

Spatially Dynamic Time-Series Animations of Raster and Vector Data Through ArcIMS

ABSTRACT

Web-based geospatial technology provides the means to enhance the decision making of natural resource managers and the education of natural resource users. This paper discusses the development of a web-enabled application that allows the temporal and spatial representation of raster, vector, and tabular data for natural resource processes and uses through spatially dynamic time series animations. Stakeholder needs for interactive browsing of spatial and temporal event data in the areas of strategic wildfire modeling, support of natural resource management, and natural resource conservation education, were met through the development of a database-driven, Java web server application in combination with ArcIMS and ArcSDE.

TABLE of CONTENTS

- [1. INTRODUCTION](#)
 - [1.1. Problem Statement](#)
 - [1.2. An Adaptive Research and Development Model](#)
- [2. REQUIREMENTS](#)
 - [2.1. Functional Requirements](#)
 - [2.2. Non-Functional Requirements](#)
- [3. Data](#)
 - [3.1. Time Series Data](#)
 - [3.1.1. AVHRR](#)
 - [3.1.2. MODIS](#)
 - [3.2. Reference Data](#)
 - [3.3. Climate Data](#)
 - [3.3.1. El Niño Southern Oscillation Index \(ENSO\)](#)
 - [3.3.2. Weather Data](#)
 - [3.3.3. Palmer Drought Severity Index \(PDSI\)](#)
- [4. DESIGN PROCESS](#)
 - [4.1. Overall System Design](#)
 - [4.1.1. Architecture](#)
 - [4.1.2. MVC](#)
 - [4.1.3. Component Design](#)
 - [4.1.4. Integration Considerations](#)
 - [4.2. Business Logic](#)
 - [4.2.1. Controller](#)
 - [4.2.2. Model](#)
 - [4.3. Presentation Logic](#)
 - [4.4. Database Design](#)
 - [4.4.1. Conceptual Design](#)
 - [4.4.2. Logical Design](#)
- [5. IMPLEMENTATION](#)

5.1. System Environment

5.1.1. Software

5.1.2. Hardware

5.2. Software Development

5.2.1. Presentation Logic

5.2.2. Business Logic

5.2.2.1. Process Flow

5.2.2.2. Data Flow

5.2.2.3. Utilities

Database Utility Package:

Map Utility Package:

Precipitation Graphing:

PixelGrabber:

5.3. Database Implementation and System Integration

5.3.1. Database Creation

5.3.2. Database Implementation

5.3.2.1. Optimization

5.4. Maintenance

6. CONCLUSION

7. ACKNOWLEDGMENTS

8. CITATIONS

APPENDIX A - List of Acronyms

AUTHOR INFORMATION

1. INTRODUCTION

In response to demands from natural resource managers, the University of Arizona (UA) has created a web-based information system that allows users to assimilate disparate datasets and visualize time-varying spatial relationships. This system serves to enhance users' understanding and management of the environment and natural resources as they vary across the landscape and change through time. The application has the flexibility to handle raster, vector and tabular data at varying spatial and temporal resolutions, such as value-added products derived from Earth Observation System (EOS) satellite imagery, spatially distributed climate information, point-based weather data, and other geospatial data. It provides the user with the spatial dynamic of an Internet map server, the temporal dynamic of a time series animation, and the analytical capability of graphing software.

The application is the result of a collaborative, on-going knowledge exchange between stakeholders (natural resource managers from government agencies, ranchers, resource conservationists, wildfire managers and information officers, educators, etc.) and researchers (including the web and database development team). This two-way exchange provides continual assessment of user requirements while simultaneously educating those users on how to make use of the application and interpret the data and value-added products it provides.

In order to support future developments necessary to accommodate the growing number and variety of users, the application framework has been designed to be extensible. To achieve increased flexibility, the application involves a tiered architecture that separates application functionality into three independent layers, effectively decoupling the business logic from the presentation and the database in the software architecture. In this way, complex flows are separated into more manageable layers and changes in one layer do not affect other layers. A standard interface protocol maintains each layer of the system, facilitates communication between them, and controls process logic. By sharing a standard protocol, system components can be substituted and the overall system can be revised irrespective of the technology used in the front-end, middleware, or back-end. Wherever possible, reusable (or generic) components are developed, facilitating collaboration between projects with different goals or audiences but similar development needs. This development strategy encourages team programming, which increases the efficiency of the process and the effectiveness of each application element. It also permits an adaptive research and development approach that permits end-users more direct input not only in the initial systems specifications, but also in enhancements and modifications throughout the life of the project.

The application, which can be customized for specific project needs, has been implemented in two major natural resource management research and development initiatives at the University of Arizona: a NASA/Raytheon sponsored "Infomart" called RangeView (<http://rangeview.arizona.edu>), and an EPA Star Grant called Wildfire Alternatives –

In order to integrate user needs, the creation of value-added spatial products, and applications development, a team from several departments in the University of Arizona and from other participating institutions has been formed. The Arizona Remote Sensing Center (ARSC) at The University of Arizona is primarily responsible for the development of the system.

1.1. Problem Statement

Natural resource management in the Western U.S. aimed at longer-term, sustainable use of public land requires communication and interaction between natural resource agency personnel, ranchers, developers, and other land users. Bridging the knowledge gap between these groups in terms of new tools and technologies is key to effective communication and management. Natural resource managers have been making requests for value-added, easy-to-use, and spatially explicit products derived from remote sensing data since the launch of the first civilian satellite in 1972. Government officials (ARSC 2000) and natural resource managers (Marsh et al. 2001; 2002) in the agriculture sector (Moran et al. 1997) have echoed the constraints to adoption of such products. From the end user's standpoint, the adoption of geospatial technology is predicated on the regular availability of timely, quantitative, validated, operational, and site-specific information products and the direct involvement of end users in product development, implementation and evaluation. Traditionally, the response to these demands has been on a project-by-project basis, often without the continuity necessary to generate broader-based demand. For sectors where geospatial expertise is often unavailable at the local level, the project-by-project approach has done little to transfer technology or expand the overall user base. Successful natural resource management applications have tended to be spatially explicit (classifications, inventories, assessments, etc.) or have focused on change detection. While these applications are useful to natural resource managers for a specific location or a specific time period, it has been difficult to provide frequent, operational access to satellite imagery and other geospatial data that could support the monitoring of natural resources and related phenomena through time.



Figure 1. Technology transfer with stakeholders at the V-V Ranch, Arizona.

1.2. An Adaptive Research and Development Model

“The voice of the customer” that drives the research and development is being captured and incorporated through an iterative, adaptive research and development model. This model allows midstream adjustment in development through responsiveness to ongoing evaluation and user feedback. The strategy for technology transfer is based on a “high tech – high touch” approach that integrates a web-based system of product delivery with informal knowledge exchange, both between potential users and between users and the research and development team. Futurist John Naisbitt demonstrated powerful insight in 1982 when he suggested, “whenever new technology is introduced into society, there must be a counterbalancing human response—that is, *high touch*—or the technology is lost. The more high tech, the more high touch.” The mechanism we are using for diffusion of innovation is based on Cooperative Extension’s

traditional “technology adoption cycle” that was first formally documented by Ryan and Gross (1943), a model that has been embraced by marketing experts in virtually every high technology company (Moore 1999). It was hypothesized that the pool of adopters might be segmented by the time needed to adopt, as shown in [Figure 2](#) below, which depicts a theoretical curve for general technology diffusion and segmentation of technology adopters. Identifying the different groups allows for conscious facilitation of interactions between early adopters and later ones, speeding the overall dissemination process.

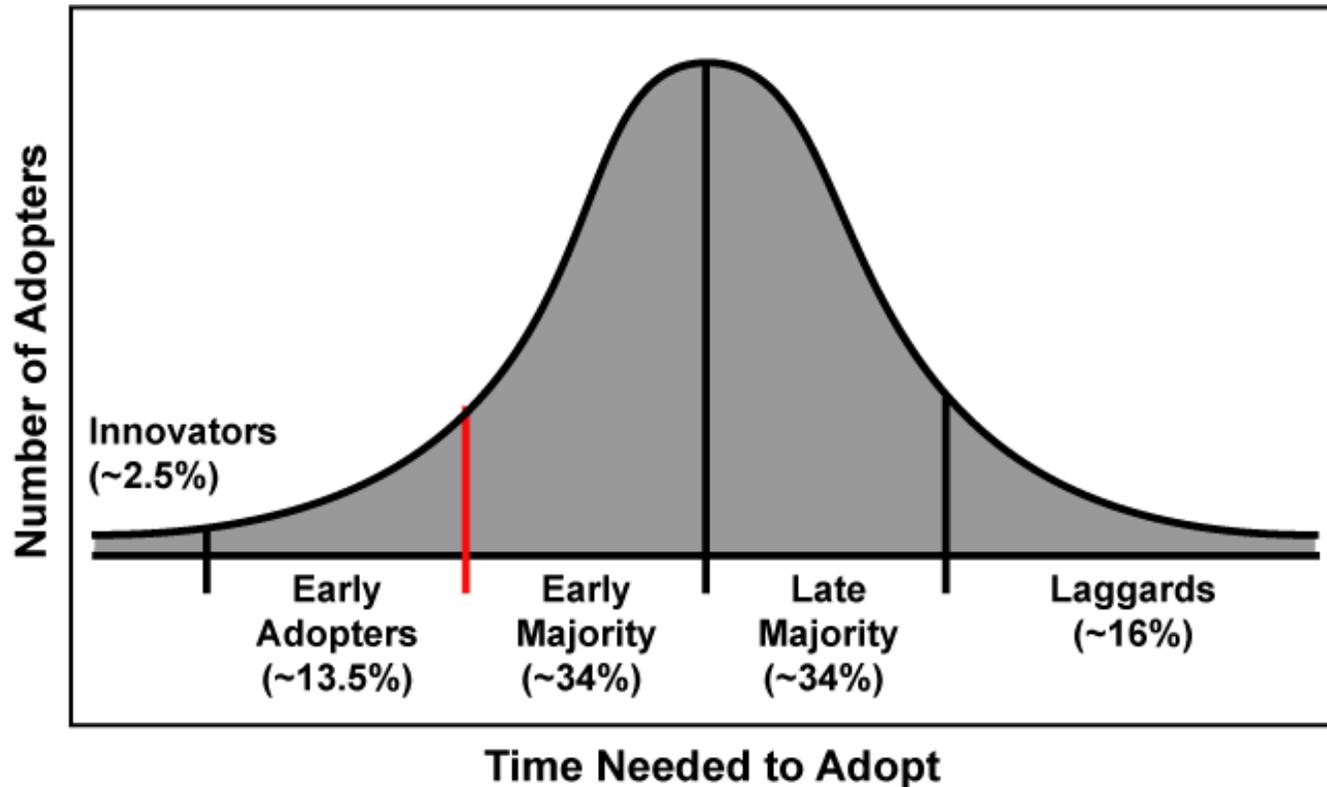


Figure 2. The technology adoption cycle (from ARSC 2000, courtesy of Ada Leung).

In our application, two interrelated activities make it possible to understand end user needs. The first is the use of marketing research to systematically assess user needs. We began the assessment with the state, local and tribal government sectors already using geographic information systems (GIS) (ARSC 2000) and later focused specifically on natural resource managers (Marsh et al. 2001; 2002). Secondly, UA Cooperative Extension and UA/NASA Space Grant teamed up to create a “Geospatial Extension Program” designed to bring geospatial research and technology to the public, with an emphasis on natural resource managers (Orr et al. 2001).

The survey of state, local and tribal governments focused on identifying factors that would encourage or constrain adoption (ARSC 2000). This research demonstrated that a) data applications need to cut across sectors, b) potential users of remote sensing products lack necessary knowledge, c) institutional challenges surrounding the perceived costs and risks in making an investment in this technology inhibit adoption, and d) local “networks” for data and technical support are critical for both adoption and sustained use. Barriers to the diffusion of innovation and the adoption of high technology data products are not exclusively the concerns of applied research and development groups. A “whole product” marketing methodology suggests that approaching the development of the application from the perspective of the user and the technology simultaneously can enhance the potential to adopt.

In response to the survey results, the UA team has conducted a series of site visits with natural resource extension agents across Arizona to inventory preferred applications, to assess educational requirements and to identify early adopters or “lead users.” These lead users now regularly assist us in systems specification and in evaluating intermediate products. The UA has also held a series of workshops and seminars with lead users and other stakeholders (Marsh et al. 2001; 2002; Garfin and Morehouse 2001). Finally, the UA has partnered with Marketing Intelligence, LLC (MI), a marketing research firm that is helping define a business model for the web-based dissemination of satellite remote sensing data and other geospatial products for land use management and determining critical factors necessary to achieve long-term sustainability (Jain et al. 2001; 2002).

Numerous presentations, workshops and training sessions have been held to identify lead users, to introduce intermediate products, and to educate end users on how these innovations can be used as a decision aid that

compliments their current management practices ([Table 1](#) and [Figure 3](#)).

Table 1. Face-to-face contacts with lead users and potential future users.

	# of Events	Participants
Presentations	36	1852
Training Programs	17	504
Workshops	24	532
Total	77	2888



Figure 3. Training Scene at the October 4-5, 2002 RangeView Workshop.

The initial marketing research revealed a particular need for an application that could capture vegetation dynamics through time and over large areas. In response, the University of Arizona set up a web-based information system that provides users with access to near real-time spatial data (i.e. vegetation indices derived from satellite imagery acquired frequently across large areas from the NOAA AVHRR and NASA Terra MODIS satellite sensors, spatially distributed climate data, ancillary vector data layers to help locate a particular area of interest, and analytical products associated with time series data). The initial interface took a number of different forms early on as the developers attempted to provide functionality to the lead users, who in turn provided feedback on needed modifications and enhancements. These requests became more specific as users came to understand the data and information products and as the functionality of the application developed.

Involving user communities in product development throughout the project and soliciting periodic feedback on services and products has proved critical for a broader adoption of geospatial data and technology. The vast majority of end users had little or no initial experience in computerized mapping or time series satellite imagery. Most only became comfortable in recommending system changes after attending a presentation or workshop. By integrating the continual assessment of user specifications with a systematic educational outreach program, we were able to refine our understanding of user requirements and institute the architectural flexibility in back-end systems necessary to handle new requests that will inevitably come as natural resource managers become more familiar with these data products.

2. REQUIREMENTS

The results of the marketing research, site visits, workshops, surveys, and one-on-one conversations with users

described in [Section 1.2](#) demonstrated the need for an integrated web application that will allow users to assimilate many disparate data sources and perform complex analysis of the relationships between the various datasets. The research and development team worked directly with the marketing research team and the geospatial extension program to transform a list of functions requested by natural resource managers (including ranchers, government agency personnel, and others interested in natural resources), wildfire managers and information officers, and educators into user requirements and systems specifications.

Natural resource managers from government agencies listed a) comparing current greenness to long-term normal levels, b) forecasting greenness based on historical patterns, c) understanding historical variations in greenness, and d) analytically accessing site-specific data while spatially visualizing the time series information (e.g. temporal graphs of vegetation greenness and/or rainfall for specific locations) as the most important benefits they would derive from the web-based, spatially dynamic time series application. Ranchers indicated that the spatially dynamic time series functions should allow them to a) monitor the immediate response of vegetation to rainfall, b) monitor overall range conditions, c) determine the relative amount of vegetation growth between pastures that will occur in response to rainfall or climate events such as El Niño, and d) observe patterns of vegetation growth during a year. Both ranchers and agency personnel emphasized the role such access to EOS data and value-added products could play in encouraging dialogue when management disagreements over vegetation trends occur, particularly during periods of drought or wildfire risk, or where wildlife and cattle might be expected to use the same pasture in the months ahead. The high temporal resolution and landscape coverage helped in longer term planning as well as shorter-term identification of areas that needed specific ground monitoring. Moreover, the coarse spatial resolution was actually applauded by agency personnel and ranchers alike: the data is too coarse to be used in regulatory applications, and yet powerful enough to provide visual insight into vegetation trends and a basis for dialog when resource management strategies are in dispute.

Wildfire managers indicated a need for information that could help them assess the probability of wildfire under specific intermediate and longer-term climate scenarios. They also requested vector data on past extents of wildfires embedded in time series maps. Fire information officers and educators were equally interested in the longer-term climate and human influences on wildfire probability, with particular emphasis on the ability to spatially and temporally visualize the landscape and climate trends that lead to increased or decreased risk (longer-term climate patterns and vegetation response leading to changes in fuel loads). Both groups expressed interest in being able to “see behind the model,” or to visualize an individual model and thereby analytically assess the overall wildfire probability for a given location (a virtual sensitivity analysis).

All natural resource managers and educators expressed interest in the ability to readily extract (download electronically for inclusion on reports or print directly to hard copy) data (tables), graphs, images and maps. While privacy was of major concern, many of the stakeholders requested the ability to maintain one or multiple sessions so that a given map extent and time frame as well as any analysis resulting from a session on one day could be revisited and continued the next time. They also requested multiple means of locating areas of interest, including strong supporting ancillary data such as jurisdictional boundaries and township/range/section locators and a locator based on the entry of GPS coordinates. Of particular interest to local managers and ranchers was the ability to upload their own datasets (rain gauge data, vegetation transects, wildfire severity ground observations, etc.). The needs assessment also helped hone data presentation for enhanced utility and comprehension, such as choosing mapping color pallets and transforming research jargon into terminology more meaningful to stakeholders.

The application described in this document will attempt to provide a variety of functionality for analysis of time series data, particularly satellite-derived vegetation indices, in conjunction with climate, topography, and other satellite or ground-based data. The application will allow users to visualize time-varying spatial relationships that exist among datasets. This will be achieved through the use of Internet mapping, time series animations, graphing, and data browsing tools available in an integrated web application.

The researcher and user interaction demonstrated that there are potential uses for this type of application in many different disciplines and functional areas. In order to support future developments in other areas, the application framework must be extensible.

Based on user feedback and internal development goals, the following list of requirements was generated. [Section 2.1](#), Functional Requirements, outlines the intended behaviors of the system and features that it should support. [Section 2.2](#), Non-Functional requirements, lists the constraints on system performance and quality.

2.1. Functional Requirements

The application features and functional user requirements distilled from the marketing research are listed below:

1. Multiple Map Viewers
 - The application should provide multiple map viewers to enable users to compare several datasets at the same time.
2. Select Datasets
 - Users can select preferred datasets for each of the viewers.
 - The datasets can be raster (satellite imagery) or vector (lightning strikes).
 - Since the temporal granularities among datasets can vary, a date matching algorithm will be developed to match dates among different datasets for each viewer.
 - Users can select analytical products derived from time series (i.e. difference from average, difference from previous period, etc.)
3. Generate Time Series Data Based on User Queries
 - a. Select by date (start date and end date)
 - The user chooses a start date and an end date along with dataset(s).
 - The query is processed to return all images matching the criteria.
 - If the user is viewing multiple datasets, the dates must be matched and only matching dates are returned.
 - b. Select by Month
 - User chooses a particular month along with dataset(s), and a range of years.
 - All images for that month within the range of years are returned.
 - c. Select by Year
 - User chooses a particular year and dataset(s) for each map viewer.
 - Images from January to December of the year and dataset selected are returned for each viewer.
 - d. Select by ENSO Patterns
 - User chooses a particular set of ENSO conditions along with dataset(s) and a range of years.
 - All images that meet the specified criteria are returned.
 - e. Select by Precipitation Parameters
 - User chooses precipitation conditions (e.g. above/below average, above/below a specified amount) along with dataset(s), and a specified time period.
 - All images that meet the specified criteria are returned.
 - f. Select by Palmer Drought Severity Index
 - User chooses drought conditions (e.g. above/below average) along with dataset(s), and a specified month.
 - All images that meet the specified criteria are returned.
 - g. Select by Fire Risk Index
 - Choose a particular fire/watershed.
 - Find all periods where the fire risk index was above or below a particular value.
 - Display the fire risk maps.
4. User Interaction with Maps to Customize Display
 - a. Choose Spatial Extent – User Options to Zoom to Their Area of Interest
 - Standard zoom and pan tools will be provided, including the “rubber band” zoom.
 - A drop-down list of pre-defined zoom regions will be provided on the interface. A second and third drop-down menu with more detailed regions will change according to the selected region in the previous menu. For example: If the user chooses Arizona as the region in menu 1, then menu 2 will contain counties, national forests, climate zones, etc. for Arizona.
 - b. Add Reference Layers
 - The users will be able to add reference layers (cities, roads, rivers, etc.) to the map to add location information. Multiple resolutions of reference layers (high, medium and low detail roads, etc.) are available and become accessible depending on the current map scale.
 - The labels for the reference layers will toggle on and off.
5. Map Tools
 - a. Identify Vector Features on Map
 - b. Identify Raster Cell Value on Map
 - c. Measure Distances on Map
6. Graphing: Generate Time Series Graphs Based on User Selections
 - a. Precipitation Graphing
 - User chooses a precipitation gauge, or a set of gauges, along with the time range of interest.
 - Time series graphs are generated and displayed along with maps of selected datasets.
 - b. Pixel Values Through Time
 - User selects a pixel from the viewing map and a time range of interest.
 - A graph is displayed of the chosen pixel through time.
 - User can overlay related information (a comparable time range of interest or the mean value across a specific time range of interest).
7. Display Resulting Maps and Graphs
 - a. Animation of Maps and Graphs
 - All maps/graphs generated in response to user selections are animated in order of date.
 - The animator should include controls for the user to play/stop the animation, step through individual images, and

rewind to the beginning.

b. Static Display of Maps and Graphs

- The maps generated in response to user selections are displayed in the map viewers.
- Forward and Back buttons will allow the user to browse through the results.
- The selection menu remains available so that changes can be made and the new results displayed.

8. Other Features

a. Save User Sessions

- The user can save several sessions.
- This will save all of their settings, including number of viewers, spatial extent, datasets, reference layers, and graphs.

- The session can be loaded when the user logs on in the future.

b. Resulting Maps and Graphs Printable with High Quality

c. Download Functionality

- The user can download original or value-added data.
- The user can download analytical components for reports (specific time series images, graphs, tables, etc.)

2.2. Non-Functional Requirements

Internally, it was also necessary to make decisions on non-functional requirements that arose from the strategies proposed to help address the functional requirements. These include factors that pose constraints on the systems as a whole, system performance, and the ability of the system to meet the functional requirements:

1. Web-based system that allows user to access functionality through a browser - compatible with Internet Explorer 6+, Netscape 7+, and Mozilla 1.3+.
2. Flexible system design, allowing developers to easily change data and business logic to create new applications.
3. Development of reusable components that can be used by other projects.
4. Development of automated routines to “pull” time series datasets, to create analytical products from these data, and to automatically add the datasets to the application.
5. Reliable system that can provide accurate data and visualization.
6. User-friendly interface that is easy for users to navigate and that provides immediate access to instructions, tutorials and help functionality.
7. Maintainable system that can be upgraded smoothly without changing a lot of functions.
8. Integrated application – a “one-stop shop” - everything available from one place.
9. Quick response system that can reply to user requests in a reasonable amount of time.
10. Adaptability – the capacity in the research and development process to adjust to changes in user priorities and the identification of unanticipated needs that are identified as the geospatial extension and marketing research teams receive feedback on prototypes of recently developed functions.

3. Data

The most commonly expressed need by users was the ability to monitor landscape (vegetation, climate, wildfire, etc.) dynamics through time and across space. These requests pointed to specific data products, some already available, but not operationally accessible to most natural resource managers, and others which had to be derived from primary data. The datasets currently used by the application consist of spatially distributed vegetation indices derived from satellite imagery (AVHRR and MODIS), digital maps of related biophysical information (i.e. topography, soils maps and vegetation classifications), other reference information to help orient users spatially, and climate datasets corresponding to the same time series as EOS data. The climate data currently includes weather data (precipitation and temperature) and an El Niño Southern Oscillation (ENSO) Index, and will be expanded in the future to include other relevant datasets such as Palmer Drought Severity Index (PDSI). The application was designed to include other types of data as determined by needs identified in the future. For the purposes of this paper, the spatial data can be divided into three broad classes: Time Series Data, Reference Data and Climate Data.

3.1. Time Series Data

Time series data includes map layers that will be used to convey time-varying spatial information to the users. Although time series data are primarily in raster format, we recognize that the vector dataset may also be used to represent time series information (e.g. lightning strikes, wind vectors, etc.). The application has been designed to accommodate both raster and vector datasets as time series data.

Currently, two primary satellite sensors supply the majority of the data used in the application: the Advanced Very High

Resolution Radiometer (AVHRR) and the Moderate Resolution Imaging Spectroradiometer (MODIS). The application has been designed to support future data from other sensors such as TM and ETM+, as well as other types of image data such as Digital Ortho Photo Quarter Quadrangles (DOQQ). The AVHRR and MODIS datasets currently in use by the system are described below.

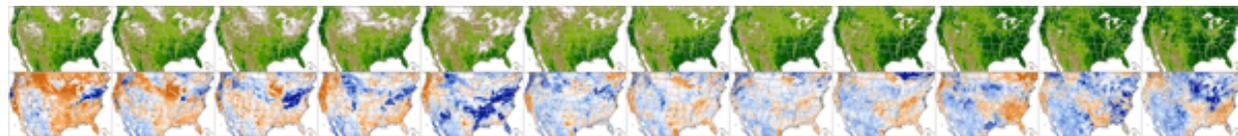


Figure 4. Time series of biweekly NDVI composite and a calculation of the Difference from Average NDVI from AVHRR satellite sensors for January to June 2003, contiguous USA.

3.1.1. AVHRR

The AVHRR sensor is an instrument aboard NOAA satellites: NOAA-9, NOAA-11, NOAA-14, and NOAA-16. The vegetation index used in the project is derived from AVHRR spectral data by the USGS EROS Data Center in Sioux Falls, South Dakota. Their Normalized Difference Vegetation Index (NDVI) is regularly used by researchers and is available on a subscription basis, with updates generally available within 10 days of satellite data acquisition. The dataset extends back to 1989 and is based on a temporal maximum value composite period of 14 days, which limits the impact of clouds on the vegetation signal. The data have a spatial resolution of 1 km and, for this project, a geographic coverage area that includes the coterminous U.S., southern Canada, and northern Mexico. In order to improve analyzability of the data, we also produce additional products from the NDVI data including difference from average, difference from previous period, and difference from previous year.

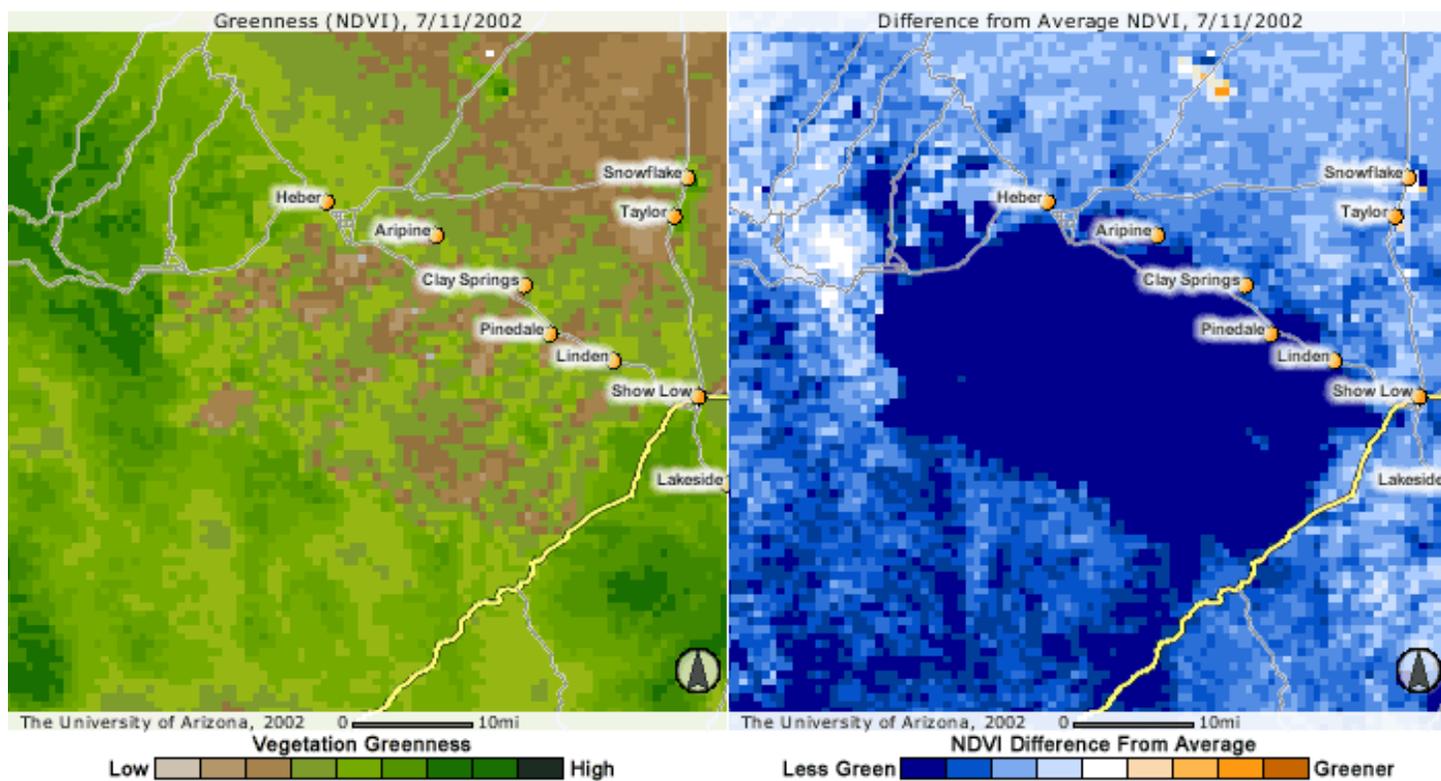


Figure 5. Biweekly NDVI and difference from average NDVI composite images from AVHRR satellite sensors for July 2002 of the Rodeo-Chediski wildfire area in Arizona (Laura Baker).

3.1.2. MODIS

The MODIS sensor is an experimental instrument aboard the Terra (am) and Aqua (pm) satellite platforms that are part of the NASA Earth Observing System (EOS). MODIS acquires multispectral data at several spatial resolutions (1000m, 500m, and 250m). Numerous MODIS data products are generated for study of land-based systems, including two vegetation indices, NDVI and the Enhanced Vegetation Index (EVI) (for more information see <http://modis-land.gsfc.nasa.gov>). MODIS data have been available since 2000, but the experimental nature of both the sensor and the generation of value-added products, such as the vegetation indices, make the data processing stream less systematic (and therefore more of a challenge to automate) than AVHRR. ARSC obtains MODIS data and value-added products

through the DAACs and their recently instituted Data Pools. The data are made available in 1200 by 1200 km tiles that cover the globe. However, we are currently focusing on the six tiles that cover the southwestern U.S. and northern Mexico. Though processed differently than AVHRR data, many of the MODIS products are composited images with one image representing the entire composite period (8 and 16 days). The products currently in use or proposed for future use in this application, their corresponding composite periods, spatial resolutions, and geographic coverage areas are listed in [Table 2](#).

Table 2. Displayed MODIS Products

MODIS Product	Composite Period	Spatial Resolution	Coverage Area
Vegetation Indices (NDVI, EVI)	16-day	1000 m	Southwest U.S. & northern Mexico
Vegetation Indices (NDVI, EVI)	16-day	250 m	Arizona
Land Cover/Land* Cover Change	96-day	1000 m	Southwest U.S. & northern Mexico
Thermal Anomalies, fires, biomass burning*	8-day	1000 m	Southwest U.S. & northern Mexico
Leaf Area Index (LAI) and FPAR*	8-day	1000 m	Southwest U.S. & northern Mexico
Net Photosynthesis and Primary Productivity*	8-day	1000 m	Southwest U.S. & northern Mexico

* Proposed

The Vegetation Indices: Normalized Difference Vegetation Index (NDVI) and Enhanced Vegetation Index (EVI) are the principal products currently used in this project. In addition to the vegetation indices acquired from EOS, ARSC produces several products from the NDVI and EVI data using in-house image processing protocols. These include difference from previous period and difference from previous year for both NDVI and EVI. ARSC is also actively pursuing data continuity research in an attempt to make the most of the long-term nature of the AVHRR dataset relative to MODIS products.

MODIS NDVI Images for January 16, 2003

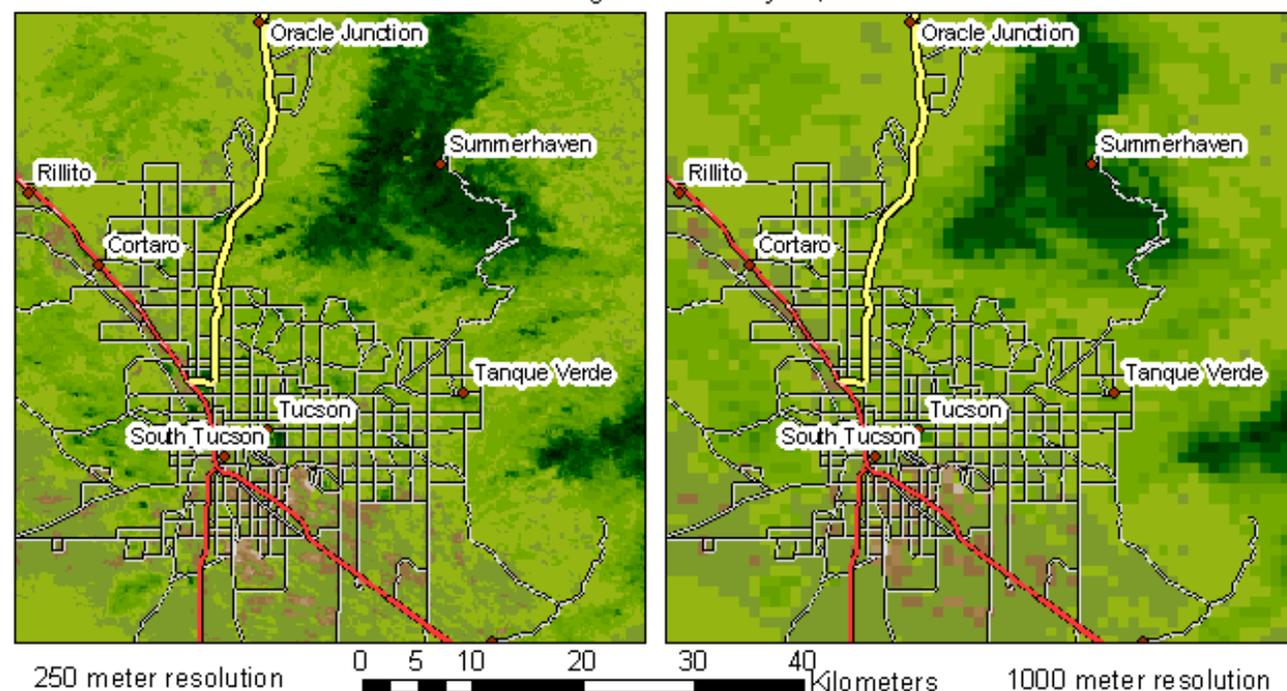


Figure 6. MODIS NDVI at 250m vs. 1000m resolution, Tucson, Arizona. MODIS 250 meter NDVI imagery provides improved resolution for interpreting and assessing vegetation greenness dynamics across landscapes. The 250m image shows greater variation in greenness than the 1000m image both in the mountainous region in the northeast and the desert region in the south (Laura Baker).

3.2. Reference Data

Reference layers include any map layers that are used for navigation and place identification. The application provides users with a map viewer to visualize time series data layers such as satellite images. Map tools allow users to zoom and pan around the map. However, a satellite image alone will not provide the user with enough information for navigation. With the addition of key reference layers such as roads, counties, rivers and towns a user will be able to easily find any area of interest on an image. These reference layers can also provide another analytical resource. For example, one may be able to identify a trend of decreasing vegetation greenness in the outskirts of developing urban areas. Vector data is primarily used for reference layers. However, there are raster datasets that may be used for reference such as Digital Elevation Models (DEM). The application will support both raster and vector reference layers.

3.3. Climate Data

Several types of climate data are used for analysis in conjunction with the spatial data layers. El Niño Southern Oscillation Index (ENSO) and Weather Data (precipitation and temperature products) are currently available. The Palmer Drought Severity Index will be made available to end-users in the near future.

3.3.1. El Niño Southern Oscillation Index (ENSO)

ENSO indices describe the interannual variation of atmospheric and oceanic conditions in the equatorial Pacific that affects climate conditions on a global scale. The Multivariate ENSO Index (MEI) used in this project is based on six main observed variables over the tropical Pacific: sea-level pressure, zonal and meridional components of the surface wind, sea surface temperature, surface air temperature, and total cloudiness fraction of the sky. The MEI is available on a monthly basis and represents conditions over the previous two months (NOAA-CIRES Multivariate ENSO Index <http://www.cdc.noaa.gov/~kew/MEI/mei.html>). This is used to describe ENSO conditions on a global scale and can be categorized as El Niño, La Niña, or neutral. Although several ENSO indices are available, we chose to use the MEI, because it is updated on a consistent and timely basis.

3.3.2. Weather Data

Weather data is available for weather stations in Arizona on a daily basis and is downloaded from NOAA-NCDC Climate Data Online (<http://cdo.ncdc.noaa.gov/plclimprod/plsql/poemain.poe>). Storing daily data in the database allows us to aggregate the data over larger time periods such as the composite periods of satellite data. While many different weather parameters are measured at the gauges, this application currently makes use of total precipitation, minimum temperature, and maximum temperature. Each gauge location corresponds to a point in a reference data layer. We also plan to include lightning strike data as either point data or lightning strike density images as the data becomes available to us.

3.3.3. Palmer Drought Severity Index (PDSI)

The PDSI uses temperature and rainfall information to describe dryness or drought conditions. The PDSI is an important climatological tool for evaluating the scope, severity, and frequency of prolonged periods of abnormally dry or wet weather. The PDSI is measured monthly for each of the climate zones (regions that have been categorized as having similar climatic conditions) in the United States. Arizona has seven climate zones. In the application, each of the climate zones corresponds to a polygon in a reference data layer. These data are currently not available in this application, but will be incorporated in the future.

4. DESIGN PROCESS

4.1. Overall System Design

The overall system design was developed based on feedback from users about existing tools and proposed product

development. In addition to satisfying the functional requirements ([Section 2.1](#)), the system design had to accomplish several internal objectives as listed in [Section 2.2](#) and be capable of handling all the datasets described in [Section 3](#). One of the primary goals of the developers was to avoid the problems caused by the inflexible designs of precursor applications. These applications had proven to be difficult to maintain, and adding new functionality or changing existing tools was problematic. The proposed system design will result in an application that is platform independent and easily modified, maintained, and expanded for future needs.

4.1.1. Architecture

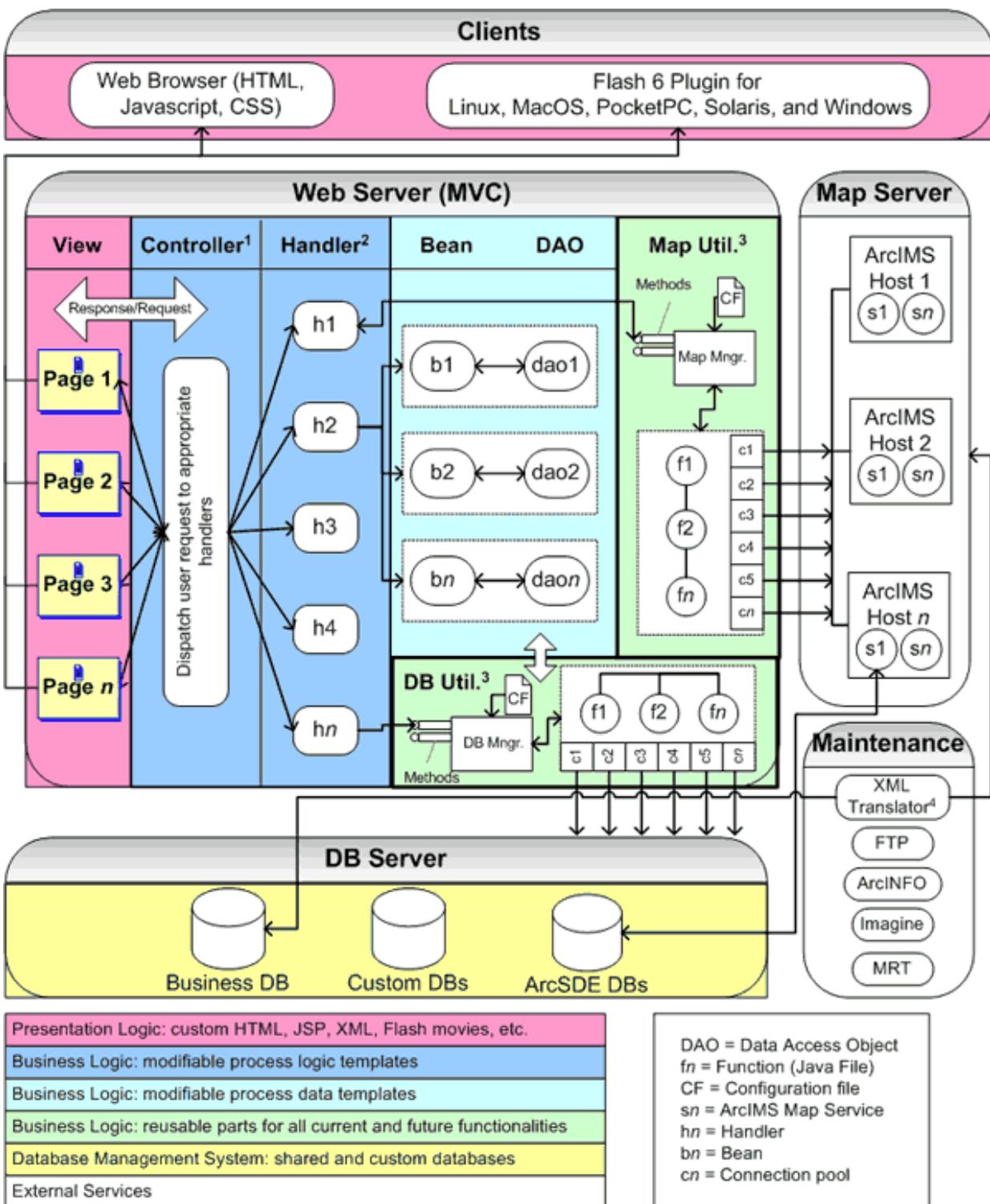
To achieve increased flexibility, the application was designed with a 3-tier architecture that separates application functionality into three independent layers (Schussel 1995; Eckerson 1995; Sadoski and Comella-Dorda 1997) ([Figure 7](#)):

- Layer 1: Presentation Logic
- Layer 2: Business Logic
- Layer 3: Database Management System

This effectively decouples the business logic from the presentation and the database in the software architecture. The layers can also be physically separated on different servers if necessary. This partitioning facilitates future software development and maintenance and makes it easier to add new applications into the existing system. Standardized APIs (Application Program[ing] Interface) can be used to interface business logic with presentation code and database access protocols.

4.1.2. MVC

The application design was also based on the Model View Controller (MVC) Framework, a common object oriented design pattern (Sun Microsystems http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5.html). The MVC pattern contains three main components: the model, or the underlying logical structure of the application which is independent from the user interface; the view, which represents everything the user sees in the interface to interact with the application; and the controller, which is used to communicate between the model and the view. The business logic layer contains the controller and model, whereas the view is represented in the presentation layer ([Figure 7](#)).



1 The controller contains process flow and dispatches request to specific handler.
 2 The handler interacts with resources such as Beans, DAO, DB Manager, and Map Manager.
 3. Each Utility package has only one wrapper (DB and Map Manager) to provide the methods for other functions to use.
 4. The XML translator is responsible for keeping data consistent between the business table and the map services.

Figure 7: Overview of System Architecture (Chih-Lung Kuo).

4.1.3. Component Design

One of the goals of the system design was to develop reusable components when possible. We first created a detailed list of the conceptual components that would be necessary to satisfy functional requirements. Then, we extracted the logical components and corresponding interfaces that could be reused. Two areas were identified that could provide reusable functionality for current and future applications: database and map functions. Packages were designed for these utilities using the wrapper and singleton design patterns (Gamma et al., 1994). An online API will provide a way for other projects and applications to access the utilities provided by these packages. Other system components are designed so that we can easily modify the business rules for use in a different application.

4.1.4. Integration Considerations

Each layer in the system architecture performs specific tasks. Communication between the different layers is the most challenging and important task in our project. To resolve the communication problem, we summarized all of the parameters that will affect the system state during execution. We will use a standard interface to maintain these parameters and control process logic. By sharing a standard protocol, we can easily substitute any component in the system. No matter what technology we use in the front-end, middleware, or back-end, the system can easily be revised.

4.2. Business Logic

The layer framework used in our system architecture provides three major benefits: complex process flows are separated into more manageable layers, changes to one layer do not affect the other layers, and a standard interface is defined for easier integration. In the MVC Framework, the Controller and the Model are present in the business logic layer ([Figure 7](#)).

4.2.1. Controller

The controller is a process kernel with built in knowledge about business process flow and system state. It's a state machine that reads from user input, and processes the requests by dispatching them to the appropriate resource handler, a dedicated program that takes charge of one function. The controller then returns the results to the view/presentation layer.

4.2.2. Model

The Model component contains all of the application logic required for data flow and utilities that support the communication between the business layer and the data layer or other external services such as ArcIMS. Because of stateless feature of a browser, the system will store some permanent records and user session variables into entity or session containers. This will allow the controller to know the state of the system and to send information through multiple layers.

4.3. Presentation Logic

The functional requirements ([Section 2.1](#)) and non-functional requirements ([Section 2.2](#)) for the application demand a relatively thin web client such as a web browser with low computer and Internet bandwidth requirements. This thin client must still offer the functionality usually found in thick stand-alone applications, such as GIS desktop software. The decoupling of the presentation logic from the business logic enabled both of these conflicting goals to be met by developing multiple web clients ([Figure 7](#)) that can be scaled according to the user's software, hardware, and Internet bandwidth limitations. Although the presentation layer is split into multiple client components, they are seamlessly meshed with each other through the use of a common human interface design that invokes the appearance of one tool with multiple tool functions – the "one-stop shop." The defining key features for the presentation logic design can be summed up as:

- One to four interactive maps that can be simultaneously and independently used to browse multiple datasets in space and time.
- One interactive map used to query time series data and to display the results as graphs, tables, and downloadable files.
- Simultaneous display of tool functions and tutorials.
- Saving and restoring of the user's tool session (selected tool function, map extents, datasets, dates, map layout, etc.).
- Solutions for multiple web browser types, bandwidth, and client screen resolution combinations.

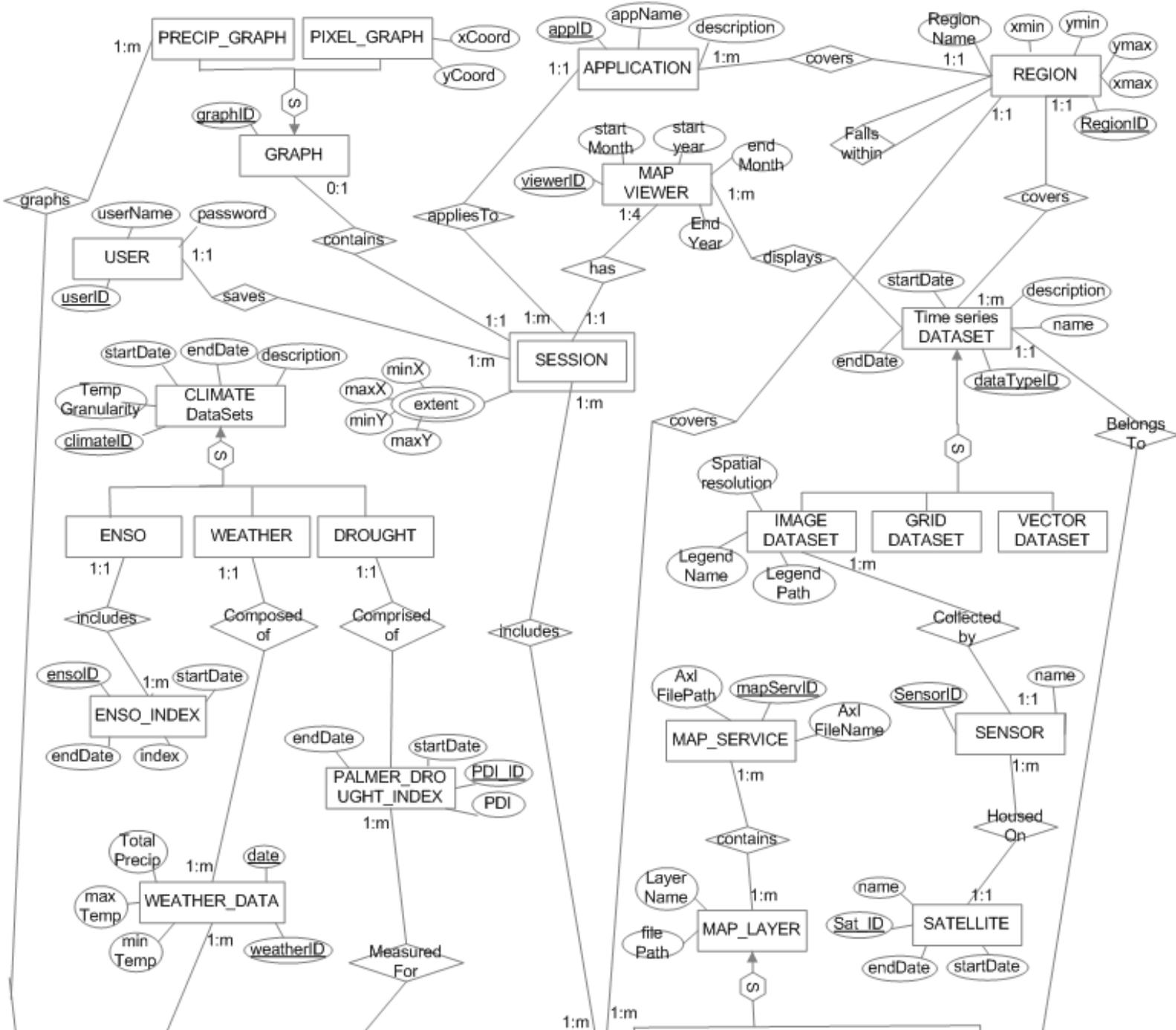
4.4. Database Design

The web-based relational database is an essential component of this project. It will be used to supply the dynamic content to the web applications and will allow the development team to easily manage data, map services, users, and applications. The database was designed to support the project goals of creating a flexible, expandable system.

4.4.1. Conceptual Design

The development team identified a set of conceptual database design goals to help respond to the user requirements and meet project goals. One of the primary objectives of the web application was to provide functionality to allow users to assimilate disparate data sources and visualize time-varying spatial relationships that exist among datasets. Because the assessment of user needs is ongoing, the conceptual design also had to include the flexibility to handle raster, vector and tabular data of varying spatial and temporal resolutions. It was also necessary to allow for easy incorporation of new datasets in the future. It was also important to have a database design that would allow developers to easily manage applications by keeping track of necessary information related to each application. Finally, it was essential to provide the ability to manage users, allowing for authentication and the capacity to store user settings for retrieval in future sessions.

Having established the conceptual design goals, the development team completed a detailed analysis of the user requirements, reviewed the proposed process and data flows, determined what questions would be asked of the database, and determined how to represent the data. An initial list of potential entities was established based on this analysis. The attributes were assigned to the entities and relationships were found between the entities. The design went through several iterations before reaching the final version. The database design for this project followed standard rules (Rob and Coronel 1999; Date 2000). The conceptual design process resulted in the entity relationship diagram (ERD) (Figure 8).



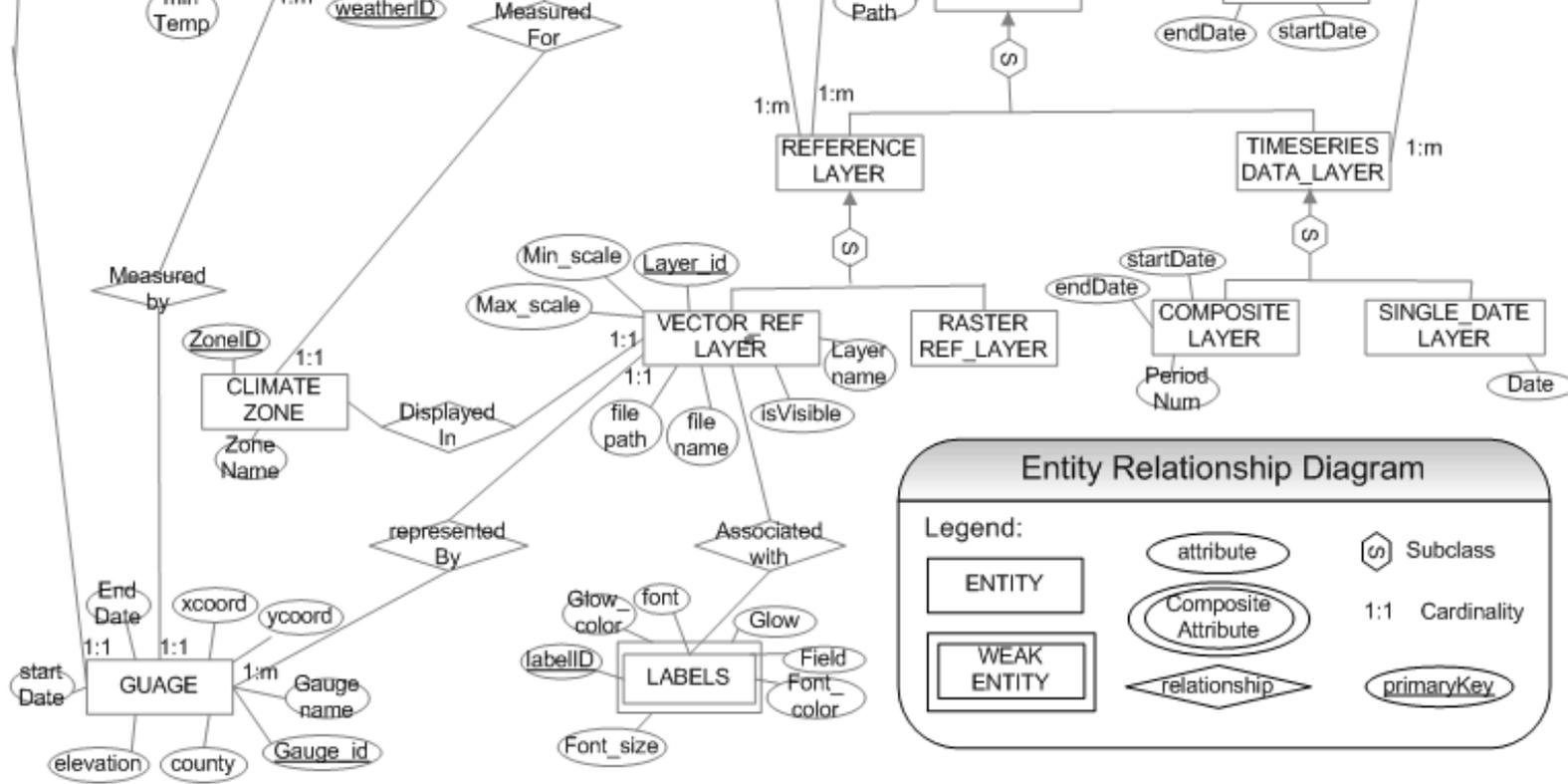
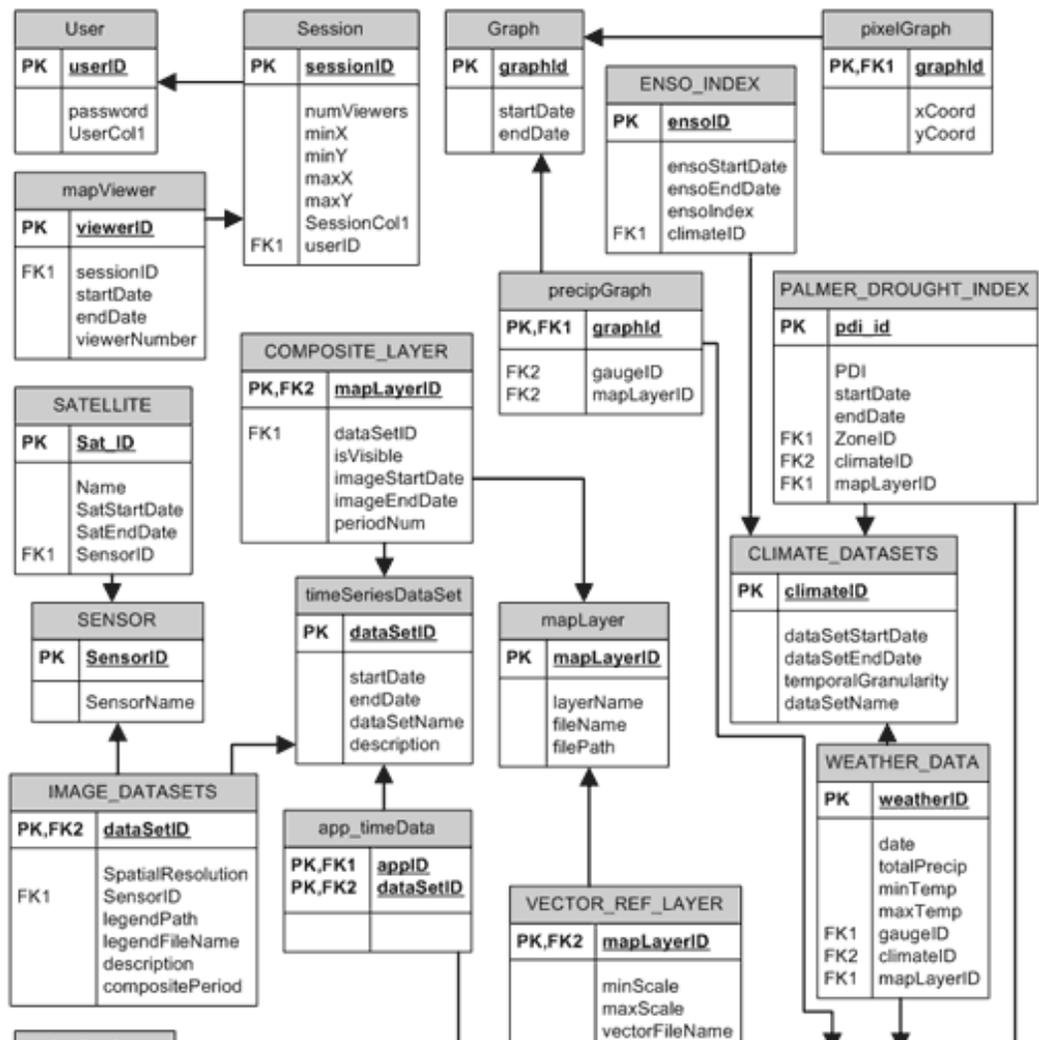


Figure 8. Entity Relationship Diagram (Ann-Maree White)

4.4.2. Logical Design

The goal of the logical design was to develop a good relational schema that could easily be converted into physical database tables. This step involved mapping the entity relationship model to a logical model, and eliminating data redundancy and data modification anomalies (Figure 9).



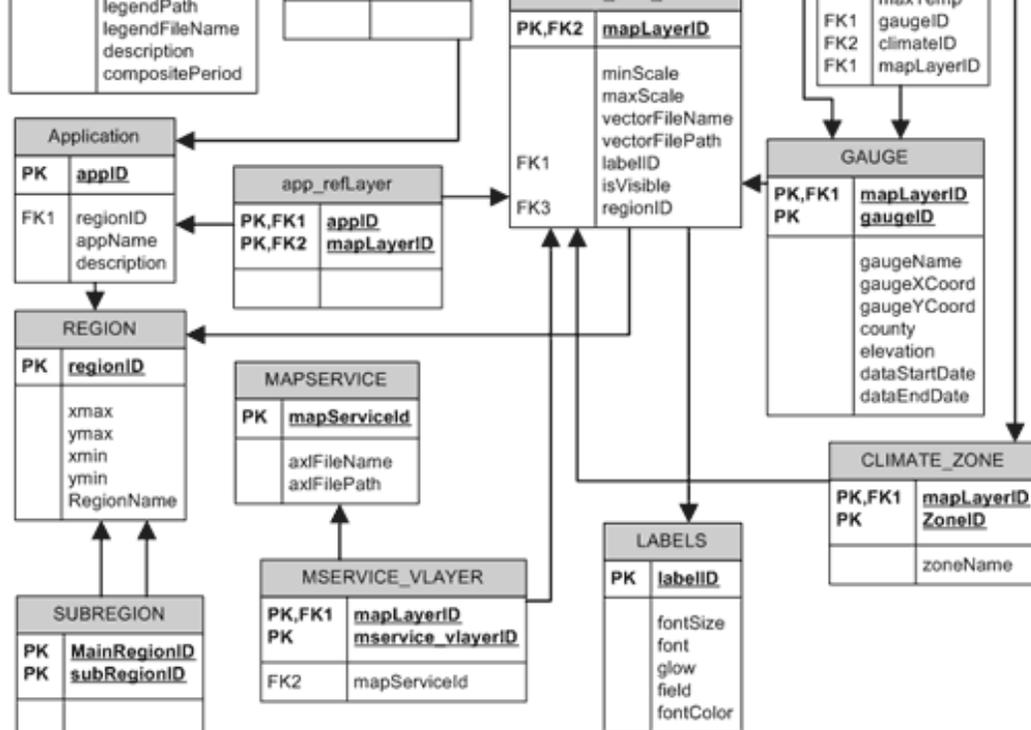


Figure 9. Relational Schema (Ann-Maree White)

5. IMPLEMENTATION

5.1. System Environment

5.1.1. Software

Our web, map, and database servers run on MS Windows 2000 Server while our image processing and maintenance workstation runs on MS Windows 2000 Pro. We use MS IIS 5 (processes HTML and ASP) and Apache Tomcat 4.1.12 with Sun Java 2 JDK SE 1.4.0 (processes HTML, JSP, and Java Servlets) as our web server. We use MS SQL Server 2000 Standard as our database server. We use ESRI ArcIMS 4.01 with the Java Connector as our map server and ESRI ArcSDE 8.3 for MS SQL Server 2000 to store our spatial data. The JFreeChart 0.9.8 library (<http://www.jfree.org/jfreechart/>) is used to render custom graphs and charts and the Java Web Services Developer Package 1.1 (<http://java.sun.com/webservices/>) by Sun Microsystems is used to parse XML and other maintenance functions. ESRI ArcINFO 8.3, EDAS Imagine 8.6, and the Modis Reprojection Tool (MRT) are used for image processing and maintenance.

5.1.2. Hardware

Our current hardware set-up is divided in a web/map server, a database server, and an image processing and maintenance workstation. The web and ESRI ArcIMS web server is a dual 1GHz Pentium 3 machine with 1GB of RAM, 2 18GB SCSI hard drives, and 1 120GB IDE hard drive. The database server is a dual 1.4GHz Xeon machine with 2GB of RAM and 8 120GB hard disks in a RAID 10 configuration. The image processing and maintenance workstation is a 1.26GHz Pentium 4 machine with 1GB of RAM and 300GB of storage space. Future hardware setups can include clustering of SQL server.

5.2. Software Development

5.2.1. Presentation Logic

The presentation layer (or view) serves as an interface between the user and the underlying application. The presentation logic is not only a graphical user interface, but acts as a conduit for the response and request cycles between the remote client, such as Netscape Navigator (NN), MS Intern Explorer (IE) and Mozilla, and the web server through HTML, XML, Java Server Pages (JSP), and Java Servlets. User input is captured and the requests are passed to the controller for processing. The results are passed from the controller to the presentation layer for display back to the user.

To meet the design goals (see [Section 4.3](#)), we decided to develop two web browser clients:

a) a thin JSP-based web client with minimal use of JavaScript. This version will be supported by most web browsers and versions (such as the relatively common NN 4.7) and b) a thicker JSP-based web client with extended use of JavaScript, W3C DOM, and Cascading Style Sheets (CSS). The thin JSP client has minimal user software and hardware requirements and is compliant with the Americans with Disabilities Act (see Section 508 site at <http://www.section508.gov>). However, it lacks many interface features such as a “rubber-band” zoom. The thick JSP client is currently under development. It has a minimum web browser requirement (NN 7, IE 6, and Mozilla 1.3), requires significant client side processing of the request response cycle, is not ADA compliant due to the use of IFRAMEs, and requires Remote Procedure Calls (RPC) through XML or JavaScript RPC (Costello and Apple Developer Connection 2002).

The presentation logic offers the greatest flexibility for customization. However, the implementation of a modular software design strategy is limited by the lack of uniform HTML, JavaScript, and W3C DOM implementation in available web browsers. Therefore, we are exploring alternatives such as the use of ActiveScript with the relatively widely available Macromedia Flash 6 plug-ins. In addition we are exploring ways of generating print quality custom maps in Adobe PDF file format.

5.2.2. Business Logic

The implementation of the business logic is currently in progress. It is the core of the application, and has the knowledge to handle and coordinate both process and data flow ([Figure 7](#)). The business logic layer is separated into three major sections: process flow, data flow and utilities. The system runs through Apache Tomcat, the Servlet engine, and most of the modules were implemented with pure Java. One exception is the pixel grabber, which uses a C program to communicate with ArcSDE. The following sections provide details about the components that have been implemented as well as proposals for the remaining sections.

5.2.2.1. Process Flow

The process flow components handle all of the requests from users and resulting responses. This layer serves to communicate with the presentation layer and coordinate with the data layer. The user requests come from the presentation layer, and are dispatched by the controller to a dedicated handler. The dispatch operation behaves differently according to the input variables and the current state of the client. Results are obtained and sent back to the presentation layer.

The Controller is a Java Servlet that handles user request from the presentation layer in the following flow: read form parameters sent by a user request; read current state from session attribute and Java Bean; dispatch request to the dedicated handler to process; and present the resulting information to the presentation layer. The controller has the knowledge of the whole process flow for the system, like a manufacturing plan or blueprint.

The Handler, on the other hand, has specific logic of a self-contained task. The primary flows in the handler are to read the input parameter, and determine the correct processing procedure based on internal or external states. If necessary, the handler will ask for resources, such as objects, from the data flow layer or map utility package.

The user registration function demonstrates the interaction between the presentation and data layers. In the presentation layer, when the user hits submit in the form, the system triggers an action and sends the form element to the controller. The Controller checks whether a user of the same name exists in the system. If not, it sends it to the Register Handler, which processes an insertion to the database by calling the insert method in the User Data Access Object (DAO). If the name does exist, the controller forwards it to an error page that informs the user that the name already exists. After successfully inserting a new user, the Controller forwards it to the next page to continue the process.

5.2.2.2. Data Flow

The data flow layer is responsible for the formulation of SQL queries and keeping track of session variables.

Java Beans are used as temporary storage containers and provide Set and Get methods for the handlers to access data values from the data layer. The Bean Object should stay consistent with form components in the presentation layer and attributes in the table. If changes are made to the database schema or the business logic, the Bean Object must also change. In order to provide this consistency, the Bean and DAO objects will be created automatically from the table schema and a configuration file. This allows developers to easily make changes and extend functionality.

DAO objects are used to provide data manipulation methods for the handler to communicate with entities in the database. For example, the Find method will send an SQL string to DBQuery in the Database Utility Package and set the attribute value of the related Bean Object. The Insert method will set attributes in the Bean Object as parameters and run a SQL command to perform Update and Delete methods.

5.2.2.3. Utilities

The Utility packages provide the mechanisms for interaction with the data layer and other external services, such as ArcIMS. Presently, the application contains a Database Utility and a Map Utility package. Both have a Manager component that acts as a coordinator for all of the available functions and tools. The Manager provides an interface for communication between the process flow and data flow layers and the underlying database server and ArcIMS Server.

Database Utility Package:

Table 3: Database Utility Package

Requirements	<ul style="list-style-type: none"> • J2SDK 1.3 or later version. • JDBC Driver for vendor's database • XML Parser
Features	<ul style="list-style-type: none"> • JDBC connection with pooling mechanism; • XML format configuration and SQL in property file that enhance maintenance capability; • Support multiple database types. • Pure Java-based package.
Major Components	DBManager, DBQuery, other Java files, etc.

Component Descriptions:

- The DBManager was designed by following the wrapper and singleton design patterns. It is responsible for coordinating invisible components to provide customized methods for database access by encapsulating complicated interior function calls. The connection pool mechanism takes place when the system is initialized. The advantage of the connection pool is that connection objects are stored in memory and reused without having to use system resources to request additional connections. The XML configuration and SQL property file allow developers or Database Administrators (DBA) to change a database instance or table schema without changing any code in the Java programs. These features facilitate easy updates and extensibility. In addition, multiple relational database management systems (RDBMS) can be used by setting driver information in the configuration file and loading the JDBC driver into the classpath environment.
- The DBQuery component stores the SQL property file into a property object and allows the DAO to run SQL strings. In the initialization, the DBQuery object acquires a connection object from DBManager. It supports Statement, PreparedStatement and CallableStatement from the Java.SQL package to communicate with database by general SQL or Stored Procedures.

Map Utility Package:

Table 4: Map Utility Package

Requirements	<ul style="list-style-type: none"> • J2SDK 1.3 or later version • JFreeChart, Java Connector API from ESRI • ArcSDE and ArcIMS server • XML Parser • Database Utility Package
Features	<ul style="list-style-type: none"> • Cluster framework and load balancing of ArcIMS map servers • Enhanced and integrated visualization tools • Pixel and index graphing through ArcSDE and business database
Major Components	MapManager, other Java files, etc.

Component Description:

- MapManager: The design concept is the same as DBManager. It provides customized map tools by encapsulating the ESRI ArcIMS Java Connector API. Various ArcIMS functions are made available through our customized API. If ESRI makes changes to the API, we can simply update our customized API, without having to track down every piece of code that conflicts with ESRI's changes. The only effort required is to change the references to the correct methods and provide the correct interface to the Handler.

Cluster and load balancing is used to enhance ArcIMS map server performance by implementing an elegant algorithm to provide map service from different hosts. Again, configuration of the host information is easy to update and expand.

In addition to the ArcIMS functions provided by the MapManager, precipitation and pixel graphing functions are also available in the Map Utilities package.

Precipitation Graphing:

The precipitation graphing functionality allows users to view a graph of precipitation for a chosen weather station while viewing the corresponding time series data maps for the surrounding area. The user chooses a time series dataset and a start and end date. The database is queried for time series data matching the user input, then queries for the corresponding precipitation values. If the time series data is composite data, then the daily precipitation values must be summed over the composite period. The precipitation values are put in an array and sent to JFreeChart to generate a graph. The graph in [Figure 10](#) demonstrates how JFreeChart can be used to graph precipitation through time.

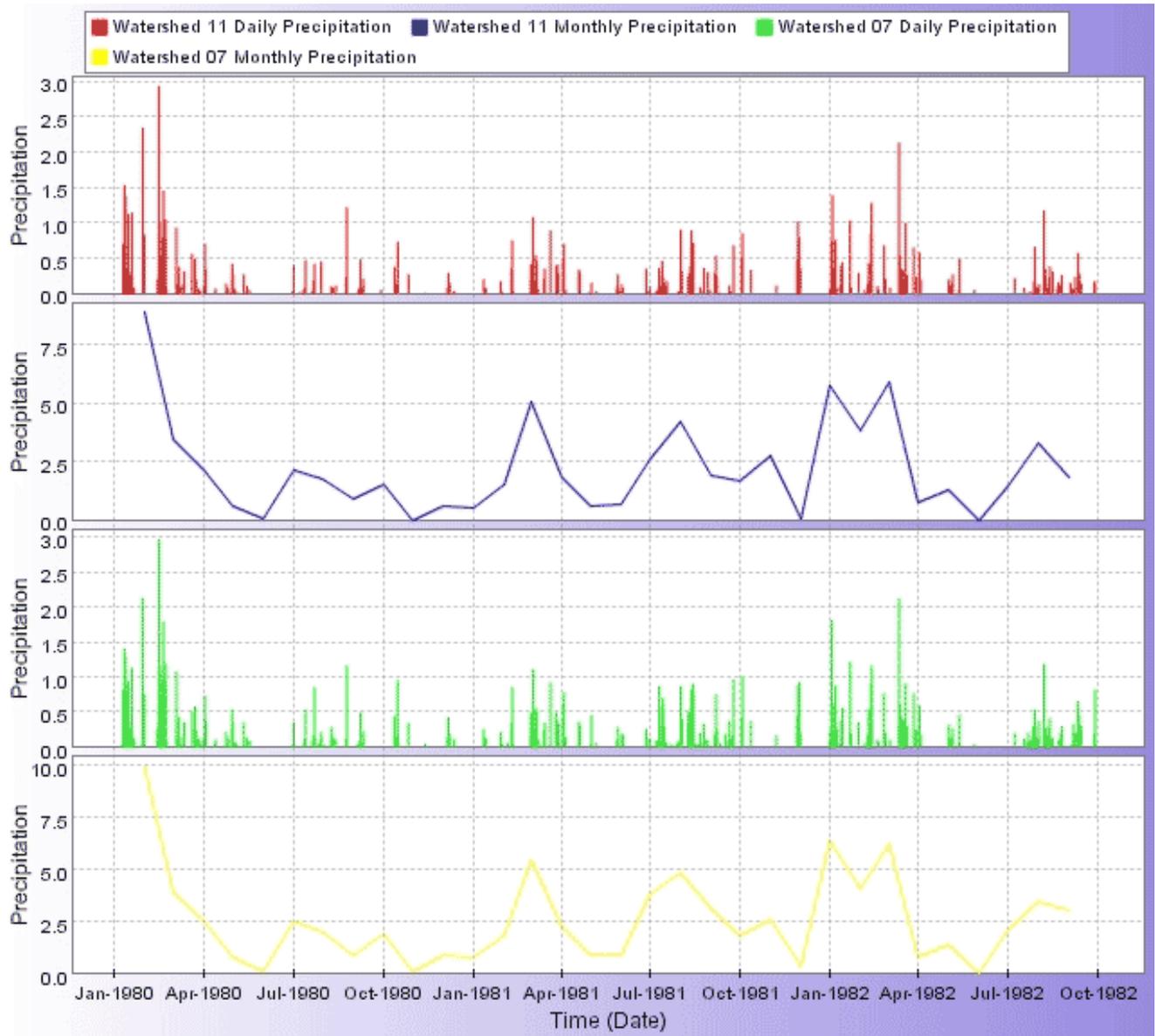


Figure 10. Example of precipitation graphs rendered through JFreeChart (Yu Yang).

PixelGrabber:

The PixelGrabber developed for RangeView project allows web applications to access individual pixel values in satellite imagery stored in an ArcSDE database. PixelGrabber offers similar functionality to the familiar ArcGIS IDENTIFY tool; however, PixelGrabber is a stand-alone Java class that can easily be implemented in any Java application. Currently PixelGrabber is used on the RangeView development website to allow web users to dynamically create graphs of raster data pixel values from time series datasets such as the vegetation indices derived from satellite imagery.

PixelGrabber is a Java wrapper class that implements code written in C using the Java Native Interface (JNI) and the ArcSDE C API. Java function calls in the PixelGrabber class access private C functions that communicate directly with the ArcSDE database and return pixel values back to the Java application.

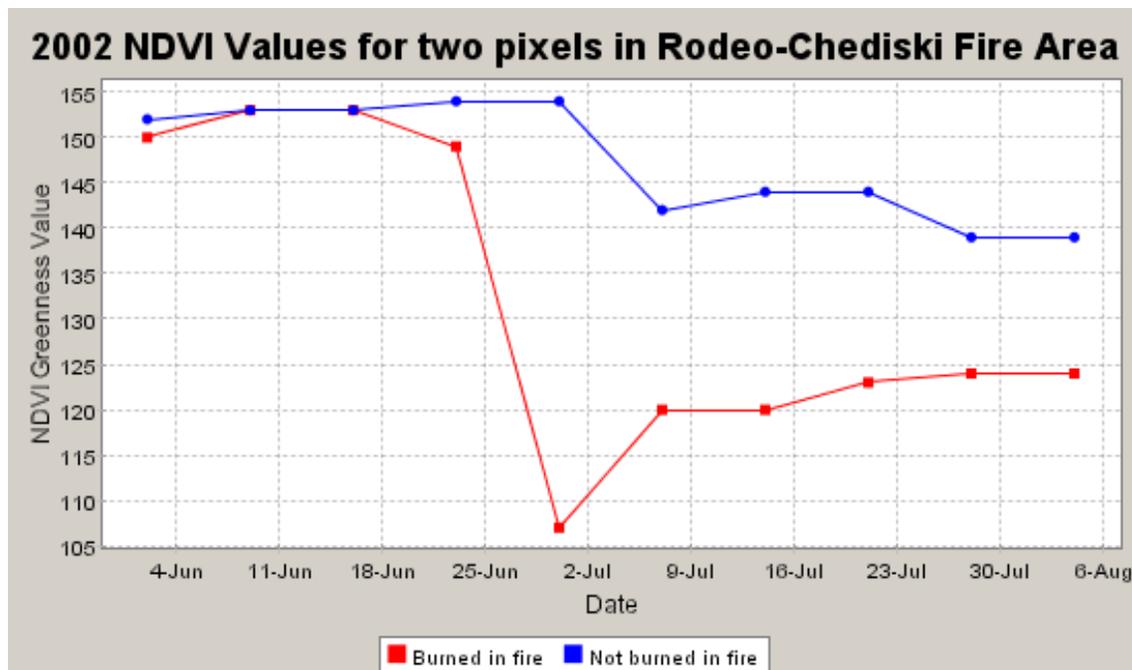


Figure 11. PixelGrabber results rendered through JFreeChart: AVHRR NDVI values for two pixels near the Rodeo-Chediski Fire of Summer, 2002, in northern Arizona. The red pixel was burned in the fire. The blue pixel was not burned, but shows a drop in NDVI due to smoke cover from the fire. See Figure 5 for NDVI images of the fire (Jonathan Saints).

5.3. Database Implementation and System Integration

5.3.1. Database Creation

Microsoft SQL Server 2000 is the RDBMS for this project. SQL Data Definition Language (DDL) was used to define the database tables, primary and foreign keys, and other constraints on the database. Referential integrity rules are enforced in the database, maintaining all logical relationships between tables when data are added, modified or deleted. Referential integrity is enforced through the use of primary and foreign keys. Tuples cannot be added with duplicate values. The value of the foreign key column must correspond to the value of the primary key column of the referenced table and action is taken to ensure that if a record is deleted with foreign key references, those records will either be deleted or set to null.

The time series image data is stored in ArcSDE databases. We created separate ArcSDE databases for each satellite sensor (MODIS, AVHRR, etc.) in SQL Server 2000. Separating the ArcSDE databases improves its performance at the cost having to use more client access licenses for the increased number of ArcSDE and database connections. Pyramid layers were used to increase performance. Each image in the ArcSDE database is referenced in the business database along with metadata such as start and end dates.

5.3.2. Database Implementation

In the implementation of the 3-tier client server architecture, the database server is in the bottom tier. To the database server, client-side programs, such as applets or applications, and server-side programs, such as the servlets, are all SQL clients. A database utility component is provided in the business layer of this application for access to clients that connect to the database server during execution. The middleware component acts as a server to upper layers and as brokers for, and clients to, the database server. Java Database Connectivity (JDBC) is used to connect to the database, create and execute SQL statements, and return result sets to the clients.

5.3.2.1. Optimization

The web applications rely heavily on dynamic information supplied by the RDBMS. Because of this, there will potentially be a large number of database hits that will grow with an increasing user base. The middleware database utility uses connection pooling to decrease overhead and increase performance.

When the application is launched, the database will primarily be used for reading data. Database updates will be done on a regular schedule to add new data as it is processed ([Section 5.4](#)). Although the database will be effectively read-only, many of the queries will be processing intensive, and demand on the database will grow with the number of users. Indexes and query optimization will be used to improve performance.

Stored procedures and prepared statements will also be used to increase performance of processing intensive queries. For example, a stored procedure is used to calculate the precipitation values that correspond to every image in a result set. The precipitation data are stored as daily values, which must be summed over the time period of interest. In order to graph precipitation values to match a set of images through time, a query must be performed that finds all images that fall within the user-selected time period, gets the start and end dates for the composite periods for each image, and sums the daily precipitation values over each composite period. A stored procedure is used to minimize the time required for this processing-intensive query.

5.4. Maintenance

This section covers components that are external to the three-tier architecture but are essential to it ([Figure 7](#)). The Map Server is comprised of ESRI's ArcIMS. Separate ArcIMS AXL files are used for each dataset to reduce the number of vector and raster layers per map service to hundreds instead of thousands of layers. ESRI ArcSDE is used as a middleware to load, manage, and retrieve raster and vector data from SQL Server 2000.

To keep the application up-to-date, we run a weekly semi-automatic maintenance plan ([Figure 12](#)). That plan currently includes the downloading of new satellite imagery, processing of the same, loading the images into ArcSDE, updating the business database, updating the AXL files, and restarting the ArcIMS map services.

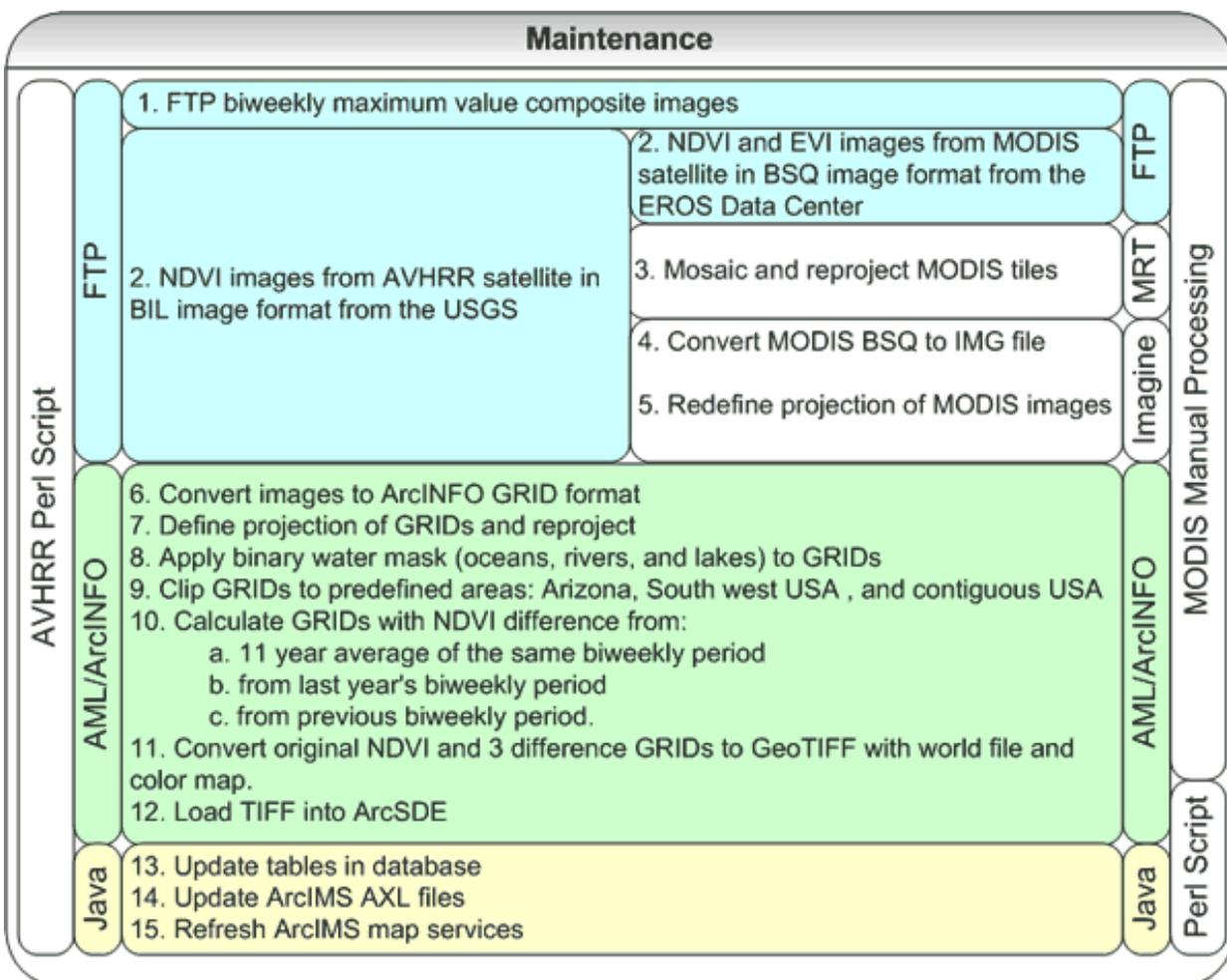


Figure 12. Maintenance plan for AVHRR and MODIS images (Kalyani Mandapaka, Laura Baker, and Stefanie Herrmann).

6. CONCLUSION

Outside of the research realm, data collected on a continual basis over time and over larger areas have been difficult to analyze, pose data integration challenges, and are often too complex to visualize in a simple but meaningful way. Nevertheless, the demand is high for such data to be made available on an accessible, operational, and timely basis. This is particularly true for natural resource managers who have long expressed the desire to take advantage of EOS satellite imagery collected at frequent intervals along with other data (e.g. climate indices, precipitation, wildfire occurrences, etc.) that coincide with the time of image acquisition.

This paper outlines the University of Arizona's attempt at meeting this challenge, both from an application development perspective and from the perspective of technology transfer and its relationship to understanding user requirements. While the Internet is an ideal medium to disseminate spatially variable time series data, the design of any application to accommodate the needs of end users must involve their continual input. Moreover, the corresponding systems architecture and business logic must accommodate a level of flexibility and agility that matches the pace of new or revised demands that come from those same end users.

The utility of the final application has been enhanced by laying a foundation for integrating disparate data sources and allowing the user the ability to select from a variety of datasets, zoom in on an area of interest, visualize time series animations of spatially explicit data, analyze cell values of those data through time by selecting individual or groups of pixels, and graph time series data.

The role of the user was made more central to the research and development process by creating a two-way knowledge exchange where stakeholder needs and feedback on intermediate products flows one way and education about the data and the information to be had from value-added products flows the other. The integration of user interaction and systems design has resulted in an application that we hope can provide sustained access to multitemporal spatial and tabular data to support resource management applications today and in the future.

7. ACKNOWLEDGMENTS

Two projects underlie the development outlined in this paper: a NASA/Raytheon sponsored Infomart called RangeView (<http://rangeview.arizona.edu>) and an EPA Star Grant called Wildfire Alternatives - WALTER (<http://walter.arizona.edu>).

Within the larger NASA/Raytheon Synergy initiative, prototype Infomarts are being implemented to demonstrate how Earth Observing System (EOS) satellite imagery and other commercial data can be used to address real-world problems confronting users, particularly in state, local and tribal government agencies. The EOS Data and Information System (EOSDIS) Synergy project is intended to enhance the efforts of NASA and university researchers around the United States who are investigating potential applications of remote sensing data. NASA sponsors these efforts under a contract with Raytheon, the prime contractor for the development of the EOSDIS Core System.

An EPA STAR Grant funded initiative called "Wildfire Alternatives" (WALTER) has set out to maximize advances in geospatial analytical, as well as web delivery technology, to improve our understanding of and provide access to applied research on the relationship between wildfire, climate, and society (Grant #R-82873201-0; Morehouse et al. 2000).

We thank the ARSC development team at the University of Arizona for their efforts in support of the RangeView project: Dr. Stuart Marsh (Principal Investigator), Laura Baker, Wolfgang Grunberg, Chih-Lung Kuo, Kalyani Mandapaka, Dr. Barron Orr, Jon Saints, Anne Thwaites, Ann-Maree White, and Yu Yang. We would also like to thank our RangeView collaborators who have provided critical insight on rangeland science and management (Drs. George Ruyle and Larry Howrey), wildlife biology (Dr. Paul Krausman), remote sensing and data continuity (Dr. Wim van Leeuwen and Stefanie Herrmann), information science (Barbara Hutchinson), and marketing research (Dr. Kapil Jain and Chris Baker of Marketing Intelligence, LLC.)

We also thank the entire WALTER team at the University of Arizona, led by Dr. Barbara Morehouse, Dr. Jonathan Overpeck, Dr. Thomas Swetnam, Dr. Steve Yool, Dr. Gary Christopherson and Dr. Barron Orr, with web design and development conducted by Amanda Cockerham, Wolfgang Grunberg, Heather Severson, and Anne Thwaites.

We specifically would like to thank Laura Baker, Kalyani Mandapaka, Heather Severson, Anne Thwaites, and Yu Yang who provided critical content and carefully reviewed this paper.

8. CITATIONS

Apache Software Foundation. Tomcat. Online: <http://jakarta.apache.org/tomcat/index.html>. Accessed June 2003.

ARSC. 2002. NASA Regional Workshops for State, Local and Tribal Governments: Findings and Implications. Tucson: Arizona Remote Sensing Center. Online: <http://aria.arizona.edu/report.pdf>. Accessed June 2003.

Costello, E. and Apple Developer Connection. 2002. Remote Scripting with IFRAME. Online: <http://www.oreillynet.com/pub/a/javascript/2002/02/08/iframe.html> O'Reilly & Associates. Accessed June 2003.

Date, C.J. 2000. An Introduction to Database Systems, Seventh Edition. Reading, Massachusetts: Addison-Wesley Publishing Inc. 938 pp.

Eckerson, Wayne W. 1995. Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications. Open Information Systems 10, 1 (January 1995): 3(20).

ESRI. ArcIMS, ArcINFO, and ArcSDE. Online: <http://www.esri.com>. Accessed June 2003.

Gamma, E., J.Vlissides, R. Johnson, and R. Helm. 1994. Design Patterns: Elements of Reusable Object Oriented Software. Reading, Massachusetts: Addison-Wesley Publishing Inc.

Garfin, G., and B. Morehouse. 2001. Fire and Climate Workshop Proceedings, February 14-16 and March 28, 2001. Tucson, (AZ): CLIMAS-ISPE, University of Arizona.

Jain, K., C. Baker, and A. Rusu. 2002. Synergy II InfoMart Marketing Plan. Tucson, (AZ): Marketing Intelligence, LLC.

Jain, K., J. Klewer, and C. Baker. 2001. Synergy II InfoMart Marketing Plan. Tucson, (AZ): Marketing Intelligence, LLC.

JFreeChart. Online: <http://www.jfree.org/jfreechart/>. Simba Management Limited. Accessed June 2003.

Marsh, S.E., B. Orr, A-M White, L. Baker, K. Mauz, C. MacDonald, G. Oldham, H. Rodriguez, C. Wallace, B.S. Hutchinson, A. Thwaites, L. Howrey, G.B. Ruyle, P.R. Krausman, A. Heydlauff, D. Schafer, E. Trobia, J. DeVos. 2001. Final Report: An Infomart For Natural Resources: EOS Science Results And Data For Public Land Management In The Western States. Tucson (AZ): University of Arizona.

Marsh, S.E., B.J. Orr, W. van Leeuwen, A-M. White, C.F. Hutchinson, L. Baker, M. Farah, W. Grünberg, S. Herrmann, C-Y. Huang, C. McDonald, G. Oldham, H. Rodriguez, C. Wallace, B.S. Hutchinson, A. Thwaites, L.D. Howery, G.B. Ruyle, P. R. Krausman, A. Heydlauff, D. Schafer, K. Jain, and C. Baker. 2002. Final Report Synergy III: Extending and Sustaining Infomart Tools and Data for Natural Resource Management. Tucson (AZ): Arizona Remote Sensing Center.

Moore, G.A. 1999. Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers (revised edition). New York: HarperCollins.

Moran, M.S., Y. Inoue, and E.M. Barnes. 1997. Opportunities and limitations for image-based remote sensing in precision crop management. Remote Sensing of Environment 61:319-346.

Morehouse, B.J., T.W. Swetnam, J.T. Overpeck, S.R. Yool, G.L. Christopherson, and B.J. Orr. 2000. Assessing the consequences of interactions between human activities and a changing climate. Proposal submitted to the Environmental Protection Agency, 2000-STAR-F1, April 25, 2000.

Naisbitt, J. 1982. Megatrends: Ten New Directions Transforming Our Lives. New York: Warner Books.

NASA. MODIS Land Discipline Web Site. Online: <http://modis-land.gsfc.nasa.gov>. Accessed June 2003.

NOAA National Climatic Data Center. NOAA-NCDC Climate Data Online. Online: <http://cdo.ncdc.noaa.gov/plclimprod/plsql/poemain.poe>. Accessed June 2003.

NOAA-CIRES Climate Diagnostics Center. Multivariate ENSO Index (MEI). Online: <http://www.cdc.noaa.gov/~kew/MEI/mei.html>. Accessed June 2003.

Orr, B.J., M. Hertzfeld, and A. Thwaites. 2002. Geospatial Extension Program in Arizona. This is the main homepage for Arizona's "Earth Grant" Geospatial Extension Program. Online: <http://arsc.arid.arizona.edu/extension/>. Accessed June 2003.

Rob, P. and C. Coronel. 1999. Database Systems: Design, Implementation, and Management, Fourth Edition. Thomson Learning, 776 pp.

Rob, P., and C. Coronel. 2000. Database Systems: Design, Implementation, and Management, Fourth Edition. Course Technology.

Ryan, B. and N.C. Gross. 1943. The diffusion of hybrid seed corn in two Iowa communities. Rural Sociology 8(1-4) 15-24.

Sadoski, D. and S. Comella-Dorda. 1997. Three Tier Software Architectures. Online: http://www.sei.cmu.edu/str/descriptions/threetier_body.html. Carnegie Mellon University. Accessed June 2003.

Schussel, G.. 1995. Client/Server Past, Present, and Future. Online: <http://news.dci.com/geos/dbsejava.htm>.

Singh, I., B. Stearns, M. Johnson, and the Enterprise Team. Designing Enterprise Applications with the J2EETM Platform, Second Edition. Online: http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/DEA2eTOC.html. Sun Microsystems. Accessed June 2003.

Sun Microsystems. Java Web Services Developer Pack. Online: <http://java.sun.com/webservices/>. Accessed June 2003.

University of Arizona. 2002. ARIA: Arizona Regional Image Archive. Online: <http://aria.arizona.edu>. Tucson (AZ): University of Arizona. Accessed June 2003.

University of Arizona. 2002. RangeView: Geospatial Tools for Natural Resource Management. Online: <http://rangeview.arizona.edu>. Tucson (AZ): University of Arizona. Accessed June 2003.

University of Arizona. 2002. WALTER: Exploring Wildfire Alternatives. Online: <http://walter.arizona.edu>. Tucson (AZ): University of Arizona. Accessed June 2003.

APPENDIX A - List of Acronyms

API	Application Program[ing] Interface
ARSC	Arizona Remote Sensing Center
AVHRR	Advanced Very High Resolution Radiometer
CSS	Cascading Style Sheets
DAAC	Distributed Active Archive Center
DAO	Data Access Object
DBA	Data Base Administrator
DDL	Data Definition Language
DOM	Document Object Model
DOQQ	Digital Ortho Photo Quarter Quadrangles
ENSO	El Niño Southern Oscillation
EOS	Earth Observing System
EOSDIS	EOS Data and Information System
EPA	U.S. Environmental Protection Agency
ERD	Entity Relationship Diagram
EVI	Enhanced Vegetation Index
GIS	Geographic Information System
IE	Internet Explorer
JDBC	Java Database Connectivity
JSP	Java Server Pages
LAI	Leaf Area Index
MEI	Multivariate ENSO Index
MI	Marketing Intelligence, LLC
MODIS	Moderate Resolution Imaging Spectroradiometer
MRT	MODIS Reprojection Tool
MVC	Model View Controller
NASA	National Aeronautics and Space Administration
NDVI	Normalized Difference Vegetation Index
NN	Netscape Navigator
NOAA	National Oceanic & Atmospheric Administration
PDSI	Palmer Drought Severity Index
RDBMS	Relational Data Base Management System
RPC	Remote Procedure Call
UA	The University of Arizona
USGS	U.S. Geological Survey
WALTER	Wildfire Alternatives

AUTHOR INFORMATION

Wolfgang Grunberg

Research Specialist

[Center for Applied Spatial Analysis](#) and [Arizona Remote Sensing Center](#)

[Office of Arid Lands Studies](#)

[The University of Arizona](#)

1955 E. 6th Street, Suite 205

Tucson, AZ 85719

E-mail: grunberg@email.arizona.edu

Phone: (520) 621-5694

Fax: (520) 621-3816

Barron J. Orr

Assistant Professor and [Geospatial Extension Specialist](#)

[Office of Arid Lands Studies](#) -- [Arizona Remote Sensing Center](#)

[University of Arizona Cooperative Extension](#)

[Arizona Space Grant Consortium](#)

1955 E. 6th Street, Suite 205

Tucson, AZ 85719

E-mail: barron@ag.arizona.edu

Ann-Maree White

Research Specialist

[Arizona Remote Sensing Center](#)

[Office of Arid Lands Studies](#)

[The University of Arizona](#)

1955 E. 6th Street, Suite 205

Tucson, AZ 85719

Chih-Lung Kuo

Research Assistant

MIS Department

[The University of Arizona](#)

E-mail: chihlung@email.arizona.edu

Jonathan Saints

Research Assistant

[Arizona Remote Sensing Center](#)

[Office of Arid Lands Studies](#)

[The University of Arizona](#)

1955 E. 6th Street, Suite 205

Tucson, AZ 85719