CITY OF DENVER ADDRESSING: MODELING, MAINTENANCE, AND SUPPORT

Daniel Hauser, GIS Developer, City and County of Denver, Colorado USA

Abstract

The City and County of Denver relies on accurate addressing in order to provide location-based services to its residents, businesses and visitors. A centralized addressing system has been developed to support increasing demands for address-related services. Denver has developed a comprehensive addressing geodatabase model using Visio UML. Denver's address maintenance staff and DenverGIS have performed extensive data quality control and clean-up tasks to migrate data from legacy sources into the new Denver Address Database (DAD). A custom ArcMap application has been developed using C# and Visual Studio .NET 2005 to support address creation and maintenance in a versioned geodatabase environment. The geodatabase model, the development of the Address Maintenance Application and the lessons learned in creating the Denver Address Database will be discussed.

1 Introduction

DenverGIS is a centralized GIS department supporting over 30 departments and managing 800+ data layers and associated information resources within the City and County of Denver. As a municipality, the City and County of Denver is responsible for providing many geography-related services to its residents, businesses and visitors. In 2005, DenverGIS set out to create a centralized addressing database to provide consistent addressing services for City and County agencies. DenverGIS focused its efforts on researching what agencies currently used addresses and where those addresses are stored. Over 300 address databases were found across the City and County's network. DenverGIS performed extensive research on known best practices and looked at various resources available to learn about addressing. Included in this research was the ESRI Calgary address geodatabase model [1], the new (and evolving) Federal Geographic Data Committee (FGDC) addressing standards [2], and the United States Postal Service (USPS) addressing standards (Publication 28) [3]. DenverGIS worked extensively with Denver's Public Works department to identify existing business processes related to addressing. The City Engineer (Public Works) by ordinance is responsible for addressing in the City and County of Denver.

2 History

Within Denver, many agencies require addressing and use addressing as an identifier for much of their assigned work such as permitting, inspections, licensing, appraisals, etc. By ordinance, the City Engineer (Public Works) is deemed responsible for addressing all the structures found within Denver. Prior to 2000, addresses were issued on paper address cards and recorded on Mylar plat maps. In 2000, a simple relational database called the Corporate Address Database (CAD) was created. The CAD was created to serve the needs of multiple agencies, but specifically to support permitting. Rather than manually enter addresses from the historical address cards, the City and County of Denver bulk loaded addresses from the Assessor's mainframe system. The CAD did not meet the needs of all the City and County agencies, as it focused on structures and specifically exterior facing addresses. The Assessor's addressing contained miscellaneous, approximate and range addresses. This meant that vacant land and interior (subaddresses) were largely ignored and many addresses were incomplete. Also the CAD did not provide any geographic coordinate information. Since the CAD was not complete, many agencies continued to create and support their own data sources (usually Microsoft Access or Excel) to meet their addressing needs.

In 2002, DenverGIS created an ArcGIS based solution that would allow Public Works to add coordinate data to the addresses. Through the use of event layers, as addresses were maintained, they populated an X coordinate and Y coordinate field in the database. After 3 years, approximately 3000 addresses (out of +200,000) had been placed, and little additional address cleaning had occurred.

The major goals of the current addressing project include creating a single repository for addressing that would meet the needs of all of our constituents. The system should be GIS based, will provide tools to help streamline address production, maintenance and clean-up, and shall replace all the home-grown addressing solutions.

3 Geodatabase

The Denver Address Database (DAD) was designed using Visio Professional 2003 to create a geodatabase model that could be applied to an Oracle database. The model is based upon the ESRI Calgary Data Model. Modifications were made to the model to support the City and County of Denver's addressing needs and to meet the FGDC and USPS addressing standards. Also included in the model are various relationships and tables designed to support the future needs of various data layers such as Parcels, Street Centerline, Building Outlines, etc.

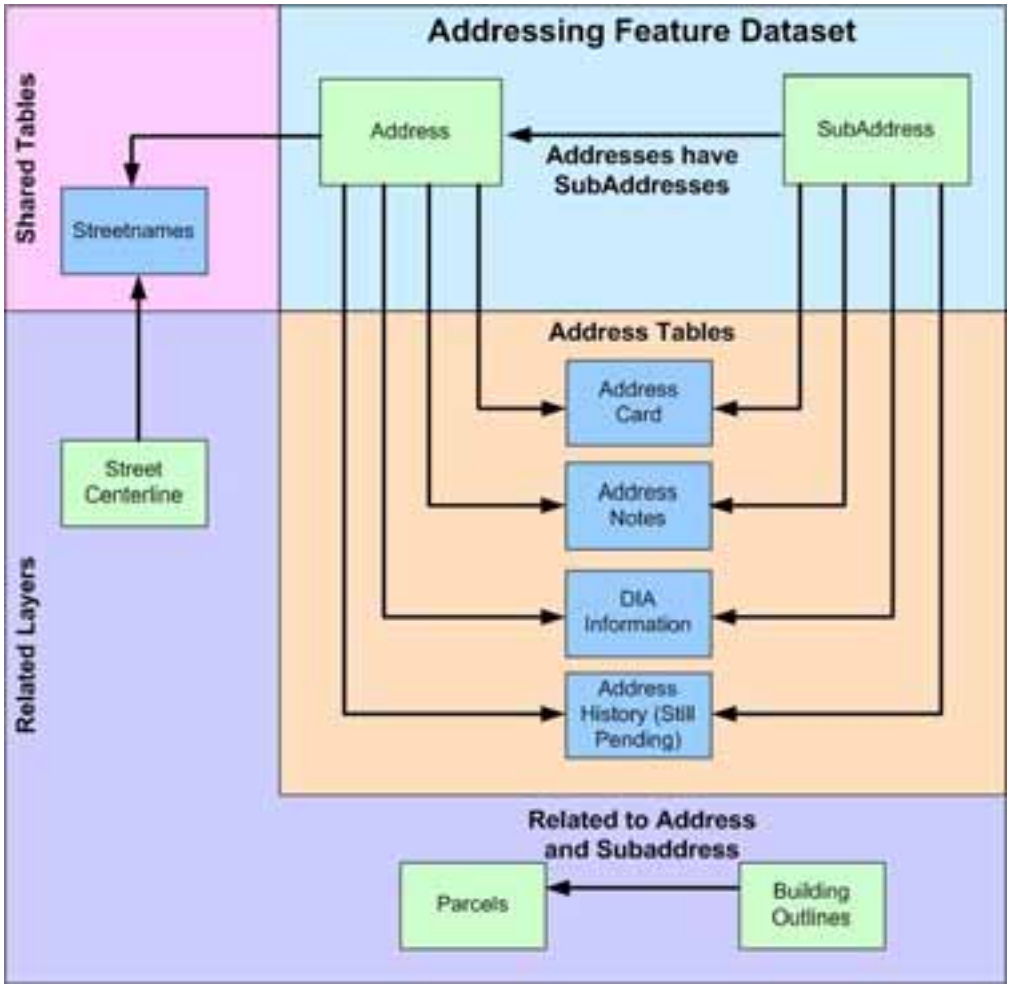

Figure 1: Address Geodatabase Model

The DAD data model was designed to support the existing addressing data capture requirements. The address and subaddress feature classes share multiple attributes. To prevent duplication, abstract classes were created to hold shared attributes. Whenever possible, domains were created to enforce data integrity in the attribute fields. Various feature class subtypes were created and attributed allowing us to assign specific domains based upon the selected subtype. Addresses/Subaddresses also maintain descriptive/informational attributes. To capture these attributes, Address Card and Address Note tables were created. To maintain historical changes to the addresses/subaddresses, an Address History table has been added to the model. There are also addressing attributes specific to Denver International Airport that are required by the FAA (such as crash zone). These attributes are stored in their own tables and related to individual address/subaddress. Addresses share street name information with Denver's street centerline data layer. A street name table was created allowing support for one maintained source of street names. This also allowed addresses to support multiple names. This is important for legacy name changes and temporary/vanity names. Finally, attributed relationships were created between Address/Subaddress and Parcels and Buildings. Parcels and Buildings were identified as layers requiring addresses. There will be additional relationships built in the future as the need arises. One of the key benefits to using Visio to model the geodatabase is the simplicity of modifications and re-application of the data model. Figure 1 is a high level representation of the Denver Address Database model.

4 Address Cleanup and Migration

As mentioned previously, there were known problems with the existing addressing data. Prior to the launch of the Denver Address Database, DenverGIS created a systematic work plan to identify required address clean-up tasks. DenverGIS and Public Works staff spent hundreds of hours performing address clean-up work. This work was designed to place the addresses geographically, create consistency and integrity across the database and identify and clean-up duplicate or missing addresses.

Since the data was coming from a relational database, lacking geographic information, the first task was to place the addresses geographically. Matching parcel situs addresses and legal descriptions (to identify parcels) to the addresses in the CAD, DenverGIS was able to automate the placement of approximately 96% of the addresses. An additional 2% were placed using custom placement tools provided to the addressing staff. Once geographically located, a manual process was used to review address placement to recognize potentially misplaced addresses. These addresses were reviewed and corrected if necessary.

The previous addressing application provided text boxes to enter in attributes for addresses. This created a poorly controlled environment where users were not required to enter standardized information. The next step in address clean-up was to create standardized lists of acceptable input for as many text fields as possible. These lists were then used to standardize the existing attributes. The lists were also implemented as domain values in the new data model.

Over 700 paper maps were created to identify parcels that did not have a corresponding address in the CAD and identify addresses that were "known" bad addresses". In many cases these were vacant parcels or parcels with multiple buildings that contained range addresses. (See Figure 2) When ever possible these addresses were added or cleaned-up in the CAD.

There were many other clean-up tasks performed (standardizing street names, predirectional confirmation, etc), and there are on-going clean-up tasks continuing today even following our move into the new maintenance environment.

In order to transfer the data from the existing relational database to the Denver Address Database, SQL migration scripts were written. These scripts (~3000 lines of SQL) would take the existing data, standardize the attributes and fields and move it into temporary tables that could then be loaded seamlessly into the new geodatabase. Model builder was used after the SQL scripts were run to load the data into tables, populate relationships, calculate latitude/longitude and X, Y coordinate fields, and register the feature classes and tables as versioned.

In these scripts, code was written to identify problem data that could not be transferred. This allowed DenverGIS to work on the migration scripts while Public Works continued to maintain and clean-up addresses in the existing system. Addresses identified as "Bad", were cleaned up prior to the final migration.

The migration scripts also performed many of the data standardization tasks such as transforming Single Residence to Single Family Residence or the name Monaco St Pky to N Monaco Street Pkwy. The SQL scripts were also responsible for determining the type of address to be added to the new geodatabase. In the CAD, Public Works only captured Structure Addresses. To support other agency needs, the new DAD contains Addresses with the subtypes: Structure, Land, Associated and Utility. The DAD also supports Subaddresses. The SQL scripts could identify subaddresses and utility addresses and place them in the proper subtype or feature class. The scripts would even create associated addresses for subaddress that did not have an existing parent address in the CAD.

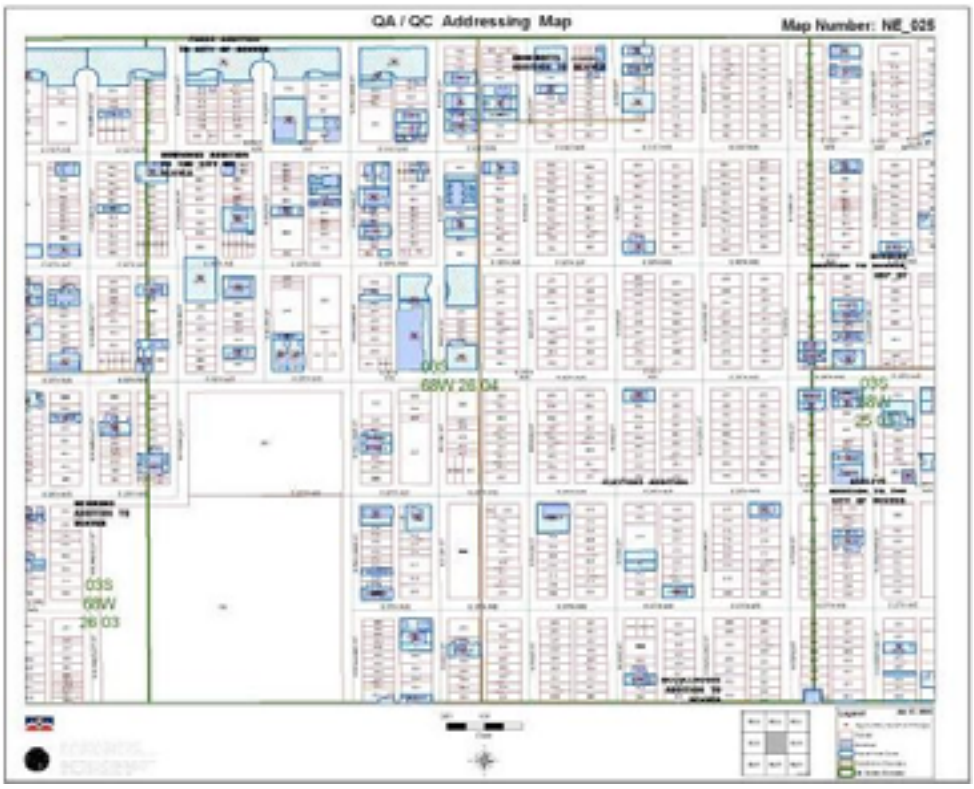

Figure 2: Address Clean-up Maps - Parcels missing addresses or with addresses identified as 'Bad' addresses.

5 Address Application

Creation of a custom maintenance application for addressing was an important aspect of this project. None of the address technicians in Public Works are trained in GIS and have only had

limited exposure to GIS. The development of the Denver Address Maintenance Application (DAMA) followed a standard software development lifecycle, including: Planning and Analysis, Design, Development, Testing and Training, and Deployment followed by a warranty/maintenance period. DenverGIS is providing on-going support for bug, issues and enhancements following the warranty support period.

DenverGIS spent several months gathering requirements, studying current applications and identifying current workflow processes. Workshops were conducted to discuss existing processes, technical bottlenecks and questionable practices. Public Works was asked about what could make their jobs easier and more efficient. The address staff was asked what features they would like to have added to the existing application and discussions were held to identify changes that would be necessary in the new system to allow the DAD to support addressing needs outside of Public Works.

Following the requirements and analysis period, DenverGIS focused on how to design an application that would support the user's current workflow, support the maintenance in a versioned geodatabase, provide the ability to capture both addresses and subaddresses and manage relationships between the various tables and data layers. It was determined that the application would be an ArcGIS Extension with a custom toolbar and custom commands. Wherever possible, existing ArcGIS tools would be leveraged and no existing GIS functionality would be taken away from the user. The graphical user interface (GUI) was designed to look

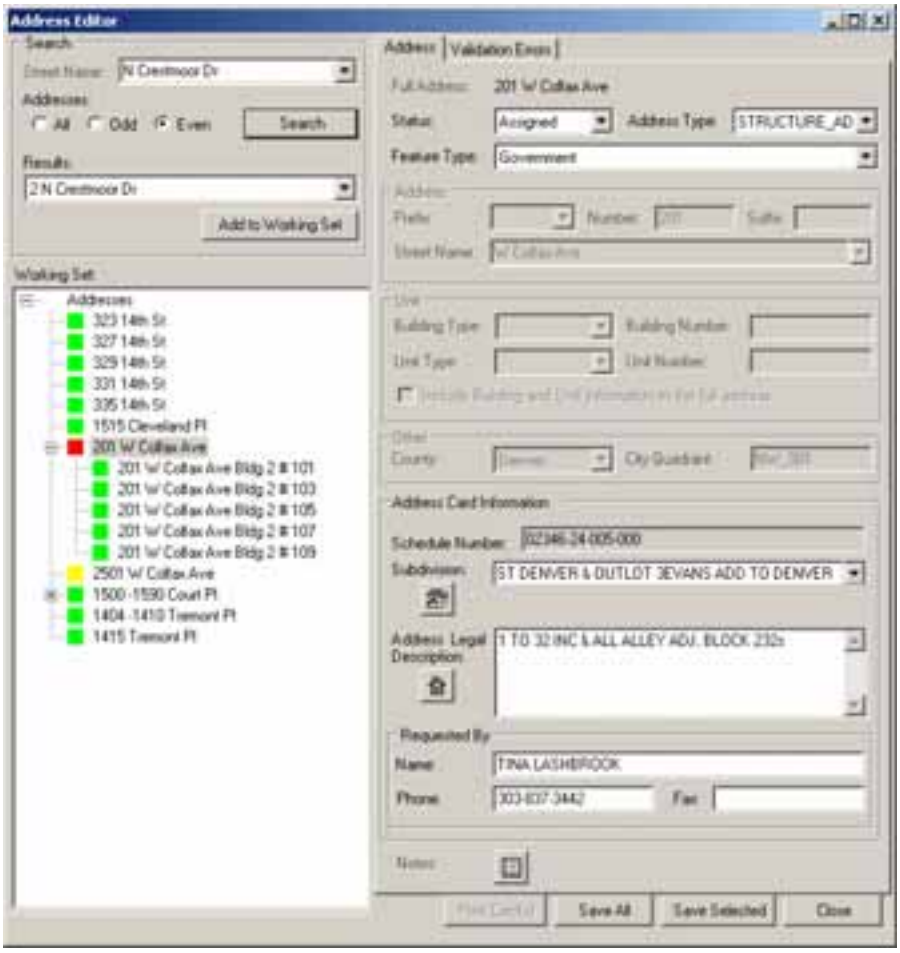

Figure 3: Address attribute editor showing both committed, modified and save error addresses.

similar to the existing address application with efficiencies added to simplify the form and allow for multiple addresses to be maintained concurrently. Similar to the ESRI ArcGIS attribute editor, the address attribute form contains a treeview control on the left and the attribute controls on the right (See Figure 3). Other tools (See Figure 4), such as the ArcGIS sketch tool, an address placement tool, a modified selection tool, a simplified target layer control, and a custom address attribute editor would also be added to assist the addressing staff.

Figure 4: Custom address toolbar

Development of the DAMA was initially started using ESRI 9.1 and Microsoft Visual Basic 6.0. Soon it was determined that the application should instead leverage the power of Microsoft's .NET environment and the application was re-written in Microsoft Visual Studio 2005 using C#. Other applications integrated into the application include Crystal Reports for report creation, Adobe Acrobat for printing and emailing of reports, and Microsoft Word's spell checking capabilities to support data integrity in the free form notes field. The application leverages C# and ArcObjects event capabilities to react to map driven events. Most of the fields in the GUI are combo boxes that can be populated from domains when the treeview select event is fired after selecting an address. Several user controls were created to support existing address searches and custom forms manage functionality such as association of legal descriptions, addition of notes, creation of new subaddresses, and break-out of range addresses. Context menus were added to the treeview control to allow the user to flash features on screen, clone addresses, delete designed addresses, select an address on the map, select the subaddresses related to an address, and several other functions.

The extension is designed to load the appropriate data layers at start-up. The layers are loaded through layer files that contain custom symbology for all the applicable data layers. The application contains 27 layers (including Address and Subaddress) and 8 tables. When a user starts the application an edit session is started. If there is not a version associated with the user's username, one is automatically created and set as the current version. The version is reconciled with SDE.Default at start-up to make sure any changes to SDE.Default are seen by the user. User's may add or select an address or addresses. Users are only allowed to modify specific attributes based upon the status of an address. Once an addresses status is set to assigned, only non-address specific attributes may be modified (owner, subdivision, legal description). When an address is saved, it is first validated. Validation errors will stop the save and provide feedback on what validation error occurred (See Figure 5). Validation errors include: duplicate addresses, address part validation, street segment validation, side of the street validation, Denver County boundary validating, etc. Validation correction requires either changes to the offending address or in some cases allow overriding by a manager. When an address is saved successfully, it is automatically reconciled and posted to SDE.Default. A multi-version view is used by a extraction, transformation and loading (ETL) script, to make new or modified addresses available to the permitting system within 1 minute of processing. By using a multi-versioned view, existing applications that used addresses can continue to operate without major enhancements to connect to the DAD. A placement tool is also provided to assist in the placement of new addresses, cloned addresses and addresses that were not able to be geographically placed prior to migration into the DAD.

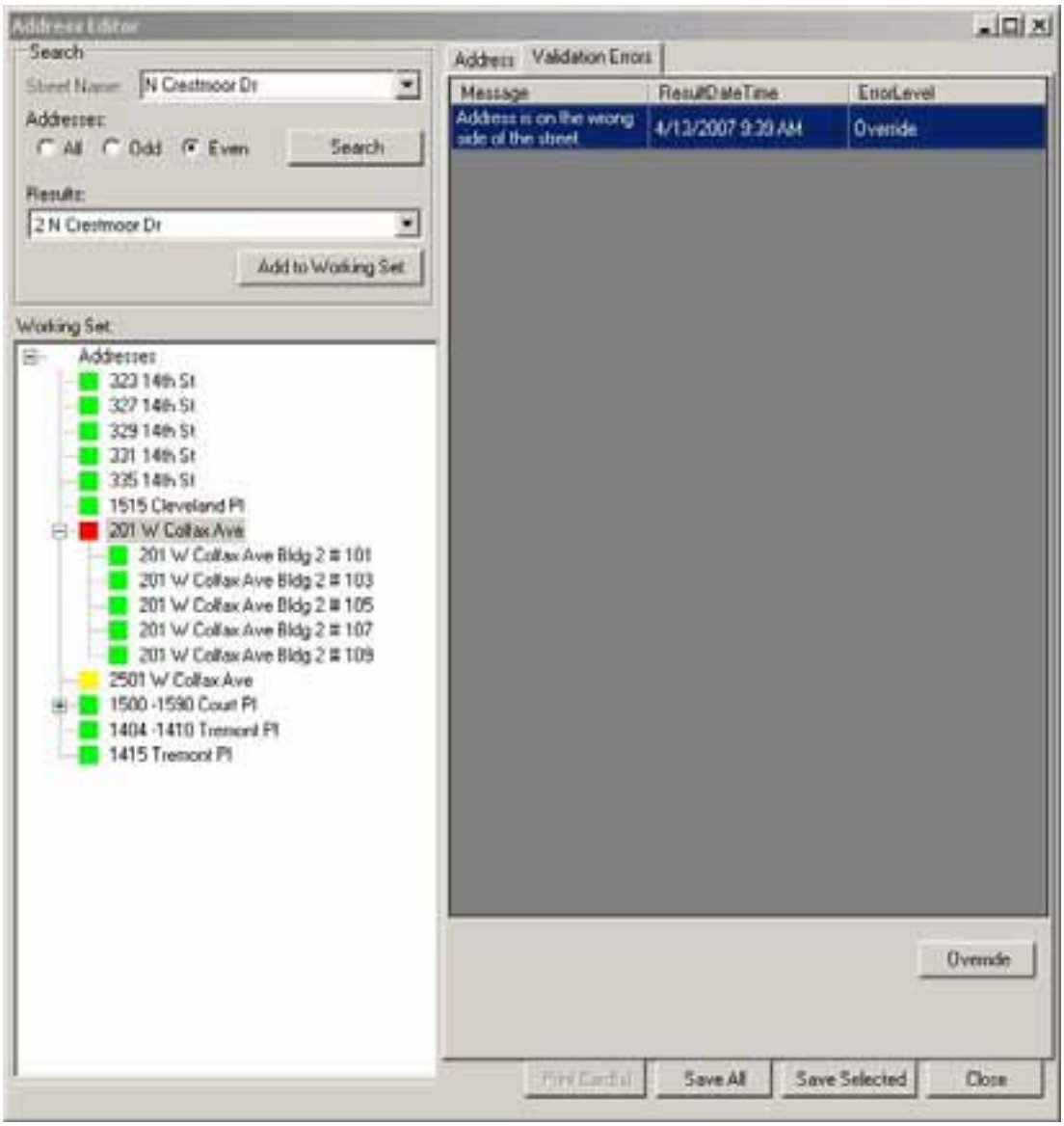

Figure 5: Side of street validation error

Users were involved throughout the development, consulting on functionality. Users spent more than 2 months conducting user testing on the application. During user testing several lead users were trained on the application and asked to perform standard work tasks in the new environment, testing the application against current processes. The lead user also used this time to develop a user manual for the addressing staff. The lead user then performed user training for the remainder of the address staff.

Migration to the new environment was done over a weekend. The migration scripts had been thoroughly tested throughout development with test data and full data migration performed more than 10 times. The application installation procedure was tested and refined until all known issues were resolved. The application set-up included loading of a Crystal Report .msi file, running a set-up file to load and register the .net application assemblies, add configuration files, register the logging assembly, and load the layer files. Next, the extension and toolbar were assigned to the proper ArcGIS categories in Categories.exe and the application was tested on each machine.

Following deployment there was an informal warranty period where DenverGIS provided weekly maintenance releases fixing identified issues or adding new enhancements. DenverGIS the moved into a support role, providing bug fixes and requested enhancements on a monthly release schedule. DenverGIS continues to support the address application, geodatabase and addressing staff on an on-going basis. A web service was created to provide address validation services to city agencies throughout the City and County of Denver. The web service validates addresses both through the DAD and through a Postal Data validation product.

6 Lessons Learned

Many lessons were learned through the Denver Address Maintenance Application project. The creation of a new geodatabase data model, migration to from an existing database, data clean-up and the development of an address maintenance application was a major impact in time, effort and inter-agency cooperation. Each area of focus contained potential problems that could de-rail the effort. Yet with a strong champion, DenverGIS was able to succeed.

When building an address geodatabase, it is important to leverage known commodities, such as using ESRI Calgary Address Data Model as a basis for the data model. Do not make up fields or leave out fields in the address and street tables. Using the USPS and FGDC standards identify the fields that should be included in an address geodatabase. Use Visio to develop a geodatabase model. Documenting your model is critical, and by using a UML model, your documentation will always be up to date. Also using the schema checker tools, you are able to check your model prior to deployment. Use abstract classes in your model to prevent duplication of fields. This will save time and effort and eliminate the risk of the same field in two classes with different attributes. Building the initial database model took several months of research and exploration. And the model continued to evolve and change late into the project. Do not feel that you can not change your model. If the change makes sense then it is worth the effort to change.

Create a plan to migrate and clean-up your data. Address data is rarely mistake free and not in need of work. Don't let this dissuade you from moving forward toward a geodatabase. Use the migration as an opportunity to clean the data and identify issues within the data. Creating a plan can help identify areas that can be cleaned up easily. Look at the data to identify trends. Often, the same mistakes are made repeatedly. Identification of these trends will allow for the clean-up of large blocks of data. Look for big wins that will motivate the staff to do more. Success breeds success. In 2005, the addressing staff touched approximately 3500 addresses throughout the year. When the clean-up tasks were implemented in 2006 the addressing staff (7 persons) edited (created or edited) over 14,000 addresses or nearly a 400% increase in work. The extra work does require resources or overtime or changes to work prioritization. Make sure everyone knows of the effort and success. People are proud of what they do. Make sure everyone knows their work is appreciated. Migration should be treated in a manner similar to clean-up. Identify the data that is easy to migrate and write those scripts. Then, identify other trends in the data that prevent migration. Write scripts to address each problem. Performing this task will assist in migrating the data and will assist in identifying potential clean-up tasks. Run the migration process multiple times from your development environment. This will assure smooth transition into the production environment.

The following are recommendations for development of an address maintenance application. Don't rush the phases of development. Spend the proper time talking to the client and understanding their needs. The development staff must become subject matter experts at addressing. Developers should be able to understand the work processes and life cycle of an

address. They must understand addressing not only from the perspective of the addressing authority but from the address consumer perspective. Leverage existing ArcGIS functionality. Do not remove existing ArcGIS tools. When necessary, create new buttons that simplify tasks for the users. Involve users in the project throughout the development life cycle. Make sure the users are involved in testing of the application to verify that their needs are met. Practice deployment prior to the launch of the application. This will help identify areas that could cause the deployment to fail.

7 Conclusion

Since the launch of the Denver Address Database in December 2006, the addressing staff have created or edited over 6000 addresses. There are now 5 agencies using the Address Web Service to connect to the DAD. Denver's 311 Customer Service program is very closely integrated with the DAD. This integration allows for real-time reports and maps to be created showing agency response to incoming complaints and issues throughout the city. Also utilizing the data available from the DAD is Denver's Public Safety group, Excise and License, Public Works, et cetera. The development of the DAD has shed light on addressing problems in other agencies and is encouraging the various agencies to clean-up their addresses and tie into the DAD when possible. The DAD is changing the way Denver is addressed.

8 Acknowledgements

For more information about the Denver Address Database please contact:

David Luhan
City and County of Denver
denvergis@ci.denver.co.us

9 References

1 ESRI Address Data Model, www.support.esri.com/index.cfm?fa=downloads.dataModels.filteredGateway&dmid=32

2 URISA/FGDC Street Address Data Standard (Executive Summary) Final Draft, http://www.fgdc.gov/standards/projects/FGDC-standards-projects/street-address/06-09-17.ASWG.ExecSumm.Rev4a.pdf

3 United Status Postal Service Publication 28 – Postal Addressing Standards, http://pe.usps.com/cpim/ftp/pubs/Pub28/pub28.pdf

4 City and County of Denver Geographic Information Systems "City of Denver Addressing: Enterprise Integration Using Web Services" ESRI International User Conference 2007