CITY OF DENVER ADDRESSING: ENTERPRISE INTEGRATION USING WEB SERVICES

Allan Glen, Senior GIS Developer/Applications Development Supervisor
City and County of Denver, Colorado USA
May 2007

Abstract

The City and County of Denver's many agencies and departments require accurate addressing to support daily business operations. A centralized addressing system has been developed to support increasing demands for address-related services. Multiple address databases have been centralized into an enterprise geodatabase. Web services have been developed to provide real-time integration with enterprise applications including customer relationship management (CRM) and permitting. Service-oriented architecture (SOA) enables cross-platform, standards-based integration and supports the consistent address validation, formatting, and identification across enterprise systems. In addition, the address service integrates information from other sources including street intersections and postal addresses from the United States Postal Service (USPS). The benefits of extending GIS services to the enterprise, challenges encountered, and lessons learned will be discussed.

1 Introduction

The City and County of Denver's Technology Services Geographic Information Systems group (DenverGIS) provides corporate GIS services to all city agencies and departments and to the public. Services provided include geographic data management, application development and data distribution. Providing city-wide access to integrated GIS information and applications is critical to support daily decision-making and to promote inter-agency communication and collaboration.

Denver's city and county are combined into a single jurisdiction and is comprised of a diverse group of offices, departments and agencies. For simplicity, these organizational groups will be referred to as agencies throughout this paper.

In 2005, DenverGIS initiated an effort to create a centralized addressing geodatabase to provide consistent addressing services for City and County agencies. The development of the new Denver Address Database (DAD) is described in more detail in an additional paper entitled "City of Denver Address: Modeling, Maintenance and Support" (1).

Over 300 address databases were identified within the City and County and there was clearly a need for a centralized addressing system that was accessible to all enterprise applications. Providing cross-platform access to the address geodatabase was a critical component of the addressing effort and was a key requirement to ensure success.

2 History

A centralized address database was developed in 2000 to support Denver's permitting system. This database was known as the Corporate Address Database (CAD). The permitting system was tightly integrated with the CAD for validation of addresses prior to issuance of permits. This integration was performed through ODBC database links and direct queries of the database tables.

Few additional systems were integrated with the CAD as a result of the effort required to integrate with the database. Integration was performed by using direct database connections and each application contained the separate business logic for querying and interpreting the address database. Stored procedures were developed to aid in centralizing business logic but not all applications were able to use these procedures.

As a result of these limitations, many applications across different agencies developed address databases that were specific to their business function. Many of these databases were developed using Microsoft Access and contained few constraints. Addressing rules across databases were inconsistent and unique identifiers were not available to correlate addresses across systems. The lack of easily accessible enterprise addressing resulted in an inconsistent experience for customers and limited the ability to aggregate and compare information across multiple systems.

3 Enterprise Integration Goals

3.1 Cross-Platform Access

Denver's agencies represent a diverse set of functional areas, most all of which require accurate addressing. The technology platforms that support these agencies are equally diverse with a wide variety of operating systems, databases, and development languages.

Throughout the design process for the new addressing system, it was evident that providing cross-platform, standards-based access to the database was critical to the overall success of the project. A well-designed geodatabase would not reduce the proliferation of address databases unless integration capabilities could be provided that would enable enterprise access to the centralized addressing system.

3.2 Loosely-Coupled Integration

Integration with the previous address database was enabled through direct database connections with direct access to the database tables. Applications were required to understand the database schema and the structure of the database relationships. This tightly-coupled integration limited the potential for database changes and restricted the ability of the database to adapt to changing business requirements.

The ability to provide loosely-coupled integration, where clients know as little as possible about the database, was established as an important objective of the enterprise integration architecture.

3.3 Consistent Business Rules

The ability to centralize the business rules for validating and interpreting addresses was identified as a key goal for the project. With the previous CAD database, each application was responsible for querying and interpreting the address database. This resulted in inconsistent match results across applications even when the applications were utilizing the same data source. A customer address may have validated in one system but not in another due to differences in quality of the address querying capabilities of each system (ex. Differences in matching logic, fuzzy matching capabilities, addressing rules, parsing logic etc.)

3.4 Centralized Monitoring and Quality Assurance

The ability to monitor and report on address validation requests was identified as an important goal for the new addressing system. Validations against the CAD were unmonitored and no metrics were available (ex. number of address validation requests, validation success rates, etc.).

3.5 Centralized Feedback

Many agencies perform field verification of addresses that are associated with their business functions: appraisers verify addresses when visiting properties and inspectors verify addresses when completing inspections. As a result, many addresses are 'visited' but there was centralized mechanism in place that could leverage these existing address verification efforts.

3.6 Multiple Data Sources

Location-based validation is not limited to street addresses and the need to integrate other data sources was identified.

The DAD data model includes relationships to the street centerline database for consistent naming of streets and verification of address numbers against street address ranges. The street centerline information could also be used to validate intersections and this functionality was identified as a requirement for the web service.

Additionally, many City and County of Denver agencies are required to deliver bulk mailings to constituents (ex. Property re-evaluation notices for property tax assessment). Denver's addressing team issue addresses for entities within Denver but do not maintain mailing information for those addresses. In many cases the address does not become a deliverable address until some time after the address has been assigned. The ability to integrate postal address validation into the addressing system would provide accurate addressing for mail pieces and reduce costs associated with bulk mailings.

The Mayor's 3-1-1 initiative was an additional business driver for the integration of mail addresses into the addressing system. The 3-1-1 system provides a single phone number that constituents can call to conduct business with all agencies in the City and County of Denver. The multi-channel system provides access via telephone, email, walk-in and the web. Many customers that contact the call center do not reside in Denver and postal address validation is important to ensure the accuracy and quality of the customer addresses. The ability to identify the location of callers in surrounding jurisdictions ensures that cases are not initiated for service in surrounding jurisdictions.

4 Service-Oriented Architecture (SOA)

Service-orientation describes an architecture that uses loosely coupled services to support the requirements of business process and users (2). A service-oriented architecture (SOA) provides a means of sharing services that can be accessed by multiple platforms and systems.

The City and County of Denver is increasingly leveraging SOA-related technologies to enable application integration and sharing of services between enterprise applications.

A service-oriented approach using web services was followed to provide address validation services to enterprise applications and provide cross-platform access to the address geodatabase

5 Address Web Service

Denver's Address Web Service is based on the Simple Object Access Protocol (SOAP). SOAP is a standard-based protocol for exchanging XML messages, typically using HTTP for transport. The Address Web Service is developed using the Microsoft .NET platform (2.0) and is written in C#.

5.1 Cross-Platform Access

The Simple Object Access Protocol provides the definition of XML-based information that can be used for exchanging structured and typed information between peers in a decentralized, distributed environment (3). SOAP is based on eXtensible Markup Language (XML) and is supported on a wide variety of platforms.

The Address Web Service, at the time of this writing, is being consumed by multiple platforms (Windows, AIX, Linux) using multiple languages (C#, Java, Ruby). Additionally, multiple enterprise software applications consume the service (Oracle PeopleSoft, EMC Documentum).

5.2 Loosely-Coupled Integration

Consumers of the Address Web Service are not required to be aware of the underlying database technology and structure of the database schema. The complexities of the ArcSDE multi-versioned geodatabase are not required to be understood to access the addressing system. The web service interface is fully described using WSDL (Web Services Description Language).

Web services consumers only need to be aware of the web service endpoint (URL) and the WSDL to communicate with the service.

5.3 Consistent Business Rules

Web services provide the ability to implement business rules behind the web service interface. All City and County of Denver addressing rules are implemented in the web service ensuring that all consumers validate and interpret address consistently across applications. All improvements to address validation through the service are immediately made available to all consumers.

5.4 Centralized Monitoring and Quality Assurance

The Address Web Service logs requests from all consumers and provides a centralized mechanism for auditing address validation requests. This information is used to monitor performance, evaluate usage, measure validation quality, estimate completeness of the database, and provide real-time analysis of address validation requests.

5.5 Centralized Feedback Mechanism

An added benefit of centralized monitoring is the ability to identify addresses that are likely to be missing from the address database. Valuable feedback is collected through 3-1-1 by identifying constituent addresses that are not present in the DAD but are identified as USPS mail addresses. Weekly reports of these addresses are prepared for the addressing team to improve the completeness of the address database.

The ability to collect verification feedback from agencies directly through the web service has not been fully implemented at the time of this writing.

5.6 Multiple Data Sources

The Address Web Service provides access to multiple data sources including the DAD, the Denver street centerline, and United State Postal Service (USPS) national address database. The web service manages the integration and access to these data sources and this complexity is hidden to web service consumers.

Postal addresses are provided by the web service through a subscription to the MelissaDATA AddressObject (4). The AddressObject is a COM component that provides address validation against USPS addresses. The COM component is accessed by the web service using COM Interoperability. Denver's subscription includes bi-monthly updates to the USPS address database. Additionally, Denver subscribes to the GeoCoderObject, an additional component that provides latitude and longitude information for addresses at the zip-plus4 level.

Postal address information is integrated into address results that are returned by the address web service. When a valid address is identified in the DAD the result is then validated against the USPS address database. If an exact match is identified in the postal database, the relevant postal information is merged with the result from the DAD and the result is returned in a single response from the web service.

If the address request to be validated is outside of the City and County of Denver then the full postal address is returned. Additionally, postal addresses may be returned within Denver if no match is found within the DAD.

6 Address Web Service Architecture Overview

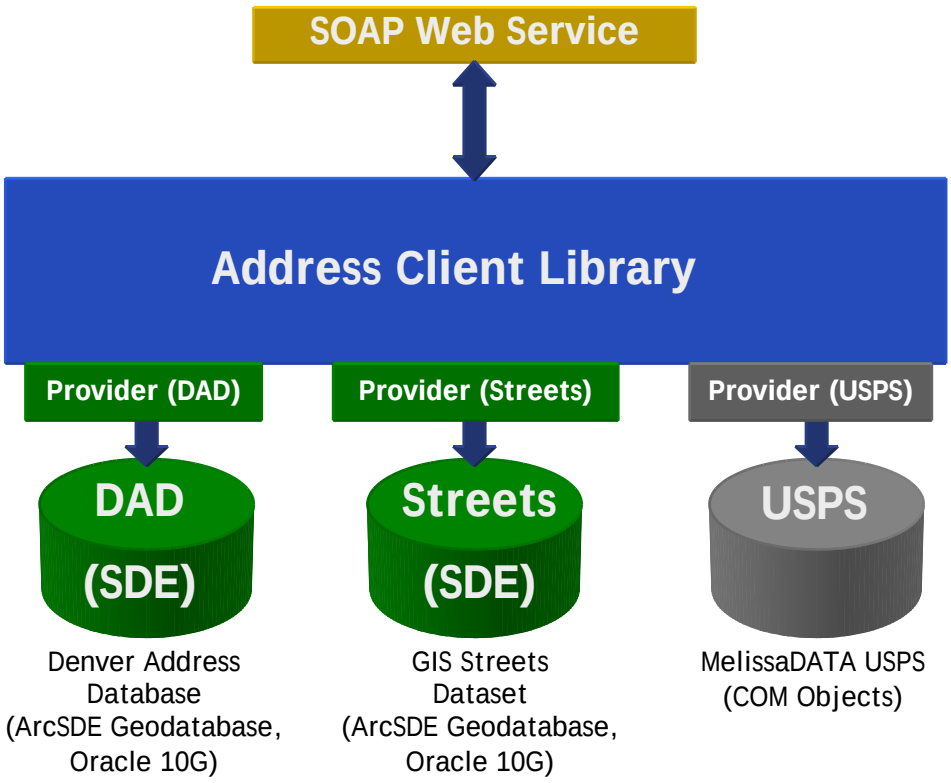

Figure 1: Major components and data sources of the Address Web Service.

6.1 SOAP Web Service

The SOAP Web Service is a light-weight endpoint that delivers the web service and provides the SOAP interface to client applications. The web service component provides the service WSDL and manages client authentication using the WS-Security Username Token Profile (5). The web service communicates with the Address Client Library to process all web service requests.

6.2 Address Client Library

The Address Client Library manages the web service business logic and manages the access to the web service data sources. The library is responsible for parsing, validation, and integration of results from the data sources.

The Address Client Library provides an interface for custom data providers. Data providers are responsible for communicating with each data source. The library manages requests to the data providers based on the type of request received. The library also manages the merging of results from multiple sources.

6.3 Providers

Providers are components that enable access to specific data sources.

The DAD provider enables address validation against the address geodatabase. The provider accesses the database via stored procedures and multi-versioned views.

The Streets provider enables intersection validation against the Denver street centerline. The provider accesses the street intersections feature class to perform intersection validation.

The USPS provider enables address validation against postal addresses. The provider accesses the MelissaDATA COM components using COM Interoperability.

7 Operations

Operations are the methods or functions that are provided by a web service. The Address Web Service provides the following operations:

Operation Name	Description
`SimpleAddressParse`	Parse an address into address parts. If the address cannot be parsed the input address will be returned unmodified.
`SimpleAddressValidate`	Validate an address. A default search definition is applied. If the address cannot be validated no result will be returned.
`SimpleAddressSearch`	Search for addresses. A default search definition is applied. If no matches can be found no results will be returned.
`AddressParse`	Parse an address into address parts. If the address cannot be parsed the input address will be returned unmodified.
`AddressValidate`	Validate an address. The search definition provided will determine the sources and types of addresses that will be used for validation. If the address cannot be validated no result will be returned.
`AddressSearch`	Search for addresses. The search definition provided will determine the sources and types of addresses that will be searched. If no matches can be found no results will be returned.
`AddressGetById`	Get an address from the Denver Address Database by AddressId.
`AddressGetPropertyInformation`	Get property information for a Denver Address Database address by providing the AddressId.
`SimpleIntersectionSearch`	Search for intersections. A default search definition is applied. If no matches can be found no results will be returned.
`IntersectionParse`	Parse an intersection into intersection parts. If the intersection cannot be parsed the input intersection will be returned unmodified.
`IntersectionSearch`	Search for intersections. The search definition provided will determine the sources that will be searched. If no matches can be found no results will be returned.

"Simple" operations only require basic parameter information (ex. `AddressLine1, AddressLine2, City, StateProvince, PostalCode`).

The non-"Simple" operations require an address or intersection object to be passed with any known address parts. These operations allow the consumer to specify a search definition that identifies the types of addresses that are to be matched in the service request (ex. A search definition may indicate that only assigned structure or utility addresses should be matched).

The service also provides operations that allow addresses and intersections to be parsed (parts are identified and standardized).

Addresses that are returned from the service include all known address parts. Element names are standardized to follow the FGDC Street Address Data Standard (6). For more information on standards applied to the DAD please review the paper "City of Denver Address: Modeling, Maintenance and Support" (1).

8 Sample Requests and Responses

The following examples provide sample SOAP requests and responses for address information from the service. Please note that the SOAP XML messages have been simplified for brevity and do not include namespaces, authentication information, and SOAP headers.

8.1 SimpleAddressValidate

The following request illustrates the ability of the service to validate an address based on free-form address input. The parser and validator recognize the parts of the address and return a standardized response from the geodatabase. The city and state names are also standardized and the address result is provided with integrated postal information (ex. postal code, plus4, place name, delivery address, delivery notes etc.). Latitude/longitude and Colorado state plane coordinates are also returned.

SOAP Request:

```
<soap:Envelope>
  <soap:Body>
    <SimpleAddressValidate>
      <addressLine1>201 west colfax ave.</addressLine1>
      <addressLine2></addressLine2>
      <city>denver</city>
      <stateProvince>colorado</stateProvince>
      <postalCode></postalCode>
    </SimpleAddressValidate>
  </soap:Body>
</soap:Envelope>
```

SOAP Response:

```
<soap:Envelope>
  <soap:Body>
    <SimpleAddressValidateResponse>
      <SimpleAddressValidateResult>
         <AddressLine1>201 W Colfax Ave</AddressLine1>
         <AddressLine2 />
         <CityStateZip>Denver, CO 80202-5329</CityStateZip>
         <City>Denver</City>
         <StateProvince>CO</StateProvince>
         <PostalCode>80202</PostalCode>
         <ZipPlus4>5329</ZipPlus4>
         <Country>USA</Country>
         <AddressNumberPrefix />
```

```
          <AddressNumber>201</AddressNumber>
          <AddressNumberSuffix />
          <StreetNamePreModifier />
          <StreetNamePreDirectional>W</StreetNamePreDirectional>
          <StreetNamePreType />
          <StreetName>Colfax</StreetName>
          <StreetNamePostType>Ave</StreetNamePostType>
          <StreetNamePostDirectional />
          <StreetNamePostModifier />
          <BuildingType />
          <BuildingIdentifier />
          <UnitType />
          <UnitIdentifier />
          <CompositeUnitType />
          <CompositeUnitIdentifier />
          <PostalPlaceName>Denver</PostalPlaceName>
          <CountyName>Denver</CountyName>
          <XCoordinate>3143361</XCoordinate>
          <YCoordinate>1694989</YCoordinate>
          <Longitude>-104.99021559</Longitude>
          <Latitude>39.74044472</Latitude>
          <PostalDeliveryLine>201 W Colfax Ave</PostalDeliveryLine>
          <PostalLastLine>Denver, CO 80202-5329</PostalLastLine>
          <PostalDeliverable>N</PostalDeliverable>
          <PostalDeliveryNotes>AAN1</PostalDeliveryNotes>
          <Source>DAD</Source>
          <AddressId>168091</AddressId>
          <Type>Street</Type>
          <Score>100</Score>
          <Parsed>true</Parsed>
          <ParsedRemainder />
        </SimpleAddressValidateResult>
    </SimpleAddressValidateResponse>
  </soap:Body>
</soap:Envelope>
```


8.2 SimpleAddressSearch

The following request illustrates the ability of the service to search for addresses when partial input is provided.

Multiple standardized responses are returned from the service and match scores are provided. Postal information and location information are included in the response.

SOAP Request:


```
<soap:Envelope>
  <soap:Body>
    <SimpleAddressSearch>
      <address1>201 col</address1>
      <address2></address2>
      <city>Denver</city>
      <stateProvince>CO</stateProvince>
      <postalCode></postalCode>
    </SimpleAddressSearch>
  </soap:Body>
</soap:Envelope>
```


SOAP Response:


```
<soap:Envelope>
  <soap:Body>
    <SimpleAddressSearchResponse>
      <SimpleAddressSearchResult>
       <AddressDetail>
         <AddressLine1>201 N Columbine St</AddressLine1>
         <AddressLine2 />
         <CityStateZip>Denver, CO 80206</CityStateZip>
```

```
          <City>Denver</City>
          <StateProvince>CO</StateProvince>
          <PostalCode>80206</PostalCode>
          <ZipPlus4 />
          <Country>USA</Country>
          <AddressNumberPrefix />
          <AddressNumber>201</AddressNumber>
          <AddressNumberSuffix />
          <StreetNamePreModifier />
          <StreetNamePreDirectional>N</StreetNamePreDirectional>
          <StreetNamePreType />
          <StreetName>Columbine</StreetName>
          <StreetNamePostType>St</StreetNamePostType>
          <StreetNamePostDirectional />
          <StreetNamePostModifier />
          <BuildingType />
          <BuildingIdentifier />
          <UnitType />
          <UnitIdentifier />
          <CompositeUnitType />
          <CompositeUnitIdentifier />
          <PostalPlaceName>Denver</PostalPlaceName>
          <CountyName>Denver</CountyName>
          <XCoordinate>3152563</XCoordinate>
          <YCoordinate>1687531</YCoordinate>
          <Longitude>-104.95765283</Longitude>
          <Latitude>39.71982483</Latitude>
          <PostalDeliveryLine>201 N Columbine St</PostalDeliveryLine>
          <PostalLastLine>Denver, CO 80206</PostalLastLine>
          <PostalDeliverable />
          <PostalDeliveryNotes />
          <Source>DAD</Source>
          <AddressId>162254</AddressId>
          <Type>Street</Type>
          <Score>50</Score>
          <Parsed>true</Parsed>
          <ParsedRemainder />
        </AddressDetail>
        <AddressDetail>
           <AddressLine1>201 W Colfax Ave</AddressLine1>
           <AddressLine2 />
           <CityStateZip>Denver, CO 80202-5329</CityStateZip>
           <City>Denver</City>
           <StateProvince>CO</StateProvince>
           <PostalCode>80202</PostalCode>
           <ZipPlus4>5329</ZipPlus4>
           <Country>USA</Country>
           <AddressNumberPrefix />
           <AddressNumber>201</AddressNumber>
           <AddressNumberSuffix />
           <StreetNamePreModifier />
           <StreetNamePreDirectional>W</StreetNamePreDirectional>
           <StreetNamePreType />
           <StreetName>Colfax</StreetName>
           <StreetNamePostType>Ave</StreetNamePostType>
           <StreetNamePostDirectional />
           <StreetNamePostModifier />
           <BuildingType />
           <BuildingIdentifier />
           <UnitType />
           <UnitIdentifier />
           <CompositeUnitType />
           <CompositeUnitIdentifier />
           <PostalPlaceName>Denver</PostalPlaceName>
           <CountyName>Denver</CountyName>
           <XCoordinate>3143361</XCoordinate>
           <YCoordinate>1694989</YCoordinate>
           <Longitude>-104.99021559</Longitude>
           <Latitude>39.74044472</Latitude>
           <PostalDeliveryLine>201 W Colfax Ave</PostalDeliveryLine>
```

```
          <PostalLastLine>Denver, CO 80202-5329</PostalLastLine>
          <PostalDeliverable>N</PostalDeliverable>
          <PostalDeliveryNotes>AAN1</PostalDeliveryNotes>
          <Source>DAD</Source>
          <AddressId>168091</AddressId>
          <Type>Street</Type>
          <Score>28</Score>
          <Parsed>true</Parsed>
         <ParsedRemainder />
       </AddressDetail>
      </SimpleAddressSearchResult>
    </SimpleAddressSearchResponse>
  </soap:Body>
</soap:Envelope>
```

9 Web Service Consumers

9.1 3-1-1 Customer Relationship Management

Denver's 3-1-1 customer relationship management (CRM) system was the initial consumer of the Address Web Service and was a key business driver for the development of Denver's new addressing system. The PeopleSoft CRM application provides web service integration capabilities and is able to directly leverage the Address Web Service.

3-1-1 Call Center call agents enter addresses into the CRM application while speaking with the constituent. The address is immediately validated against the Address Web Service. Validating addresses at point-of-entry provide many benefits including the ability to request further clarification from the customer if the address does not immediately validate.

Validated address information is captured in CRM, including the standardized address and the address location. The county information returned is used to determine whether a case should be opened or if the caller should be transferred to another jurisdiction. The address validation information captured with the case provides the ability to perform mapping and analysis of 3-1-1 cases (Figure 2).

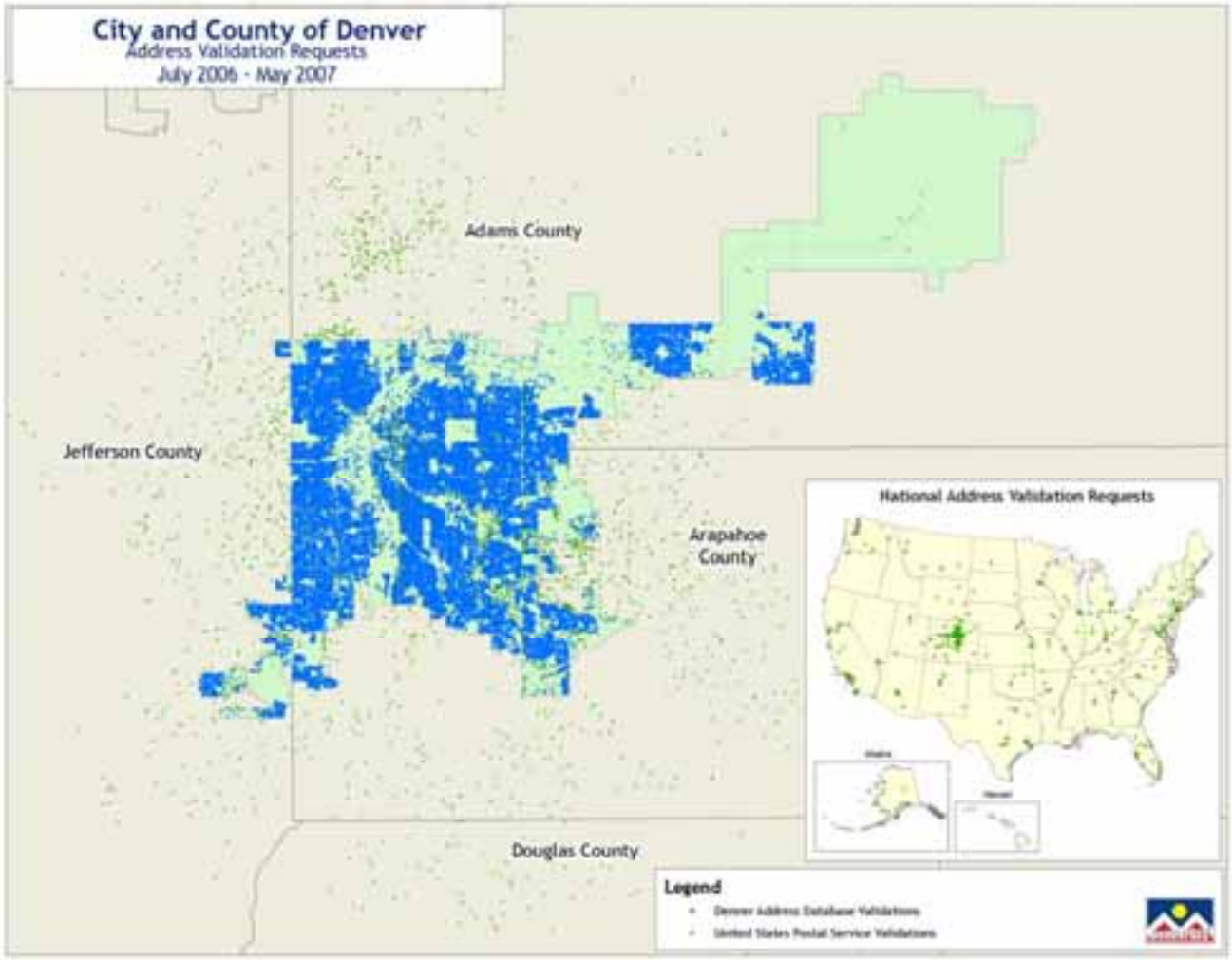

Figure 2: This map illustrates address validation requests that were received by the 3-1-1 CRM system over an 11 month period. The blue dots represent matches from the DAD and the green triangles represent matches from the USPS. A large number of calls are received from surrounding counties and calls have been received from most US states.

The 3-1-1 system is also available to constituents through the City and County of Denver's public web site (http://www.denvergov.org/311). Addresses entered through this channel are also validated against the web service (Figure 3) and constituents are provided a notice when the case address provided is not within the City and County of Denver (Figure 4).

Figure 3: 3-1-1 Self Service allows constituents to create cases online. The above example shows an address that has been validated using the Address Web Service. The address match shown is returned live from the multi-versioned address geodatabase.

Figure 4: The above example shows an address that has been validated that is not within the City and County of Denver. The customer is notified that the address that was provided is not with Denver County and that a 3-1-1 case cannot be opened for this address.

9.2 Permitting

Denver's permitting system requires accurate addressing for identification of property and issuance of permits. The permitting system was previously connected directly to the CAD database. All new permitting applications are being integrated with the Address Web Service. Addresses that are assigned in the address geodatabase are immediately available to permitting applications.

9.3 Enterprise Document Management

Denver's document management scanning system has been integrated with the Address Web Service to provide the ability to associate valid addresses with documents. Valid address identifiers are added to document metadata for indexing and searching document collections.

9.4 Other

Additional applications are using the Address Web Service to perform address validation. These include applications for Excise and License, Public Safety, Environmental Health and Wastewater Management. Additional web service integrations are in development at the time of this writing including integration with Denver's intranet and internet GIS applications.

10 Lessons Learned

10.1 Geodatabase Versioning

The DAD is maintained in an ArcSDE Enterprise Geodatabase residing in an Oracle 10G database. The Address Maintenance Application (1) automatically reconciles and posts address changes on completion of edits. Addresses are made available to the web service through stored procedures that access multi-version views. In initial tests it was identified that address changes that were reconciled and posted were not immediately made available to the web service.

When changes are reconciled and posted to "SDE.DEFAULT" the state of the version is updated. When the first query of a multi-version view occurs on a connection the SDE.VERSION_UTIL.SET_CURRENT_VERSION procedure is executed and the current state is set. Subsequent requests will access this state unless the SET_CURRENT_VERSION procedure is executed for each request to set the state to the most current state of the version.

Executing the SET_CURRENT_VERSION stored procedure within the procedures that are used to access the multi-version views ensure that the latest changes to the version are returned in web service response.

10.2 Spatial Database Indexes

Basic database indexes can be created on SDE tables and generally are necessary to optimize query performance. Additional Oracle index types are available to further improve performance including function-based indexes. Function-based indexes improve the performance of queries that use functions such as case insensitive searches or searches that contain mathematical expressions.

The address validation stored procedures that were developed required expressions to perform fuzzy matching and case-insensitive queries. Additional indexing (including function-based indexes) were required to optimize the performance of address validation queries.

When SDE tables are registered as versioned, the indexes on the table are applied to the generated adds and deletes tables. If Oracle function-based indexes are present the table cannot be registered as versioned. To successfully register the table the function-based indexes must be dropped and then created after the registration is completed. To maximize performance, these indexes can also be applied to the appropriate adds table after registering as versioned.

11 Conclusion

The Address Web Service has been in production for over a year and is being used successfully by multiple enterprise applications in the City and County of Denver.

Address validation has proven to be critical to the implementation of Denver's 3-1-1 system and provides City leadership with the mapping and spatial analysis capabilities required to improve customer service for Denver's residents, businesses and visitors.

The loose-coupling of the Address Web Service has allowed geodatabase enhancements to be made with no impact to consumers. During the past year, the address geodatabase has been migrated to a new server and no changes were required to be made by consumers.

The address service monitoring database supports monthly reporting and provides managers with key performance metrics to support service level agreements. For the current year (Jan 2007 through May 2007) the address service has received over 280,000 requests with an average response time of 0.22 seconds.

Web services provide a means for integrating geographic information systems with non-GIS applications and promote the sharing of location-based services with enterprise software applications. As an increasing number of commercial software applications are delivered with web services support, it is anticipated that GIS web services will play an increasingly important role in the enterprise.

12 Acknowledgements

The Denver Address Database and Address Web Services are developed and maintained by the City and County of Denver Technology Services Geographic Information Systems team (DenverGIS). The Geospatial Applications group is directed by David Luhan.

For more information please visit http://www.denvergov.org/gis or contact:

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

References

(1) City and County of Denver Geographic Information Systems "City of Denver Addressing: Modeling, Maintenance and Support"
ESRI International User Conference 2007

(2) Wikipedia "Service-Oriented Architecture"
http://en.wikipedia.org/wiki/Service-oriented_architecture

(3) World Wide Web Consortium (W3C) "SOAP Version 1.2 Part 0: Primer"
W3C Recommendation 24 June 2003
http://www.w3.org/TR/2003/REC-soap12-part0-20030624/

(4) MelissaData "AddressObject"
http://www.melissadata.com

(5) OASIS "Web Services Security Username Token Profile 1.0"
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

(6) Federal Geographic Data Committee "Street Address Data Standard"
http://www.fgdc.gov/standards/projects/FGDC-standards-projects/street-address/index_html