David W. Gallaher, Director
IT-Development Department
Jefferson County
100 Jefferson County Parkway
Golden, CO 80419-3530
dgallahe@jeffco.us
303-271-8781

BUILDING A PRODUCTION PARCEL OBJECT SYSTEM: MOVING GIS TO THE NEXT LEVEL

Abstract: One of most difficult problems facing the GIS community today is the storage, manipulation and updating of GIS data. We find ourselves using awkward constructs like smart blobs to store vertices in a relational database. The problem is that features in a GIS database are inherently objects (like parcels, roads, addresses, etc). They do not often fit well in a table as like features (parcels) may have many items (vertices, owners, improvements) while others may have very few. Relational databases were originally designed for accounting and insurance functions. We have always had the problem of trying to shoehorn GIS data into these relational constructs. New object oriented technologies like J2EE and .Net allow us to treat GIS features like the discreet objects they are but with a true object database we can finally deal with these features in a clean and efficient manner.

We attempted to build a Parcel Object for some years but it was not until new theory was created, based in part on the work of Ale Raza, (Object Oriented Temporal GIS for Urban Applications, 2001) that we achieved success. For the purposes of a county we determined that the Parcel represents the basic atomicity of a county. It is the discreet unit that all other features can be created (such as zoning, addresses, city boundaries). These larger features are then aggregations of one or more parcels. These features can then be easily represented by items in the database as opposed to creating whole new layers. The result is very efficient storage and representation (no overlaps), which minimizes maintenance. The problem is parcels themselves can change. They can change in size, orientation, subdivision, aggregation and elimination. Object Oriented technology allows us to use parent child relationships to manage what happens to a parcel, zoning or any boundary. With this technology also comes the possibility of introducing time into GIS, not by a snapshot of a layer, but by a discreet feature. We can now store, in a single database, all the features and the ability to create larger features and all their feature histories at the same time in the same place. This new version of GIS could be called T/SIS, for Temporal/Spatial Information Systems. It is related to conventional GIS in the same manner that GIS is related to CAD. GIS added topology to CAD making spatial analysis possible. Adding object based time functionality to GIS makes efficient spatial temporal (historical features) query possible.

THE PROBLEM

The existing GIS data models have changed little over the last twenty years. The major leap forward from CAD systems to GIS was the adding of intelligence (attribute data and topology) to the existing digital line work. Fantastic progress technology has made in the manipulation, editing, query and display of spatial data but the fundamental data model has stayed basically the same. We find ourselves using awkward constructs like smart blobs to store vertices in a relational database. The problem is that features in a GIS database are inherently objects (like parcels, roads, addresses, etc). They do not often fit well in a table as like features (parcels) may have many items (vertices, owners, improvements) while others may have very few. Relational databases were originally designed for accounting and insurance functions. We have always had the problem of trying to shoehorn GIS data into these relational constructs.

The basic unit or atomicity of a County is the parcel. From the parcel level, virtually every other political land feature can be created from an assemblage of parcels (such as zoning, addresses, city boundaries). These larger features are then aggregations of one or more parcels. These features can then be easily represented by items in the database as opposed to creating whole new layers. The result (the Parcel Object) is very efficient storage and representation (no overlaps), which minimizes maintenance. The true object databases have another useful feature in that they are extremely fast. The problem is parcels themselves can change. They can change in size, orientation, subdivision, aggregation and elimination. Object Oriented technology allows us to use parent child relationships to manage what happens to a parcel, zoning or any boundary. With this technology also comes the possibility of introducing time into GIS, not by a snapshot of a layer, but by a discreet feature. We can now store, in a single database, all the features and the ability to create larger features and all their feature histories at the same time in the same place. This new version of GIS could be called T/SIS, for Temporal/Spatial Information Systems.

THEORY

The initial work and design of the parcel object is based on the work of Ale Raza. His Ph.D. dissertation; *Object-Oriented Temporal GIS for Urban Applications* became a template for our development. His work laid the foundation to the Parcel Object problem. No attempt will be made to reproduce the theory in this paper, however several significant changes, which were required, will be covered. Ale Raza's C++ (the Raza Model) demonstration code provided initial guidance in our software design. Most changes were a result moving a technology from an academic research effort into a production product. The largest change was from a change of focus from a zoning focus too an address focus. The County needed to centrally manage all addresses as an attribute of a parcel. Our application had to provide a single address update capability to all County departments.

The Raza Model made a full reuse of arcs, points and provided for topology explicitly stored in an Object Relational database. However, this resulted in a very complex model (with many objects) in theory with little functional implementation in the code. Complexity derives from functionality that the County may never utilize (e.g. Explicitly stored topology) It was made even more complex by the use of Arc/Info to capture topology. We were concerned that debugging would be very difficult which would lead to longer development time to complete the project.

The model used by the developers (the Jefferson County Model) has many key differences with the Raza Model. The application is written in Java utilizing J2EE design strategies All data is persisted in a true Object Oriented database. This means there is no need for an Enterprise Java Bean construct as the objects can be accessed directly. The database initially stores spatial parcel data and its attributes (including addresses). The model communicates via XML (as to allow maximum numbers of existing systems to communicate with it). It is less complicated than Raza model and therefore easier to maintain and debug. There are less overall objects in the database and developers more easily follow the theory and structure. Like the Raza Model, the Jefferson County Model is able to query historical neighbors of any parcel. It can query how a parcel has evolved over time. Additionally, it is able to produce generate files for use by Arc/Info from any query. The simplicity does come at a price, in that individual arcs, points are not reused and topology not explicitly stored in the database. Both models use external gis process to compute overlays.

BACKGROUND

The Parcel Object Project was tasked to build, manage and display parcel and address information. This project became known as ENDOR (Encoded Neighborhood Data Object Repository). The County Address Management System (CAMS) is the system that was replaced by ENDOR for maintaining and managing addresses within Jefferson County. CAMS digitally managed both graphic (spatial) and tabular components. A bit of background on CAMS is useful here in that it explains the functional requirements of ENDOR.

The tabular portion of CAMS lived in a DB2 database on an IBM AS/400. This legacy system was maintained by IT Development (ITD) and was used by nearly all other County departments. County business rules and policy dictate the use of addresses by a number of county agencies, including the Building Department, the Clerk & Recorder's Office, Elections Department, Planning & Zoning, the Assessor's Office, and the Sheriff's Office.

Through the years, CAMS had developed symptoms of corruption in that some functions improperly handle data, such as through tagging items so that users can 'attach' to improper addresses, leading to the association of parcels, and other county records, to an incorrect or invalid object. CAMS was also inefficient. The problem just mentioned often results in the unfortunate expenditure of additional staff resources and time in researching and fixing data problems. This cost was borne by both CAMS user departments and by ITD data and programming staff.

Unfortunately, CAMS was initially designed to accommodate existing user work processes versus modeling and re-engineering intra/inter-departmental processes in an attempt to achieve greatest efficiencies. For example, the existing CAMS requires extra manual data entry in ‘sister’ environments such as the Assessor/Treasurer System (ATS). In the ATS, addresses are hand-entered rather than through the propagation of change through triggers or stored procedures. Moreover, CAMS does not help expedite work procedures pertaining to the integration of addresses and its associated data- whether the Assessor’s Office property data, or Planning & Zoning’s land-use and zoning data.

Synchronization of Arc/Info’s spatial databases and DB2’s tabular data was yet another issue. Updates were done in a once-weekly, en-mass, ‘snapshot’ manner versus real-time on an individual address basis. These weekly updates were susceptible to occasional failure due to the variety and nature of the disparate hardware, networking, operating system (OS), database, and programming components in place, as well as the additional complexities of incorporating geographic information systems (GIS) functionality into legacy systems containing traditionally purely-tabular data. Overcoming these problems was a major ENDOR objective.

ENDOR ARCHITECTURE

ENDOR is the result of the CAMS replacement. Although some of the old CAMS process flow was utilized, ENDOR uses component-based architectures running in a Unix/Linux environment. Address-based needs were met while improving application performance, ensuring data integrity, and processing address changes in near real-time. All current applications were retrofitted, as necessary, to use ENDOR. Regardless of platform, ENDOR provides better access to, or incorporation of, property and other ancillary data. It provides complete parcel/address history. A visualization of the system can be seen in figure 1.

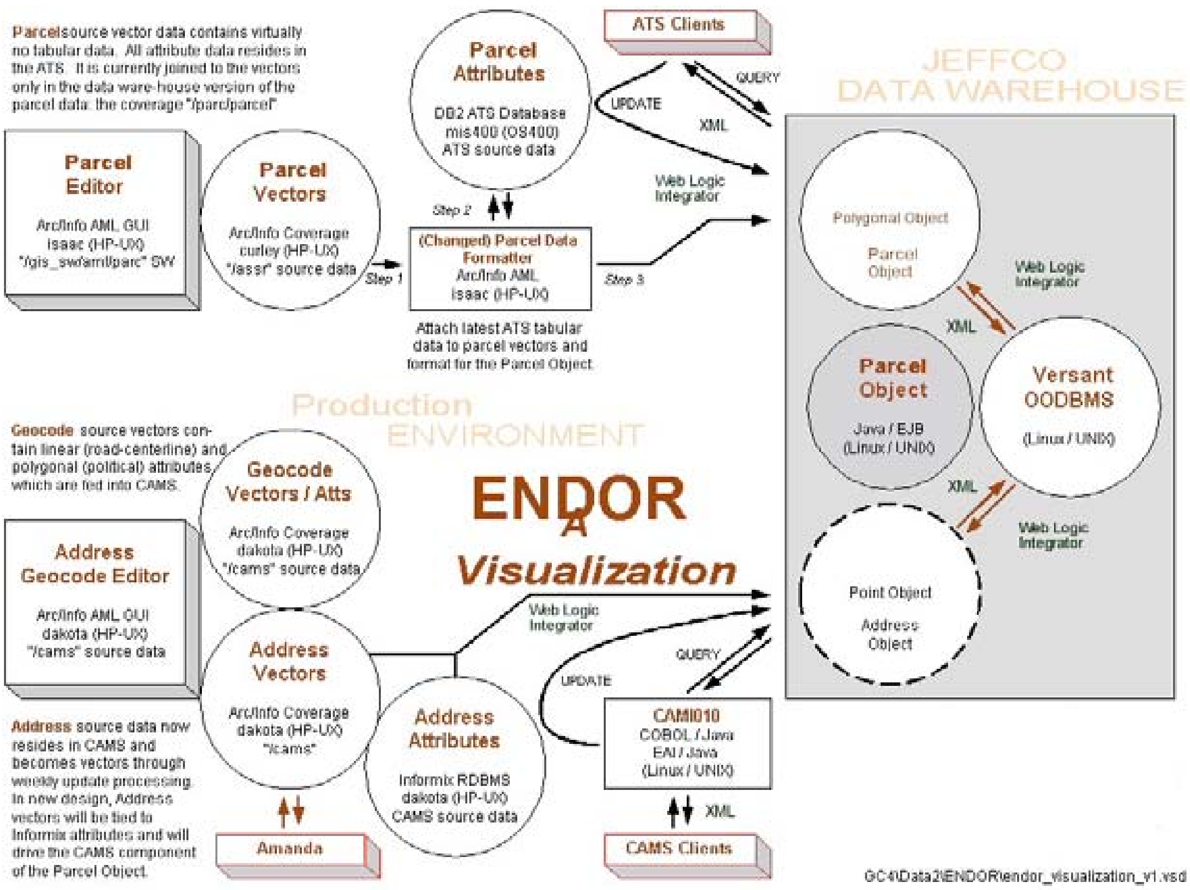

Figure 1. ENDOR visualization

ENDOR is made up of five components:

1. AGE (Address-Geocode)
2. Parcel Editor
3. Historical Attributes
4. CAMI010
5. Parcel Object

BEA Weblogic Integration is utilized to integrate these components.

The **AGE** component represents the re-engineering of the Geocode portion of CAMS and will also include, in a GUI environment, the business rules and functionality. AGE maintains the County's "Road Centerline Network", which includes political and municipal boundaries and addresses. This merger removes the need for multiple platforms and systems to perform addressing functions. The AMANDA system, which is the source of all county addresses, is integrated with the AGE component. All address data is then passed to the Parcel Object.

The **Parcel Editor** (GUI), which is under the domain of the County Assessor, is an AML-Arc/Edit application, used to modify/add spatial data of a parcel(s). This component provides the Parcel Object with spatial data updates.

The **Historical Attributes** component consists of the accumulation of Assessor tabular parcel attribute changes made on a daily basis. These attributes along with spatial modifications made within the Parcel Editor make up the bulk of historical tracking within the Parcel Object.

The **CAMI010** component provides platform independent address editing/retrieval needs of AS/400 legacy applications, COTS replacements and new development. Address data, retained within the Parcel Object, are available via this component.

The **Parcel Object** component, considered ENDOR's "engine", maintains all parcel data, both spatial and tabular and will be responsible for all historical versioning. The Parcel Object utilizes an object-oriented database (Versant). All ENDOR database processing occurs within the Parcel Object.

These components are diagrammed in figure 2.

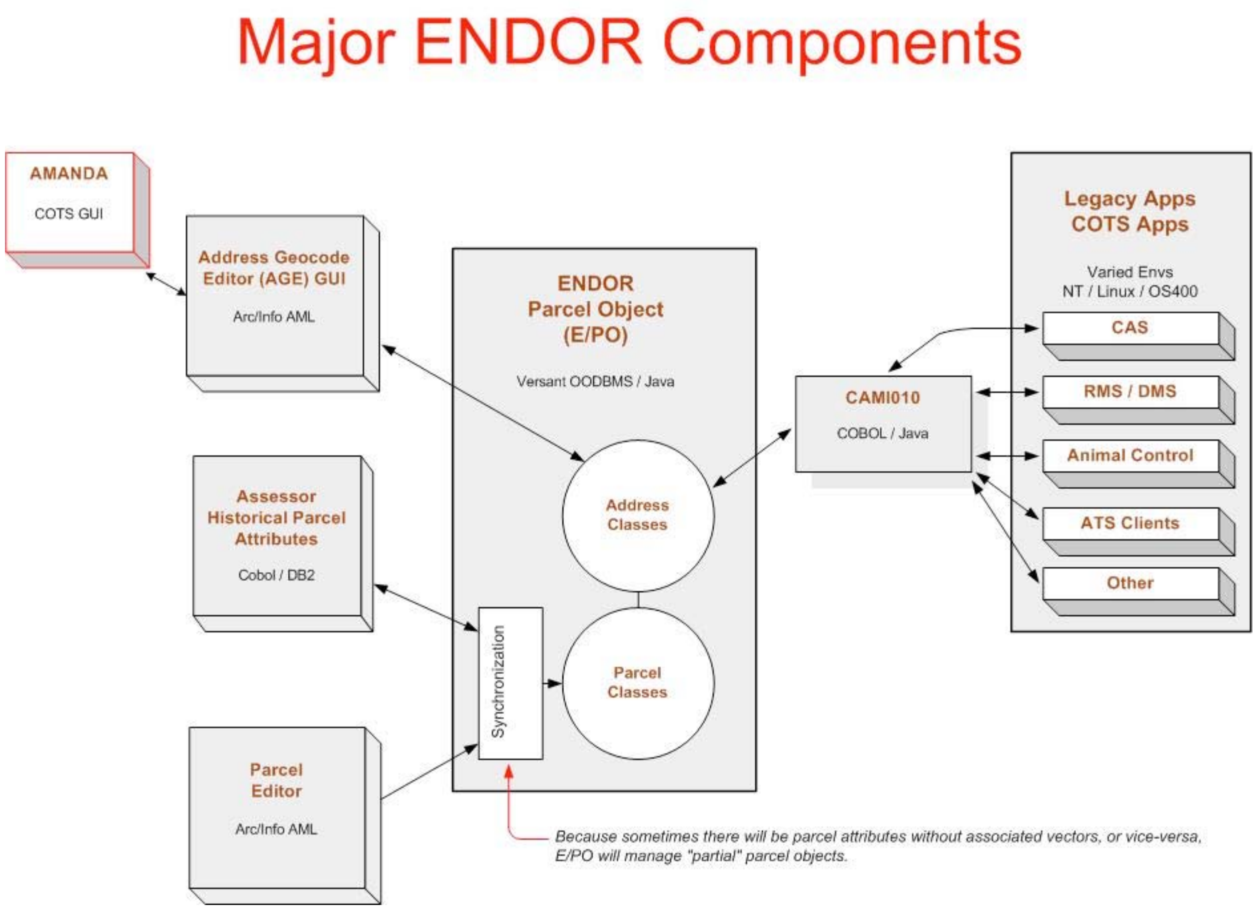

Figure 2.

The Object database (Versant) has no tables, just records. This takes some adjustment as RDBMS has been with us for so long. It some regards object databases are similar to hierarchical databases in their organization. It is completely possible to store all the information about a parcel, it's schedule numbers, ownership, zoning address and it's spatial vertices on a single record. While we examined that possibility it was rejected due to time latency of the source data.
The problem arises from source data creation occurring at different rates and at different times. The Assessor office assembles tabular parcel data on a daily basis, however the spatial data may be as much as one month behind. The address data is generated by Planning and Zoning also on a daily basis but not in sync with the Assessor's Office. This leads to the creation of a partial parcel object. All the parts are not complete but any existing part should be available for query. This lead to three discrete object databases, parcel tabular, parcel spatial and address tabular and point spatial. These three databases form the Parcel Object.

HARDWARE AND SOFTWARE

The hardware used for the application is extensive due to the n-tier nature of the application. The entire application could reside on a single server but this would have resulted in degraded performance and lower scalability. The hardware used for the production version of ENDOR is shown in table 1.

TABLE 1
ENDOR HARDWARE

Purpose	Make / OS	Processor, RAM/Disk
AGE / Parcel Editor End User Workstation	HP Envisex Pseries Model 63253A	N/A
Other End User Workstation	Intel – Windows NT 4.0	Intel Pentium
AGE Application Server	HP 9000/879 HP-UX B.11.00	PA8000 RISC 180Mhz Dual 2GB, 93 GB
Parcel Editor Application Server	HP rp7400 HP-UX 11i	PA8700 RISC 750Mhz Dual
Legacy App and Historical Attribute Application Server SHR400	AS/400 820 OS/400 V5R1	RISC 2395 1.5GB, 260 GB
ENDOR Application / Integration Server	HP / HP-UX 11i	PA8700 RISC 750Mhz Dual
ENDOR Database Server	RedHat – Linux 7.2	2GB Pentium 4 2GB, 72 GB

The databases used in ENDOR are shown in Table 2. This appears to be a long list but some of the databases reflect data being imported from legacy systems, configuration databases such as Oracle and databases associated with Arc/Info.

TABLE 2
ENDOR DATABASES

Vendor	Product	Version
IBM	Informix	9.2
Oracle	BEA WLI	8.1.7
IBM	DB2	N/A
Versant	VDS	6.0.0
ESRI	Arc/Info	8.1

The software is primarily built with Java code using BEA's Weblogic Server and Integrator components. The parcel editor and Age components were built using ESRI AML code. The hardware and software come together in figure 3, which lays out the physical model of the application.

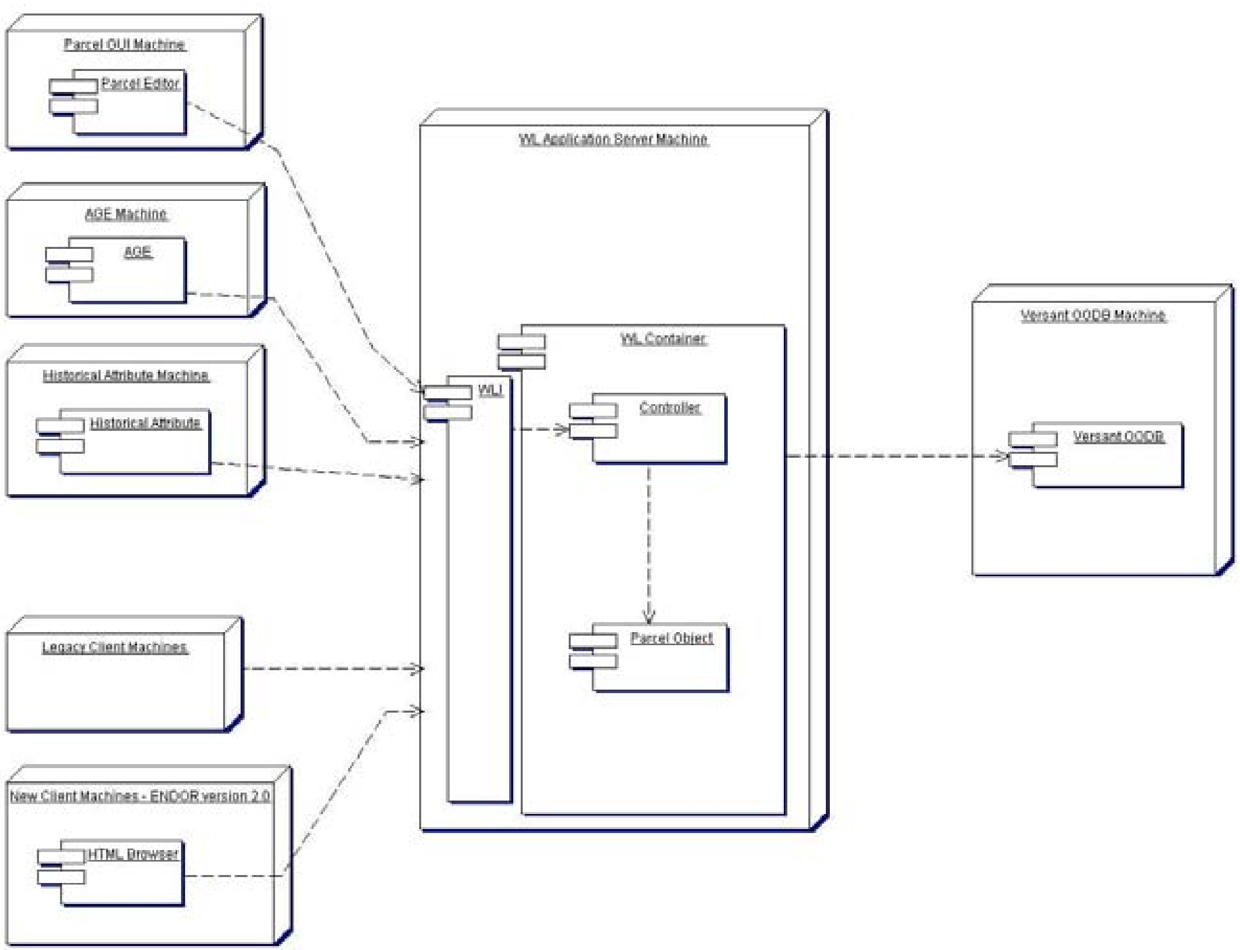

Figure 3.

Results

This project started out as a pure R&D effort to determine if it was possible to build a Parcel Object. We considered this effort as something like a search for the Holy Grail of GIS. The transformation from R&D effort to production product is a process seldom seen in local government. It has taken several years to reach this objective but we are already seeing stunning results. Whole townships of data can be rendered from the database in seconds. Any of 220,000 parcels can be queried for its current spatial and tabular data as well as its historic and spatial data. These queries and renderings are extremely fast, much quicker then the three second specification required. Endor is now a key piece of the County infrastructure. Any system can request parcel and address data from a single source and request thousands of records for any time period. We are now planning simple user interfaces and new applications using ENDOR as the engine, such as a new Assessor Treasure system. We will soon add roads and zoning. Commercial possibilities are being considered for this technology.

This project was built with the latest technology and architectures often on the bleeding edge. With the Parcel Object we have created something new. It is related to conventional GIS in the same manner that GIS is related to CAD. GIS added topology to CAD making spatial analysis possible. Adding object based time functionality to GIS makes efficient spatial temporal (historical features) query possible. We now have a means to track every change in the County, in near real time, such that no change will ever be lost and the County status at any time can be recreated.