are to be employed by the program. Though not a comprehensive selection, it provides a means to conduct a range of simple statistical comparisons. Further analyses can be performed though the R package directly linked to the database.

3. IMPLEMENTATION

To enable the reader to implement our DBMS model, we provide a general outline of the steps needed to install and configure the software packages. We make no assumptions as to the hardware on which the model is implemented. Our solution requires a number of software components, including PostgreSQL, PostGIS (with GDAL and WKTRaster extensions), Python, GRASS, and the R statistical package. In this section we will detail the required steps for the implementation of each component within open-source Debian distributions of Linux.

3.1. Software Packages

Debian distributions of Linux take advantage of the aptitude system for software distribution, and therefore you can install all packages from the terminal. To install, type

```
sudo apt-get install python postgresql
sudo apt-get install gdal-bin postgresql-8.4-postgis postgis grass r-base
```

You may also be interested in installing the Quantum GIS (QGIS) program, another convenient GUI that interfaces with GRASS, which can also be installed with aptitude:

```
sudo apt-get install qgis
```

To allow interaction between GRASS and the GDAL libraries, you need to build and configure the GRASS plugin for GDAL. Download and follow the installation instructions provided by

```
http : //trac.osgeo.org/gdal/wiki/GRASS
```

Finally, you need to install optional extensions to the R package. Start R from a terminal by typing R; then install packages with

```
install.packages ("ctv")
library (ctv)
install.views ("Spatial")
```

3.2. Initializing Postgres

You will need to log in to Postgres for the first time using the default "postgres" login. From the terminal window you will then be able to create user accounts, set access restrictions, and create the database for the project data. First start Postgres from a terminal by typing

```
psql -U postgres
```

You can now create your user accounts. In our implementation of the project, all users are granted limited permission by default. Additional rights can be given later depending on the needs of the user.

```
CREATE USER <username>;
CREATEDB <dbname>;
GRANT ALL to <username> ON <dbname>;
```

It is necessary to configure the database to enable spatial functionality with PostGIS. The instructions given are also available from the PostGIS documentation at

```
http : //postgis.refractions.net/documentation/
```

First, type

```
CREATELANG plpgsql <dbname>
```

Now, copy the PostGIS spatial definition files into your new database and navigate to the PostGIS installation directory specified during installation and type

```
psql – d <dbname> -f postgis.sql
\q #Quits the Postgres session
```

For additional security options available, you can refer to the Postgres documentation.[15] You can now log in to your new database from a terminal window by typing

```
psql -U <username><dbname>
```

Use caution when performing upgrades to PostGIS, as you will likely have to rebuild support for your database.

3.3. WKT Raster Extension for PostGIS

WKT Raster is not yet included in the PostGIS precompiled binary as it is (as of 09/15/2010) still in beta testing. Therefore, you will need to download and compile the extension. The following instructions are available from the WKT Raster project documentation.[16] Depending on your specific system configuration, it may be necessary to build the PostGIS libraries from source as well as the required dependencies. Download the source code from

http : //www.postgis.org/download/

Unpack the tar file and navigate to the wktraster directory. Generate a configuration profile by typing

./autogen.sh

Now run a configuration script. You will need to locate the installation directory for PostGIS:

./configure - -with-postgis-sources=/thesrc/postgis-version

If no errors were generated, you can install wktraster. Because you are already in the installation directory, simply type

make & make install

3.4. Configuring GRASS

The first time you start GRASS, you will have to specify a location and path for your data. Choose the data path that you set up earlier (it should already be the default entry). To create a new location, it is easiest to have a georeferenced data file that spans the region of interest. Although not necessarily required by the software, it greatly simplifies the process of adding a spatial reference system to the data library to specify it right away. Start GRASS by navigating to the icon or from a terminal:

grass64 - gui

Click on the Location Wizard icon on the right-hand side of the window. Follow the on-screen instructions, selecting your georeferenced data file when prompted. Alternatively, you can create a new location by entering GRASS with the default example "spearfish60." From the GRASS terminal navigate to the directory where your data are stored and type

r.in.gdal i=<yourimage> -o <mapname> location=<newlocation>

The function r.in.gdal will parse the image file and automatically define the region based on the extent of the selected image.

3.5. Loading Data and Interfacing GRASS with Postgres

There are a great many possibilities for importing data and interfacing with GRASS. Here we demonstrate several examples of importing data into Postgres and accessing those data within GRASS. Landsat7 data are downloadable from the USGS as georeferenced TIFF files for individual bands. To import them into a sample database Kenya, use a Python loader script:

```
gdal2wktraster.py -r * .tif -t Landsat -s 4326 -k 100 × 100 -I >
    Landsatloader.sql
```

The script does not directly load the data into the database; rather it creates the necessary SQL script to do so. In this example, the -r option enables multiple files (selected with the asterisk) to be imported simultaneously into a single table. The -t option specifies the table name the data will be imported to. With the -s option, we specify the spatial reference system using SRID numbers (spatial reference identifier, OGC specifications), WGS84 in this example (Herring 2006). The –k option splits each raster into tiles that are 100 × 100 pixels. Finally, the –I option requests that a spatial index file be created for each raster tile. Next, pass the SQL loader for processing with

```
psql -U  <username>  -f Landsatloader.sql  <yourdatabase>
```

If you do not want to create a spatial index or you forgot to do so in the first step, you can easily create one at the Postgres prompt:

```
CREATE INDEX Landsat_SI ON Landsat USING GIST
    (ST_ConvexHull (rast));
```

In this example, the GIST (generalized search trees) index is used, which has a balanced tree index structure similar to a B-tree (Hellerstein 1999; Kornacker 1999). An example of a common format vector data type is the ESRI shape file and an example is a digitized map of fly belt regions FLY.shp. Postgres facilitates the importing of vector data from the terminal with

```
shp2pgsql -s 4326 -I -D FLY.shp  <yourdatabase> .flybelts >  flybelts.sql
```

Here the –s flag specifies the spatial reference system using SRID codes. The –I option requests the script initialize a GIST spatial index on the geometry column of the data. Finally, the –D option creates a dump file (sql loader) that can be imported into Postgres from the terminal, a faster means of

adding data to the database. Now pass the sql loader to Postgres with

```
psql -U <username>  -f flybelts.sql  <yourdatabase>
```

Alternatively, import vector data using a graphical interface by loading

```
shp2pgsql -gui
```

To load the data into GRASS, initialize the Postgres driver from within the GRASS terminal. Start GRASS specifying a work location (see previous instructions for the generation of Location). Now, load the driver defining the connection between Postgres and GRASS:

```
db.connect driver=pg database="host=localhost,
  dbname=<yourdatabase>"
                db.login user=<username>
                db.connect -p
                db.tables -p
```

After initializing the connection, you can query the database, for example, retrieving the fly belts data described earlier:

```
v.in.ogr dsn="PG : host=localhost dbname=<yourdatabase>
  user=<username>" layer=???? output=flybelts type=boundary,
  centroid
                v.db.select flybelts
                v.info -t flybelts
                d.vect flybelts
```

Data in HDF formats can be read only with additional GDAL libraries, which are not included with the standard distribution. However, the version of GDAL made available through the Ubuntu repositories appears to support some limited functionality. If further interaction with HDF is required, you will need to compile GDAL manually, inputting development files downloadable from the HDF Group.[17] HDF formats are containers, and thus may hold multiple data sets. Header data are accessed with

```
gdalinfo sample.hdf
```

Subdata set names are formatted as

HDF4_SDS:subdataset_type:file_name:subdataset_index

A portion of the output reads:

SUBDATASET_8_NAME=HDF4_SDS:MODIS_L1B:GSUB1.A2001124.0855.003.200219309451.hdf:7 SUBDATASET _8_DESC=[408×271] Range (16-bit unsigned integer)

Detailed headers for this subdata set can be viewed with

```
gdalinfo\SUBDATASET
  _8_NAME=HDF4_SDS: MODIS_L1B: GSUB1.A2001124.0855.003.
  200219309451.hdf : 7
```

Inasmuch as the GRASS plugin was compiled with HDF support, image data, by individual band, are directly imported into GRASS with r.in.gdal.

r.in.gdal HDF4_SDS:MODIS_L1B:GSUB1.A2001124.0855.003.200219309451.hdf:7 out=hdfexample

Once the data are loaded into the GRASS interface, they can be imported into Postgres either directly with db.connect or by exporting the image as .TIF and importing with the gdal2wktraster.py script.

4. LIMITATIONS AND FUTURE EXPANSION

The initial phase of our project is limited to the objectives addressing the misuse, misrepresentation, and effective archiving of our data library. As advances and improvements are made to the telecommunications infrastructure in Kenya, we will be able to share a common, easily-accessible repository for data with our colleagues outside the United States. The DBMS framework, including necessary software packages and data, can be packaged together and distributed via portable hard drive. Future updates to our software model will include a Web-based interface that will allow users to interact with the DBMS, including the suite of analysis and visualization tools (R and GRASS), without the need to install and configure these programs locally. This reduces the hardware requirements for working with the data, allowing a potentially broader base of users to share in data access. This approach has been extensively applied in many of the institutional projects commonly referenced in the literature (Câmara et al. 1996; Parent et al. 2006).

5. SUMMARY

Disease ecology is a transdisciplinary field, exploring the complex interactions between diseases and the environment. Computational and collaborative barriers inhibit meaningful advances in the field. Major problems include data management schemas to facilitate the scalability (data are resampled dynamically to avoid redundant storage), reliability (concurrent access to data permitted while ensuring that the raw data cannot be changed), and data security (the database allows for a dynamic security access policy while meeting HIPAA and Institutional Review Board [IRB] requirements). As data become aggregated with decreasing spatial resolution, many of the privacy concerns disappear but tracking and management problems proliferate. The DBMS must dynamically alter the restriction rule-set to account for aggregation and application challenges. Previous implementation of data management systems required that multiple instances of the data be stored, creating a problem of exponentially increasing data storage demands. Currently, no framework for data management that addresses this set of integrated concerns exists.

Over the course of our research on AT, we have accumulated a large library of data. Our database model utilizes open-source software to allow for flexibility and extendibility in model implementation. We take advantage of the new PostGIS extension, WKTRaster, to allow for the storage of raster imagery within the database. This allows us to enforce a single standard in the way all data formats, irrespective of the contents, are managed. With this development, we are finally able to store our entire data library, spatial and nonspatial, explicitly within a single database implementation, and the restrictions and rule sets coded into the DBMS should ensure the long-term integrity and security of the data.

The overarching goal of this project is to create a multiscale predictive model for the tsetse and AT to provide a means whereby governments, communities, and Nongovernmental Organizations (NGOs) can make informed decisions for disease control or suppression that are spatially and temporally aware. Given the variable background and technical expertise of the different groups, our solution should be simple enough for the most basic user, yet powerful enough to be useful for complex analyses by the most skilled. To maximize the utility of this system, we will use a participatory design framework to develop Mac, iPhone, and Web applications for interfacing with the models and data.

Open-source software is uniquely capable of rapid adoption of new technologies and functionality due to the base of community developers working on modular extensions to the software base. Spatial databases are increasingly common. Mobile data applications are becoming increasingly location aware (e.g., Facebook, Twitter, Loopt, Turn-by-turn Navigation [Waze]),

with the purpose of providing users with context-specific information and opportunities. These GIS technologies increasingly promote Web-based interfaces to a spatial database. Perhaps the most exciting vision for geographic information systems (GIS) is the adoption of mobile technologies, now incorporating global positioning systems (GPS) technology, to provide for context-aware interaction with spatial database systems. The next step in our project is to implement an efficient, two-way Web portal for the spatial DBMS that will allow us to interact with data and model results from mobile devices in the field. Combined with automated scripting and model execution, this would have the potential to dramatically increase the amount of information shareable with local communities. Although not yet possible due to barriers in Kenya, we work toward the vision of achieving synchrony between science and practice.

NOTES

1. Data are publicly available.
2. Not all maps are publicly available.
3. International Livestock Research Institute (Nairobi, Kenya).
4. https://international.ipums.org/international/
5. http://www.postgresql.org
6. http://postgis.refractions.net
7. http://trac.osgeo.org/postgis/wiki/WKTRaster
8. http://grass.itc.it
9. http://cran.r-project.org
10. http://www.opengeospatial.org
11. http://www.gdal.org
12. http://www.opengeospatial.org/standards/is
13. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26020
14. http://international.ipums.org/international/
15. http://www.postgresql.org/docs/
16. http://trac.osgeo.org/postgis/wiki/WKTRaster/Documentation01
17. http://www.hdfgroup.org

REFERENCES

Adam, N. R., and A. Gangopadhyay. 1997. *Database Issues in Geographic Information Systems: The Kluwer International Series on Advances in Database Systems*. Boston: Kluwer Academic Publishers.

Batchelor, N. A., P. M. Atkinson, P. W. Gething, K. Picozzi, E. M. Fevre, A. S. L. Kakembo, and S. C. Welburn. 2009. Spatial predictions of Rhodesian human African trypanosomiasis (sleeping sickness) prevalence in Kaberamaido and Dokolo, two newly affected districts of Uganda. *Plos Neglect Trop D* 3(12):e563.

Bauer, B., I. Kabore, A. Liebish, F. Meyer, and J. Petrich-Bauer. 1992. Simultaneous control of ticks and tsetse flies in Satiri, Burkina Faso, by the use of flumethrin pour on cattle. *Tropical Medicine and Parasitology* 1(43):1–74.

Brun, R., J. Blum, F. Chappuis, and C. Burri. 2010. Human African trypanosomiasis. *The Lancet* 375(9709):148–159.

Câmara, G., R. C. M. Souza, U. M. Freitas, and J. Garrido. 1996. SPRING: Integrating remote sensing and GIS by object-oriented data modelling. *Comput & Graphics* 20(3):395–403.

Campbell, D., H. Gichohi, A. Mwangi, and L. Chege. 2000. Land use conflict in Kajiado District, Kenya. *Land Use Policy* 17(4):337–348.

Campbell, D., D. P. Lusch, T. A. Smucker, and E. E. Wangui. 2004. Root causes of land use change in the Loitokitok Area, Kajiado District, Kenya. *Land Use Change Impacts and Dynamics (LUCID) Project Working Paper* 19.

Cecchi, G., R. C. Mattioli, J. Slingenbergh, and S. De La Rocque. 2008. Land cover and tsetse fly distributions in sub-Saharan Africa. *Medical and Veterinary Entomology* 22(4):364–373.

Cohen, M. L. 2000. Changing patterns of infectious disease. *Nature* 406(6797): 762–767.

Cox, F. E. G. 2004. History of sleeping sickness (African trypanosomiasis). *Infectious Disease Clinics of North America* 18(2):231–245.

Devillers, R., and R. Jeansoulin. 2006. *Fundamentals of Spatial Data Quality*. Newport Beach, CA: ISTE.

DeVisser, M., and J. Messina. 2009. Optimum land cover products for use in a *Glossina morsitans* habitat model of Kenya. *International Journal of Health Geographics* 8:39–39.

DeVisser, M. H., J. P. Messina, N. J. Moore, D. P. Lusch, and J. Maitima. 2010. A dynamic species distribution model of *Glossina* subgenus *morsitans*: The identification of tsetse reservoirs and refugia. *Ecosphere* 1(1):1–21.

Drew, P., and J. Ying. 1996. GeoChange: An Experiment in Wide-Area Database Services for Geographic Information Exchange. In *Proceedings of the 3rd International Forum on Research and Technology Advances in Digital Libraries*: IEEE Computer Society.

Egenhofer, M. 1994. Spatial SQL: A query and presentation language. *IEEE Transactions on Knowledge and Data Enineering* 6(1):86–95.

Elmasri, R. M., and S. Navathe. 2004. *Fundamentals of Database Systems*. 4th ed. Boston: Pearson/Addison Wesley.

FAO. 1979. *The African Trypanosomiases: Report of a Joint WHO Expert Committee and FAO Expert Consultation*. Rome: FAO.

Gyapong, J. O, M. Gyapong, N. Yellu, K. Anakwah, G. Amofah, M. Bockarie, and S. Adjei. 2010. Integration of control of neglected tropical diseases into health-care systems: challenges and opportunities. *The Lancet* 375(9709):160–165.

Hellerstein, J. 1999. *The GiST Indexing Project*. Available at http://gist.cs.berkeley.edu

Herring, J. R. 2006. OpenGIS Implementation Specification for Geographic Information—Simple Feature Access. Part 2: SQL option. Open Geospatial Consortium Inc.

Johnson, P. T. J., and D.W. Thieltges. 2010. Diversity, decoys and the dilution effect: How ecological communities affect disease risk. *J Exp Biol* 213(6):961–70.

Keesing, F., R. D. Holt, and R. S. Ostfeld. 2006. Effects of species diversity on disease risk. *Ecology Letters* 9(4):485–498.

Kennedy, P. 2005. Sleeping sickness—human African trypanosomiasis. *British Medical Journal* 5 (5):260–267.

KETRI. 1996. *Tsetse Distribution in Kenya Showing Tsetse Belts and Conservation Areas*.

Kornacker, M.. 1999. High-Performance Extensible Indexing. Paper read at 25th VLDB Conference, at Edinburgh, Scotland.

Longley, P. 2005. *Geographical Information Systems and Science*. 2nd ed. Hoboken, NJ: Wiley.

Messina, J., E. Walker, N. Moore, S. Grady, J. Maitima, J. Olson, and J. Kaneene. 2007. *A Dynamic Ecological Simulation Model of Tsetse Transmitted Trypanosomosis in Kenya*. Kenya: Research Sponsored by the National Institutes of Health, Office of the Director, Roadmap Initiative, and NIGMS: Award No. RGM084704A.

Moore, N., and J. P. Messina. 2010. A landscape and climate data logistic model of tsetse distribution in Kenya. *PloS One* 5(7):1–10.

Muriuki, G. W, T. J. Njoka, R. S. Reid, and D. M. Nyariki. 2005. Tsetse control and land–use change in Lambwe Valley, south-western Kenya. *Agriculture, Ecosystems and Environment* 106(1):99–107.

Neteler, M., and H. Mitasova. 2008. *Open Source GIS*: *A GRASS GIS Approach*. New York: Springer.

OGIS. 1999. Open GIS Consortium: Open GIS Simple Features Specification for SQL (Revision 1.1). www.opengeospatial.org (accessed November 28, 2010).

Olson, J. E. 2003. *Data Quality: The Accuracy Dimension*. San Francisco, CA: Morgan Kaufmann Publishers, Elsevier Science.

Ostfeld, R., F. Keesing, and V. T. Eviner. 2008. *Infectious disease ecology: The effects of ecosystems on disease and of disease on ecosystems*. Princeton, NJ: Princeton University Press.

Parent, C., S. Spaccapietra, and E. Zimányi. 2006. The MurMur project: Modeling and querying multi–representation spatio-temporal databases. *Information Systems* 31:733–769.

Python Software Foundation. 2010. *General Python FAQ—Python v2.6.5 Documentation*. Available at http://docs.python.org/faq

Ramsey, P. 2005. PostGIS manual. *Refractions Research Inc*, http://www.dcc.fc.up.pt/~michel/TABD/postgis.pdf

Rivest, R. 1992. *The MD5 Message-Digest Algorithm*. MIT Laboratory for Computer Science and RSA Data Security.

Shekhar, S., and S. Chawla. 2003. *Spatial databases: A tour*. Upper Saddle River, N.J.: Prentice Hall.

Shi, W., M. F. Goodchild, and P. Fisher. 2002. *Spatial Data Quality*. New York: Taylor & Francis.

Siringi, S.. 2003. Kenya rejects drugs deal. *Lancet Infectious Diseases* 3(6):320–320.

Smith, T. R, and J. Frew. 1995. Alexandria digital library. *Communications of the ACM* 38(4):62.

Stonebraker, M., and G. Kemnitz. 1991. The Postgres next-generation database-management system. *Communications of the Acm* 34(10):78–92.

Stonebraker, M., and D. Moore. 1996. *Object–Relational DBMSs: The Next Great Wave: Object–Relational DBMSs: The Next Great Wave*. San Francisco: Morgan Kaufmann Publishers.

Stonebraker, M., and L. A. Rowe. 1986. The design of Postgres. In *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data*. Washington, DC: ACM.

Sutherst, R. W. 2004. Global change and human vulnerability to vector-borne diseases. *Clinical Microbiology Reviews* 17(1):136–173.

Tatem, A. J., S. I. Hay, and D. J. Rogers. 2006. Global traffic and disease vector dispersal. *PNAS* 103(16):6242–6247.

van der Lans, R. F. 2007. *Introduction to SQL: Mastering the Relational Database Language*. 4th ed. Upper Saddle River, NJ: Addison-Wesley.

Watson, R. T. 2004. *Data Management: Databases and Organizations*. 4th ed. Hoboken, NJ: Wiley.

Wint, W. 2001. Kilometre Resolution Tsetse Fly Distribution Maps for the Lake Victoria Basin and West Africa. Vienna: PATTEC.

World Health Organization. 2001. "Report on African Trypanosomiasis (sleeping sickness)." Paper read at Report of the Scientific Working Group meeting on African trypanosomiasis, Geneva, June 4–8, 2001.

———. 2005. Control of Human African Trypanosomiasis: A Strategy for the African Region. World Health Organization, Regional Committee for Africa.

———. 2006. *African Trypanosomiasis*. Available at http://www.who.int/media centre/factsheets/fs259/en/ (accessed February 10, 2007).