An Object Oriented Framework for Integration of Agricultural and Natural Resource Management Tools

S. N. Collis and J. D. Corbett

Mud Springs Geographers Inc. Texas, USA (stewartcollis@mudsprings.com).

Abstract: Delivering information to decision makers is often a bottleneck in turning research results into effective decisions and policies. Information technology in agriculture and natural resource management is maturing to provide more sophisticated tools for analyzing, visualizing, managing, and disseminating information. Successful integration and dissemination is dependent on creating flexible and scalable software frameworks that provide both complex analysis tools for advanced users and deliver information to a wider audience. Most natural resource and agricultural research studies are linked to a specific geographic location (e.g., GPS coordinate) or region (an agro-ecological zone or a political region). Thus using Geographic Information Systems (GIS) technology is a logical basis for a Spatial Decision Support System (SDSS) to store and manage information for any research project, model, or study in a systematic way that is both useful to researchers and accessible to those other than researchers or technicians. AWhere™-ACT represents an implementation of an SDSS providing effective analysis and information delivery tools to decision makers who are not GIS specialists. Using object oriented methodologies and Component Object Model (COM) technology AWhere™-ACT provides a scalable framework for efficient integration with other systems (for example, crop simulation models, epidemiological models and weather generators) and databases (such as geo-referenced documents and remotely sensed time series data). The AWhere™-ACT components (as can all COM compliant objects) can be reassembled into specific applications to meet tailored needs or can be incorporated piecemeal into wholly separate applications. This versatility and flexibility without intellectual property compromise, offers a highly effective and efficient mechanism to share scientific advances and contribute to decision makers information synthesis needs.

Keywords: Spatial; Framework; Decision Support; GIS; Object Oriented

1. INTRODUCTION

Effective policy decisions require a wide variety of up to date and accurate information, hence the development of Decision Support Systems (DSS). One challenge in providing an effective decision support system is to summarize and synthesize the wide variety and often voluminous information available to the decision maker into a form that can be easily accessed and utilized in a practical timeframe. In recent years, such systems have taken the form of computer software, thus enabling great advances in information integration.

The problems faced by natural resource managers require data, information, and analysis tools that are both spatial and aspatial [Lueng, 1997]. One common thread that links most research, projects, surveys and other information sources is their spatial domain. Whether the information source is an economic analysis at the regional level, a household survey at the district level or a crop trial at the field level, they all can be related using their geographic location. Regardless of the spatial scale and the research theme, this common thread, defined through geographic coordinates, allows us to manage diverse information sources within a common framework enabling innovative analysis and insightful visualizations.

Geographic Information Systems (GIS) are an established technology in the environmental management and research arena. However, adoption by non-specialists has not been widespread. The software system AWhere™-ACT[1] brings GIS a step closer to the decision maker whilst providing a sophisticated underlying structure that experts can use for modeling purposes. AWhere™-ACT is a spatial decision support system built using object oriented techniques to manage, analyse, visualise and disseminate information to a broad range of users. The stand alone software framework for AWhere™-ACT contains data specific modules, which utilize, where appropriate, the same

[1] ACT is an acronym for Almanac Characterization Tool, see acknowledgements.

underlying object oriented structure. This approach has proven to be a successful implementation of an information delivery tool that can help users make better policy decisions related to agriculture and natural resource management [Corbett et. al., 2001].

2. FOUNDATION DATABASES

One underlying principle in promoting effective use of any spatial information system is the concept of a *foundation database.* There are volumes of data available to decision makers from various sources, many of which are available via the internet for free or at minimal cost. This however is of little use to a non technician who neither has the time to track down such data sources or the expertise to reformat, re-project and load them into appropriate spatial information system tools. As such, an effective spatial decision support tool must come with pre-packaged data that allows at least initial analyses to be done, with no need to seek external data sources. As a user becomes more proficient, or requires more detailed information, the ability to add custom data sets must also be provided. Simply providing data management tools will not address the need of non GIS specialists. AWhere™-ACT comes packaged with extensive databases, and each layer is thoroughly documented with metadata if further information is required. Databases delivered using AWhere™-ACT have predominately been eastern and southern African countries including Kenya, Ethiopia, Zimbabwe and Botswana and more recently parts of the United States and Mexico. The thematic groupings of data layers are diverse and include climatic, infrastructure, ecological, topographic, demographic and hydrographic layers. To manage the diverse nature of these layers an object oriented data model was developed as outlined in section 3. This design is key to delivering such a variety of information in a usable form.

3. OBJECT ORIENTED DEVELOPMENT

Object Oriented Programming (OOP) has revolutionized the way computer software is conceived, written and maintained. Object oriented simply means looking at a problem in terms of the objects involved with that problem [Kurata, 1999]. Object oriented principles enable software to not only model the real world more closely, but enables computer code to be written in a more organized and sustainable fashion. Object oriented models or "object models" can provide intuitive models that closely represent our logical models of the real world [Zeiler, 1999].

3.1. Data Models

The AWhere™-ACT data model is the foundation on which the suite of AWhere™ applications and components are based. This data model (Figure 1) consists of hierarchical objects that represent the spatial, temporal and thematic aspects of a data set. The top level object for the data model is the *Environment* class which represents a database or an area of interest such as the country boundary of Kenya or the rice growing region of the United States. The spatial extent or boundary of an Environment is decided by the user creating the database and adding data to it. The next level in the hierarchy, *Element,* is an abstract class representing a thematic grouping of the data such as Climatic, Edaphic or Demographic. User defined Elements may also be added. Contained within each Element is a collection of *DataLayers.* A DataLayer describes the spatial component of a data set such as locations of all weather stations in Kenya, or all the counties in Florida, or a remotely sensed image of Sydney, Australia. In the case of vector data, each feature (point, line or polygon) within a DataLayer can be associated with a data value or collection of values in a variety of ways. To define this association distinctly the DataLayer object has a collection of *DataClasses,* this is the next level in the hierarchy. Each DataClass describes the type of information contained within a DataLayer including the connection type (e.g. to a flat file or a relational database) the data type (e.g. precipitation, or a population density) or the units (e.g. mm, meters, inches, miles, persons\sq km). Lastly within each DataClass is a collection of *Variables* which all have the same data type and usually represent a time series of like values (Figure 1).

To demonstrate an implementation of this model examples 1 and 2 represent similar data sets implemented in the AWhere™-ACT data model with very different database structures. Firstly the two DataLayers represent different spatial features. Example 1 is a grid of continuous square cells and example 2 is a layer of discontinuous points. Secondly the underlying data structures are also different. Example 1 has a table for which each row is associated to a single spatial feature by the FeatureId (unique spatial feature identification). The DataClass Variables collection is simply a subset of the fields stored in the table. The Variables representing the months January through February are fields in this table.

Example 1.
Environment = Kenya
Element = Climatic
DataLayer = Annual 3 arc Minute Grid
DataClass = Monthly Mean Precipitation
Variable1 = January
Variable2 = February

Example 2.
Environment = Kenya
Element = Climatic
DataLayer = WMO Meteorological Stations
DataClass = Daily Mean Maximum Temperature
Variable1 = April 1st 1936
Variable2 = April 2nd 1936

In example 2 the majority of the underlying data is stored in a relational database. For a single weather station the unique FeatureId is linked to the relational database to extract the defined variables which are physically a series of rows in a database table (as opposed to a series of fields in a table as in Example 1) with the name of the table being the FeatureId value. This example demonstrates the power of object oriented programming techniques because even though data is stored in different ways, the components and software created using this data model do not differentiate between them. This object oriented approach has implications for program development efficiency i.e. changes made to one part of an application will not effect code in other parts of the application. In addition, tools written using this data model for specific data sets are easily applied to other data sets. A Growing Degree Day calculation written for a long term normal mean data set can be extended to long term historical weather data without any changes or adaptations needed. There are also advantages for end users. Novice users can view the data model properties through a pre-defined interface such as the treeview shown in Figure 2. Each object is represented and many of the associated properties can be viewed, such as the literal description of the data rather than the often cryptic field names often associated with such data sets thus shielding the user from unnecessary complexity. Other interfaces are designed to address the most commonly requested queries without the user dealing with the underlying complexities. Similarly advanced users can access these objects programmatically through a scripting interface to do advanced level modeling, but do not have to deal with differing data access scenarios. The technology that enables such components to be built around an object oriented data model in a modular fashion is Microsoft's Component Object Model (COM).

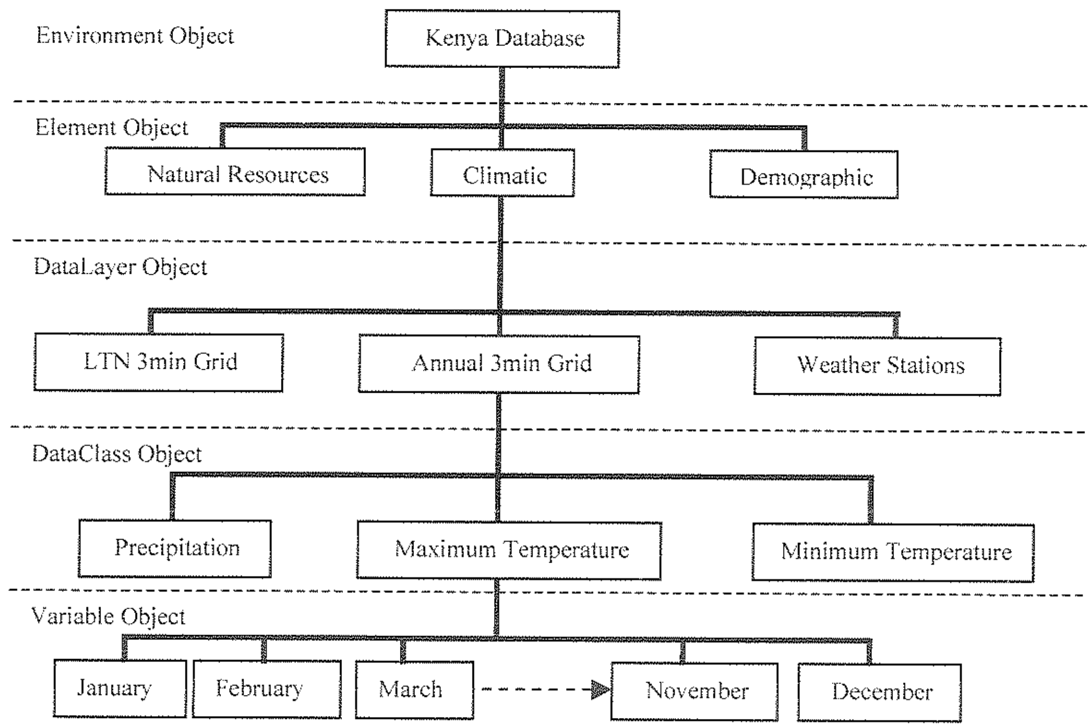

Figure 1. AWhere™-ACT data model with sample data.

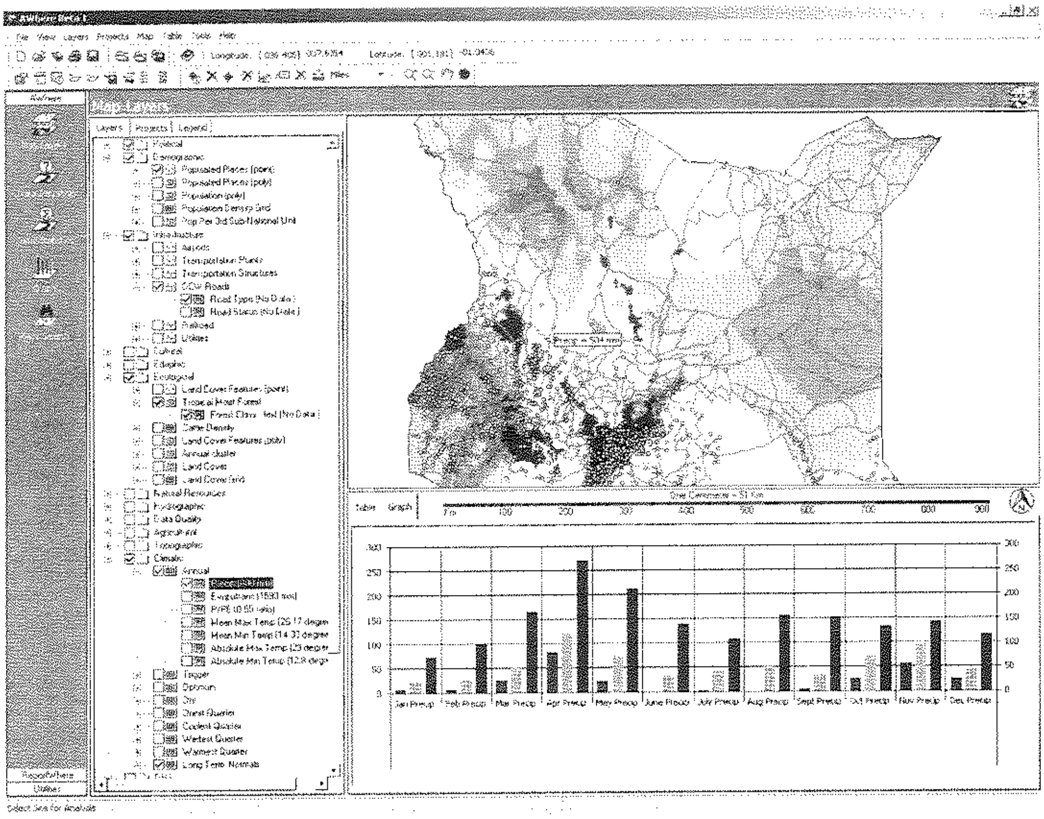

Figure 2. AWhere™-ACT interface.

3.2. Component Object Model (COM)

In the Microsoft Windows world, objects and components form the foundation on which everything else rests. The mortar for this foundation is Microsoft's Component Object Model (COM) specification and implementation [Vogel, 2000]. COM is an object-oriented programming model that defines how objects interact within a single application or between applications. In COM, client software accesses an object through a pointer to an interface—or a related set of functions, called methods—on the object [Microsoft, 2000]. COM components can be written in a variety of programming languages such as Visual Basic, Visual Delphi, Visual Fox Pro and Visual C++. ActiveX™ is a set of technologies built on COM that enable software components, regardless of the language in which they were created, to work together in a stand alone or networked environment. ActiveX™ controls are reusable, stand-alone software components that often expose a subset of the total functionality of a product or application. The exposed functionality of these controls is defined by the author, the underlying code is not available or accessible to other programmers using the control, only the functions and properties the author intends to be available can be accessed. This technology enables great efficiencies in software development. Programmers may integrate existing components (e.g., a graph control) into their applications instead of building them from scratch, thus saving an enormous amount of work. There are thousands of robust, commercial COM controls available, and AWhere™-ACT takes full advantage of these (e.g., map, graph and toolbars).

Most COM development tools (Visual Basic, Visual C++ etc.) also allow custom building of COM controls. Hence the flexibility of sharing certain components of an application with other developers is open to all developers. For example, AWhere™-ACT consists of a suite of custom built controls that other developers can access and incorporate into their applications. The ReportWhere module of the AWhere™-ACT suite is a stand alone control that could be compiled as a separate application or integrated into other applications. There is license protection on these components, so the original developers can maintain control and knowledge of who is using their tools (an asset for version control). COM technology opens up enormous cost efficiencies for software development and opportunities for truly integrated collaboration and is the basis for the development of AWhere™-ACT Shell

3.3. Three Tier Architecture

In order to take advantage of the benefits object oriented programming and COM technology offers AWhere™-ACT uses three tier architecture. Three tiered architecture is an industry standard that provides a framework for logical components of software to interact and enables flexibility in managing changes and updates in system components. The three tiers consist of the database layer, business layer and presentation layer or *client*. A major advantage of this approach is that if the rules (and subsequently the code) of one tier ever change, the programmer need only modify or replace that layer; there is no need to migrate the changes to the other layers [Sarang, 1999]. At all three levels, object models underlie the components at each layer to help organize code and provide structured models with methods and properties to allow communication between layers and to other applications. Where appropriate the AWhere™-ACT Shell modules are built on a common data model (e.g. AWhere, CensusWhere and SoilWhere share a common data model).

4. A DELIVERY FRAMEWORK

Essential to the creation of a decision support system that will ultimately have an impact is the framework used to deliver the tools to end users. This framework must provide a simple interface to end users but have the potential to provide for the needs of more advanced users. The AWhere™-ACT Shell addresses these issues by utilizing COM technology to package applications and data models in a user friendly graphical user interface (GUI) (client or presentation layer) and provides advanced level access to the underlying components and objects (business and database layers).

4.1. User Interface

To reach a largely non-technical audience the AWhere™-ACT client interface is modeled on the Microsoft Outlook™ interface for familiarity and function (Figure 2). The listbar on the left contains a set of "bars" that represent a separate COM applications or modules. Within each bar are a collection of items that refer to a function within each module. This design has the advantage of delivering a variety of applications to end users with a consistent interface. Interface design considerations are essential for application and information delivery and end user acceptance. There are also significant advantages to using a COM software framework to deliver multiple applications that relate to the interaction between these modules.

4.2. AWhere™ Shell

Using COM the many common features of the individual AWhere™-ACT applications such as toolbars, listbars, and map controls are handled by the AWhere™ Shell (Figure 3). This allows efficiencies in memory and implementation but also provides a common interface design. More importantly, the object models and data models underlying each individual application can interact with one another. This allows new modules to take advantage of the work previously done in existing modules and leads to faster development. This is an enormous advantage for developers who have the capacity to develop custom modules or stand alone applications using the AWhere™-ACT components.

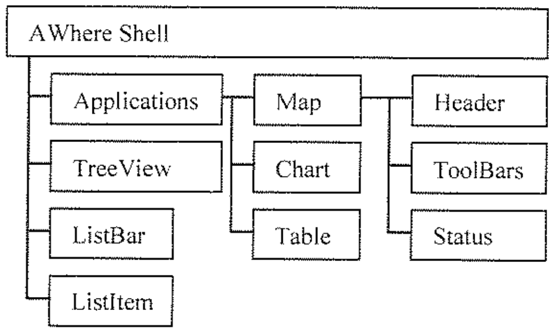

Figure 3. AWhere™-ACT Shell Object Model.

4.3. A Modeling Interface

For users who do not have the capacity or time to develop custom COM based modules for the AWhere™ Shell there is an intermediate yet powerful option available. An interface to the underlying object models of the applications delivered in the shell is provided by incorporating Visual Basic for Applications (VBA) in the shell interface. This allows programmatic access to the shell's object models in much the same way Microsoft Access or Microsoft Word can be customized by a user with some programming skills. This offers a powerful tool that allows scientists and researchers to concentrate on developing their own temporal models rather than developing data management tools and data models.

An example application of this object oriented framework would be to create a temporal model for climate to describe the growing the season. The layer shown in Figure 2 is a color gradation of a spatially interpolated surface of the long term monthly precipitation of Kenya (i.e 12 months). Surfaces have also been generated for long term normal monthly minimum and maximum temperatures. Using a VBA interface to the AWhere™-ACT data model a dynamic model can be constructed to generate a new layer describing the optimum five month growing season i.e. where moisture is maximized. This layer can be characterized for specific temperature and moisture attributes thus defining the potential spatial extent of crop varieties with such requirements. Using these tools complex models can be developed by advanced end users and the results delivered in a form acceptable to decision makers.

5. CONCLUSION AND DISCUSSION

"An intelligent spatial decision support system should possess the capability of handling interactively and effectively knowledge (structured and unstructured), information (spatial and aspatial), and human-machine interaction" [Lueng, 1997]. AWhere™-ACT demonstrates a tool that integrates GIS functionality and aspatial tools and applications. AWhere™ Shell is an object oriented framework for integrating independently developed applications and providing a common user friendly interface that can be customized to meet user's needs. Lueng [1997] also points out that "A SDSS without knowledge is a tool of no intelligence and minimal usage. The success of a SDSS in general and knowledge-based spatial information system in particular lies on it's level of intelligence". Whilst the intelligence relevant to environmental analysis and modeling underlying AWhere™-ACT have not been described in detail here [Corbett et al., 2000], the framework presented sets the stage for integration of knowledge and information systems. With the computing power available to researchers and developers, the natural resource management and environmental modeling world sees many diverse and very specific applications. Whilst it is unlikely that a single model or tool will resolve all possible concerns of decision-makers, tools like AWhere™-ACT can provide a common foundation from which to begin more specific investigation.

6. ACKNOWLEDGMENTS

The Almanac Characterization Tool (ACT) was developed at Blackland Research Center, Texas A&M University. AWhere™ is a new product being developed by Mud Springs Geographers Inc. that supports ACT databases thus the interim product AWhere™-ACT. We would like to acknowledge the support of Dr Ray Meyer.

7. REFERENCES

Corbett, J.D., J.W. White, and S.N. Collis, Spatial Decision Support Systems and Environmental Modeling: An Application Approach. Chapter 3, *Geographic Information Systems and Environmental Modeling*, K.C. Clarke, B.E. Parks and M.P. Crane (Eds), 2000.

Corbett, J.D., S.N. Collis, B.R. Bush, R.Q. Jeske, R.E. Martinez, M.F. Zermoglio, Q. Lu, R. Burton, E.I. Muchugu, J.W. White, and D.P. Hodson, Almanac characterization tool. A resource base for characterizing the agricultural, natural, and human environments for select African countries. Texas Agricultural Experiment Station, Texas A&M University System, Blackland Research and Extension Center, Report No. 01-08, documentation and CD-ROM, March 2001.

Kurata, D., *Doing Objects in Visual Basic*, Sams Publishing, Indianapolis, IN, USA, 1999.

Leung, Y., *Intelligent Spatial Decision Support Systems*, Springer-Valeg, Berlin, Germany, 1997.

Microsoft MSDN Online. http://msdn.microsoft.com/workshop/essentials/glossary. Accessed July 20 2000.

Sarang, P.G.. Develop Three-Tier Applications with VB6 and MTS. ZD Inc. http://msdn.microsoft.com/library/periodic/period99/vb6mts.htm. Accessed July 15 2000.

Vogel, P., *Visual Basic Object and Component Handbook, Prentice* Hall, Upper Saddle River NJ USA, 2000.

Zeiler, M., Modeling our World: the ESRI Guide to Geodatabase Design. ESRI Press, Redlands, CA, USA, 1999.