

**NSDI Cooperative Agreements Program
Framework Client Development
FINAL PROJECT REPORT**



**TITLE: CREATING AN ENTERPRISE MUNICIPAL BOUNDARY
DATA SERVICE USING THE OGC WFS SPECIFICATION**

AGREEMENT NUMBER: 05HQAG0117

REPORT DATE: November 30, 2008

ORGANIZATION: Land Management Information Center
Minnesota Department of Administration
300 Centennial Office Building
658 Cedar Street; St. Paul, MN 55155

PROJECT LEADERS: Christopher Cialek
651.201.2481
chris.cialek@state.mn.us

Jim Dickerson
651.201.2484
jim.dickerson@state.mn.us

**COLLABORATING
ORGANIZATION:** Minnesota Department of Natural Resources
MIS Bureau; DNR Central Office
500 Lafayette Road; St. Paul, MN 55115

TABLE OF CONTENTS

PROJECT INFORMATION SHEET	1
PROJECT NARRATIVE.....	5
Project goals.....	5
Discussion.....	5
Project results.....	7
Project result details.....	7
Municipal boundaries.....	7
Web Feature Service.....	7
GeoMoose-based web client.....	9
Sustainable service.....	9
STATUS OF YOUR DATA ACCESS ACTIVITIES.....	10
STATUS OF FRAMEWORK CLIENT DEVELOPMENT	10
PROJECT MANAGEMENT.....	11
FEEDBACK ON CAP.....	12
APPENDICES	
A. Development of the WFS service using Mapserver with ArcSDE.....	13
B. Test Tools Used to Validate WFS Service.....	15
C. GeoMoose map files with description.....	16
D. WFS URLs used in this project.....	20
E. Supporting FGDC Framework Data Layer Publishing Using MapServer.....	20
F. ArcGIS Modeling for Updating Governmental Unit Boundaries.....	21

FIGURES

1. *A process for maintaining a geospatial database, which records municipal boundary annexation change data, has been developed in Minnesota through a collaboration of three state offices..... 5*
2. *FGDC CAP 05HQAG0117 provided resources for LMIC to develop a Web Feature Service through University of Minnesota MapServer web mapping software and to display results through a WFS client built upon the GeoMoose framework. 6*
3. *GeoMoose-based web client with GML extraction tool icon circled..... 8*
4. *Process flow diagram illustrating procedures used to create an integrated Government Unit Boundary (MNGUB) geospatial dataset using ESRI's ModelBuilder application. 25*

PROJECT NARRATIVE

Summarize the project activities. Include its accomplishments, successes strengths and weaknesses, further challenges, and collaboration activities, as appropriate.

Project goals

The Minnesota Land Management Information Center (LMIC) was awarded a 2005 Cooperative Agreements Program grant to pursue the FGDC's efforts in promoting interoperability and consolidated geospatial data discovery and delivery through the use of emerging Web Feature Service protocols. To demonstrate that capability, LMIC proposed to refine the state's municipal boundary annexation authorization and documentation procedures in a way that integrates: 1) emerging Framework content standards, 2) the Web Feature Services capabilities of Minnesota MapServer software, and 3) the state's GIS coordinating council recommendations for an integrated enterprise solution that serves local, state and federal government needs from a single authoritative source.

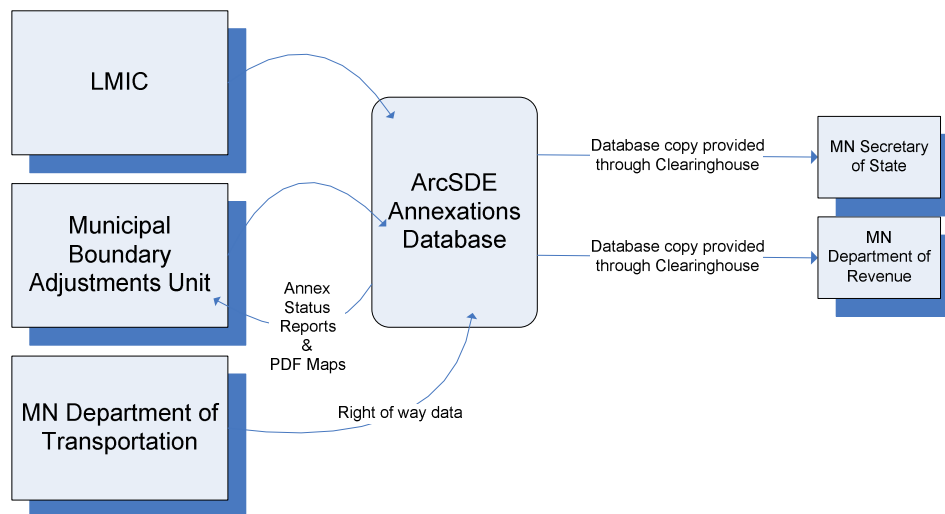


Figure 1. A process for maintaining a geospatial database, which records municipal boundary annexation change data, has been developed in Minnesota through a collaboration of three state offices.

Discussion

Coordinating the maintenance of an authoritative municipal boundary geospatial database has been particularly challenging in Minnesota. A process, partially funded by the state's former Planning Agency, the Department of Revenue and the Office of the Secretary of State, was developed prior to this CAP project and incorporated legal descriptions of annexation requests managed by the Office of Administrative Hearings' Municipal Boundaries Adjustment Unit. Roadway right-of-way data, provided by the Department of Transportation were integrated into an enterprise geospatial database. Although the

status of annexations at any point in time is difficult to assess, the geospatial rendering of all the best boundary information is updated periodically and distributed through the Minnesota Geographic Data Clearinghouse (Figure 1.). Primary customers are the Secretary of State's Office, which uses the data to adjust voting districts, and the Department of Revenue, which uses them to update augmentations to cities and counties with special sales tax districts.

Because the demand to access current municipal boundaries figures importantly in the business functions of a number of state agencies, an effort to better expose a unified version of these data through web services has been a goal of particular interest in developing a more effective and efficient state GIS enterprise. Resources provided through this CAP were used to create a functioning, OGC-compliant Web Feature Service and provide access to that service, not only through direct software WFS connectors, but also through a locally-developed WFS client (Figure 2.). These data services are currently being tested by the State Legislative Coordinating Commission's GIS Office and a number of public and local government clients. While boundary data still are delivered to the Department of Revenue through traditional means as called for through contractual obligation, data delivery through the client (Figure 3.) is being considered by the Secretary of State's Office.

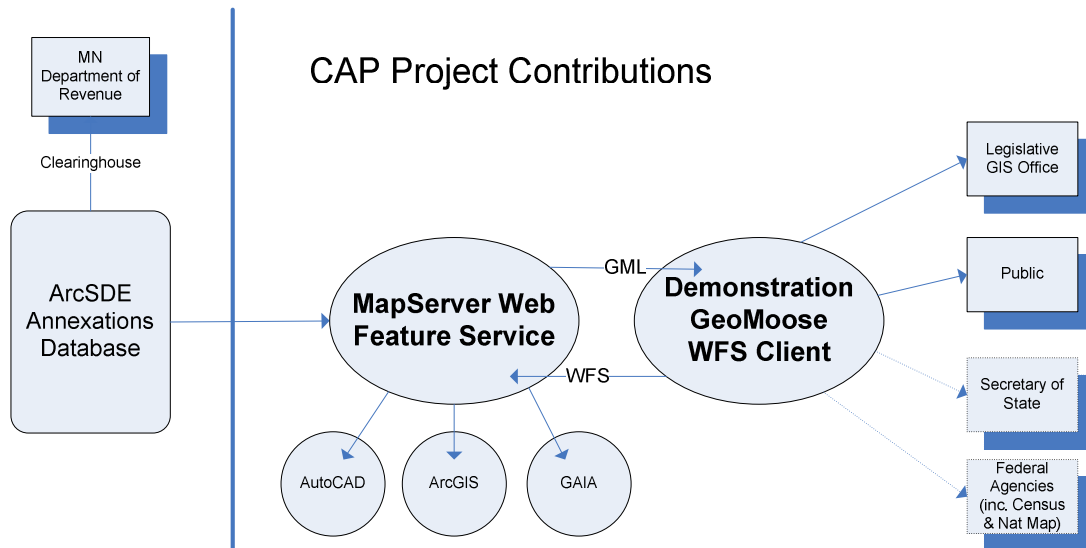


Figure 2. FGDC CAP 05HQAG0117 provided resources for LMIC to develop a Web Feature Service through University of Minnesota MapServer web mapping software and to display results through a WFS client built upon the GeoMoose framework.

Project results

This project set out to demonstrate that MapServer (<http://mapserver.gis.umn.edu/>) could use WFS to serve Framework data from ArcSDE. The data chosen were for municipal boundaries; one layer developed with 2003 annexation information, one from 2006 and a final version compiled for 2008. It was found that data could be accessed by MapServer from ArcSDE and published through the WFS without loss of fidelity. The project was successful in developing a functional WFS capable of exposing a vastly improved (but not yet authoritative) statewide municipal boundary geospatial data file than had previously existed through the coordinated efforts of three state agencies. A web site was developed leveraging the results of a previous CAP project: a web client JavaScript framework called GeoMoose (<http://www.geomoose.org>).

Project result details

Municipal boundaries

For this project's products to be useful, they needed to be able to provide access to a Framework dataset that was frequently updated and in demand by the state's GIS community. The dataset used was a state-wide layer of city, township, and unorganized territory boundaries, collectively called Minnesota Governmental Unit boundaries or MNGUB. This dataset would be used primarily by local government bodies that have made changes to their boundary via the annexation process. These users will be able to employ the updated MNGUB by accessing it as a WFS service via client tools, such as AutoCAD, ArcGIS and Carbon Tools Gaia.

This project also makes it possible to download GML for selected features by making WFS requests directly to the WFS service. This is demonstrated through a LMIC-developed GeoMoose-based Web Client, which includes a tool for making a bounding box selection of displayed features and saving the resulting GML to a file (Figure 3.).

Web Feature Service

MapServer Client: The MapServer mapfile was created with connection parameters using an ArcSDE server as the data repository and providing both WMS- and WFS-compliant output. The map file connection string was written to link to ArcSDE from MapServer. Several trials were needed to determine correct connection string syntax yielding a successful WFS service. The WFS service was tested using Carbon Tools GAIA, which verified its ability to serve and download the data in GML 3.0 format and load it into an application. The mapfile was edited to include a WMS service as well, for web viewing purposes. The WFS service was then tested with the following products:

Carbon Tools GAIA: Gaia was tested with the WFS service, both version 2.x and 3.x. We verified that the software could access the WFS server, display layers and extract GML code. Gaia was also used to read GML code that was downloaded

independently via the GeoMoose web page. Both versions successfully displayed the data and read GML.

AutoDesk MapGuide Enterprise 2008: It was found that, while multiple municipality layers were read, only one was displayed. This was discussed with AutoDesk representatives at the FOSS4G 2007 Conference in Victoria, Canada where the problem was demonstrated. AutoDesk representatives acknowledged the problem and stated that a fix would be available in the next product release.

ESRI ArcGIS with ArcGIS Framework Extension: This extension was developed by the *Center for Spatial Information Science and Systems* at George Mason University (<http://csiss.gmu.edu/>). The first releases of the extension did not work with our WFS service but we worked with Peisheng Zhao and a later release was able to display our WFS services correctly. We also were able to export (download) the WFS layer successfully.

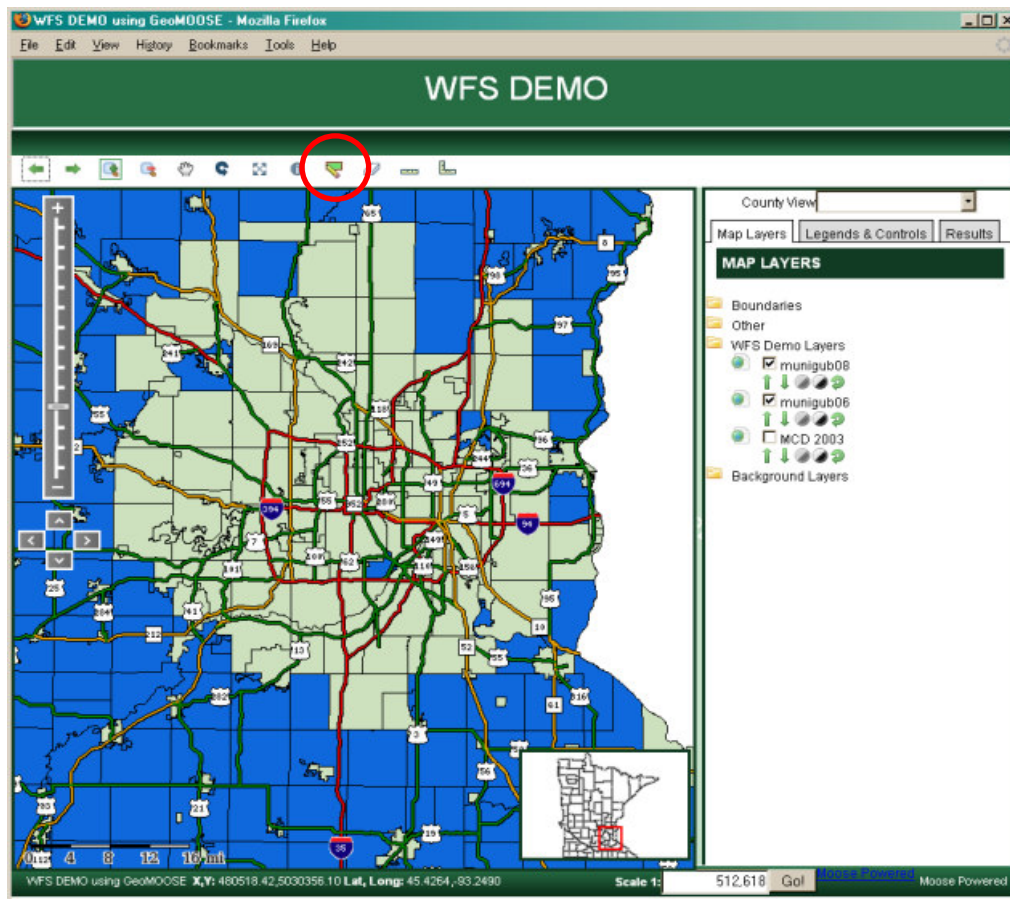


Figure 3. GeoMoose-based web client with GML extraction tool icon circled

GeoMoose-based web client

In order to promote the WFS service to those in the GIS and local government communities who do not have the clients described above, we created a web-client that could display and download data in the WFS. The 2006 CAP Grant OpenMNND project included the development of a reusable OGC web-client named GeoMoose. It was decided to take advantage of this environment to create the WFS client for this project (Figure 3.). A tool was developed for GeoMoose by LMIC to select and download the GML data returned by the WFS server. This allows us to demonstrate a capability to download and import GML data to local government users.

Sustainable service

The value of this WFS service is directly related to the currency of the data it exposes. Because LMIC maintains periodic boundary changes due to annexations for the state, the Center has the ability to update a geospatial statewide municipality file more frequently than other state entities.

This can be a labor intensive task, however, requiring unusual coordination because boundary changes are edited by multiple agencies and then consolidated at LMIC. An effort was launched, therefore, to automate the data updating process through the use of ESRI's ArcGIS modeling tool. A process for merging several data sources was developed and captured by the ArcGIS model to facilitate up-to-date data publication. See Appendix F for details.

STATUS OF YOUR DATA ACCESS ACTIVITIES

What Framework data theme(s) will be accessed under this project?

Governmental Unit Boundaries

What is the data volume of Framework data anticipated for access (geographic coverage, dataset size, feature count)?

MNGUB08

Geographic coverage: full state; Minnesota

Dataset size: 9.35 MB

Feature count

Cities: 854

Townships: 1,848

Unorganized Territories: 94

Who are the primary organizations providing data for this project?

Minnesota Department of Transportation

Minnesota Office of Administrative Hearings Municipal Boundaries Adjustment Unit

Minnesota Department of Administration

STATUS OF FRAMEWORK CLIENT DEVELOPMENT

What is the status of software development?

The development of a web-based client has been completed. The web-client called GeoMoose and developed under CAP 06HQAG0115 was used as the client software environment in order to leverage previous efforts and promote rapid development. With GeoMoose as a template, our efforts focused on developing a tool to download WFS-served data based on a user's interactive selection.

How will the client software be evaluated and quality-assured?

Because we based our development on GeoMoose, we have benefited from ongoing efforts by that resource's developers and users to test and improve the base product. Our own testing has focused on using the web-client to download and import GML into end-user applications such as ESRI's ArcGIS and CarbonTools Gaia. Details are provided in Appendix B.

Describe your experience and purpose in accessing the data services?

Using our web-based client, we are able to demonstrate to end-users how to access our WFS service and display the results in order to introduce those users to GML. Using desktop applications such as ArcGIS and Gaia, we can demonstrate connecting to our WFS service for real-time data access. In all cases, data retrieval from our Mapserver/ArcSDE configuration is as quick as for other services.

Describe any internal or external users that are using this client.

Internally, state agencies that can benefit from this WFS/WMS service include the Departments of Revenue, Transportation and Administration, as well as the Office of the Secretary of State and the Legislative Coordinating Commission's GIS Office. External organizations that benefit from this service include city and county government.

Identify plans for the promotion and distribution of this software.

It is intended that our organization continue to update and maintain the Governmental Unit Boundaries municipalities dataset. In order to promote its use, demonstrations are planned for several state agencies. In addition, our home webpage will direct viewers to a project web page (<http://www.lmic.state.mn.us/chouse/wfs.html>) describing the availability and use of this dataset.

PROJECT MANAGEMENT

Will this project's activities continue in the future?

Yes, we intend to maintain and grow WFS offerings within the Minnesota Geographic Data Clearinghouse and periodically update MNGUB as agency support allows.

Describe the next phase in your project.

We would like to gauge interest among Minnesota GIS users in exposing and using more layers via WFS. This can be done through technology demonstrations and newsletter articles. Evidence exists leading us to believe WFS will be widely used in the future as more robust software connectors are developed and exposure to its advantageous features expands. Already, the Metropolitan Council has increased the number of geospatial data layers available as Web Feature Services. As recently as last week, a forum devoted to web service development in the Twin Cities region identified GIS information sharing via WFS as a high priority investment area over the next two years [results of the MetroGIS *Geospatial Applications and Web Services Needs Forum*, held 11/20/2008; St. Paul, MN; www.metrogis.org/teams/workgroups/shared_app/forum_11-20-08/Forum_Turnaround_Document.pdf].

Requirements (more technical assistance, software, other?)

We will seek out more examples of WFS's practical value to geospatial data users in Minnesota. That evidence should help provide demand to drive further data exposure. Toward that end, the availability of effective, low cost connector extensions for popular GIS software products will be important, as will funding to promote and monitor feature service education and use.

What areas need work?

Regarding our specific solution, as WFS advances, MapServer will need to implement changes to keep up. A dependable and stable GML standard would be of great value in promoting WFS solutions.

FEEDBACK ON CAP

What are the program strengths and weaknesses?

For years the FGDC Cooperative Assistance Program has provided seed funding that leverages the resources of FGDC partner organizations and allows them to develop and implement projects and programs that advance the tenets of the NSDI in their regions. We hope the FGDC and its parent agency continue to value this progressive relationship as much as we at LMIC do.

Where does the program make a difference?

CAP provides us the opportunity to explore important and potentially valuable technical and policy topics, ones that we would not otherwise have sufficient resources to research and exploit. In the case of this project, our CAP grant allowed us to develop services that, two years ago, seemed cutting edge, but today are being acknowledged as critically important. It also has allowed us to keep current with advances being made in the Open Source world and to better understand how those advances are being utilized among commercial GIS software vendors.

Was the assistance you received sufficient or effective?

Monthly teleconferences provided useful opportunities to discuss our progress and learn about the progress of others.

Are there factors that are missing or additional needs that should be considered?

Although ad hoc project critiques that were part of the monthly telecons were helpful, we would benefit by receiving more structured, in-depth feedback on our project from FGDC staff members, other Category 2 grant recipients and their collaborating organizations. Perhaps this final report will generate interest in that type of extended examination.

Are there program management concerns that need to be addressed? Time frame?

The glacial pace at which standards – critical to the success of the implementation plans of CAP partners – are developed and ratified at the federal/national/international level.

If you were to do this again, what would you do differently?

We originally based project success measures on a few factors out of our control: the successful publication of authoritative GML and GUB standards; agreement among state agencies of an enterprise municipal boundary annexation mapping procedure; more complete development of readily available WFS connector tools by major geospatial processing software vendors. While the technical challenges within the scope of this project were rather straightforward and comfortably met, those administrative hurdles resulted in project schedule slippage, forcing us to re-scope and adjust project deliverables. In the end, however, the project managers believe that this effort has provided a positive learning experience and that the software tools developed for it will be maintained, improved and implemented within the state's Clearinghouse environment to serve the greater GIS user community in Minnesota.

Appendix A. Development of the WFS service using University of Minnesota MapServer with ESRI ArcSDE

UMN MapServer was updated under CAP Grant 04HQAG0165 to provide WFS support and return GML version 3.0. LMIC's WFS CAP grant utilized this new feature in conjunction with MapServer's access to ArcSDE to enable a direct means of distributing data from the ArcSDE Geodatabase to the Internet via WFS.

In order to do this, a mapfile definition was created with a connection string to ArcSDE, as the data source, and a metadata definition that enabled both WMS and WFS output. Connection strings were tested for ArcSDE using both Oracle and MS SQL databases.

Below is the layer definition portion of the mapfile that implements this behaviour (boldface highlights specific parameters that implement WFS and WMS for this layer):

```
LAYER
  NAME mngub08
  TYPE POLYGON

# Connection string for ArcSDE/MS SQL,
# CONNECTION "hostname,port:5151,GIS,owner,password"
# CONNECTIONTYPE SDE
# DATA "GIS.owner.mmgub08,SHAPE,SDE.DEFAULT"

# Connection string for ArcSDE/Oracle
# CONNECTION "hostname,port:5151,database,owner,password"
  CONNECTION "aquarius.lmic.state.mn.us,port:5151,GIS,owner,password"
  CONNECTIONTYPE SDE
# DATA "OWNER.Layername,Layertype,sde.DEFAULT"
  DATA "DATAOWN.MNGUB08,SHAPE,sde.DEFAULT"
  STATUS ON
  CLASSITEM "NAME"
  CLASS
    NAME "Municipal Boundaries 2008"
    EXPRESSION /./
    COLOR 15 105 220
    OUTLINECOLOR 0 0 0
    TEMPLATE "mcd90py2.html"
  END #Class
  HEADER "mcd90py2_header.html"
  FOOTER "mcd90py2_footer.html"
  TEMPLATE "blank.html"
  DUMP TRUE # allow GML export
```

```
METADATA
  WMS_TITLE "MN Municipal Boundaries, 2008"
  WMS_ABSTRACT "This layer is based on the 2000 Municipal Basemap layer and includes updates from annexation."
  WMS_SRS "EPSG:26915"
  wms_extent "189783.511127,4816314.930014,701222.390324,5433610.129821"

  WFS_TITLE "MN Municipal Boundaries, 2008"
  WFS_ABSTRACT "This layer is based on the 2000 Municipal Basemap layer and includes updates from annexation."
  WFS_SRS "EPSG:26915"
  wfs_extent "189783.511127,4816314.930014,701222.390324,5433610.129821"

  wfs_keywordlist "Minnesota Municipal Boundary City Annexation 2008"
  gml_include_items "NAME"
END #metadata
END #Layer
```

Appendix B.

Test Tools Used to Validate WFS Service

In order to validate GML produced by the MapServer WFS service, several client tools were used to insure that the results were well-formed. Over the course of the project, new tools became available and were used to further test client/server actions. This testing revealed some product problems that were discussed with each respective vendor, resulting in product changes that benefited the whole GIS community.

Product: **Carbon Tools GAIA**

Contact: Jeff Harrison

<http://www.thecarbonproject.com/gaia.php>

CarbonTools created the GAIA client software for use with OGC services. It was used from the beginning of the project to test our WFS server development. In the final stages of the project it provided a ready means to demonstrate the WFS capabilities to other agencies.

Product: **George Mason University ArcGIS Framework Extension**

Contact: Dr. Peisheng Zhao

<http://csiss.gmu.edu/products/arcgisext.html>

This project (CAP 07HQAG0098), undertaken by the George Mason University Center for Spatial Information Science and Systems, developed an extension for ArcGIS that consumes OGC services and integrates them with other ArcGIS layers. This ArcGIS extension was tested successfully with our WFS server.

Product: **ESRI ArcGIS Data Interoperability Extension**

<http://www.esri.com>

ESRI created an extension to consume OGC services as well as other data types. This extension was tested successfully with our WFS server by ESRI technical support personnel during the 2006 ESRI International User Conference in San Diego.

Product: **AutoDesk 3D Map Studio 2007**

<http://usa.autodesk.com>

This product was tested in-house and while it would read layers from the WFS service, some deficiencies were found. As a follow-up, at the 2007 FOSS4G conference, AutoDesk technical staff assisted in troubleshooting the application. The problems were confirmed and assurances were given that they would be addressed in the next release. This release has not been tested by LMIC.

Appendix C.

GeoMoose map files with description

The development of the GeoMoose client in the OpenMNND project (CAP 06HQAG0115) provided a means to quickly develop a client for our new WFS service. A WFS client allows us to demonstrate the utility of the service for downloading layers.

WFS services created by LMIC were added to the GeoMoose client by modifying the GeoMoose mapfile definition. Below are the files used to modify the standard GeoMoose package to accommodate viewing the WFS service and downloading GML from it.

Server files

The following php routine is called by GeoMoose to extract GML from the WFS. It returns an html page containing GML describing features in the area defined by the bounding box expressed in a user request.

```
<?php
// Set defaults
$_url = 'http://geoint.lmic.state.mn.us/cgi-bin/mapserv';
$_map = '/home/httpd/html/cap/gubs.map';
$_version = '1.0.0';
$_typename = 'mngub08,mngub06,mcd2003';

// Get in the information from the request
$coordinates = $_REQUEST['coords'];

if (isset($_REQUEST['url']))
    $_url = $_REQUEST['url'];
if (isset($_REQUEST['map']))
    $_map = $_REQUEST['map'];
if (isset($_REQUEST['version']))
    $_version = $_REQUEST['version'];
if (isset($_REQUEST['typename']))
    $_typename = $_REQUEST['typename'];

$coords = array();
$coords = explode(' ', $coordinates);
$p1 = explode(',',$coords[0]);
$p2 = explode(',',$coords[2]);
$x1 = min($p1[0],$p2[0]);
$x2 = max($p1[0],$p2[0]);
$y1 = min($p1[1],$p2[1]);
$y2 = max($p1[1],$p2[1]);

$fileURL =
sprintf("%s?map=%s&service=WFS&version=%s&request=GetFeature&typename=%s&BBOX=%d,%d,%d,%d",$_url,$_map,$_version,$_typename,$x1,$y1,$x2,$y2);
```



```

$xmlfile = file_get_contents($fileURL,FILE_TEXT);

// Use session_id to generate unique file name
session_start();
$name = '/tmp/gml_'.session_id().'.xml';

// Note: This will destroy the session, and not just the session data!
if (session_id != "" || isset($_COOKIE[session_name()]))
    setcookie(session_name(), "", time()-42000, '/');
session_destroy(); // Done with SESSION VARIABLES...

// write file
if ($xmlfile)
{
    $fp = fopen('/home/httpd/html'.$name, 'w');
    fwrite($fp, $xmlfile);
    fclose($fp);
}

// link to the results
echo "<a href='\".$name.\"' target='_blank'>view xml file</a>";
?>

```

GeoMoose files

This is the portion of the mapfile definition for GeoMoose that defines the WMS to be used in the web application. It contains the WMS call to the server for displaying the WFS layers.

```

<?xml version="1.0"?>

<!-- The title attribute will be displayed in the title bar and in the window title.
- stylesheet is a reference to the CSS file that you want to use to render the interface.
- root is the document root for the shared/images/toolbar and other files.
-->
-->
<mapbook>
  <group title="Basic Tools" invisible="true">
    <service title="Previous Zoom" type="internal" default="true" locked="true"
command="NavHistory.previous()" icon="shared/images/toolbar/arrow_left.jpg"
highlight="shared/images/toolbar/arrow_left_selected.jpg" selectable="false"/>
    <service title="Next Zoom" type="internal" default="true" locked="true"
command="NavHistory.next()" icon="shared/images/toolbar/arrow_right.jpg"
highlight="shared/images/toolbar/arrow_right_selected.jpg" selectable="false"/>
    <service name="zoomIn" selected="true" title="Zoom In" type="internal" default="true"
locked="true" command="Events.triggerEvent('zoom',.5)" key="i"
icon="shared/images/toolbar/zoom_in.jpg" highlight="shared/images/toolbar/zoom_in_selected.jpg"/>
    <service name="zoomOut" title="Zoom Out" type="internal" default="true" locked="true"
command="Events.triggerEvent('zoom',2)" key="o" icon="shared/images/toolbar/zoom_out.jpg"
highlight="shared/images/toolbar/zoom_out_selected.jpg"/>
    <service name="zoomPan" title="Pan" type="internal" default="true" locked="true"
command="Events.triggerEvent('zoom',1)" key="p" icon="shared/images/toolbar/pan.jpg"
highlight="shared/images/toolbar/pan_selected.jpg"/>

```

```

    <service name="refreshLayers" title="Reload Layers" type="internal" default="true"
locked="true" command="Events.triggerEvent('refreshlayers')" icon="shared/images/toolbar/refresh.jpg"
selectable="false"/>
    <service title="Full State View" type="internal" invisible="true" default="true" locked="true"
command="Map.zoomToViewString('189783.560000,4816309.330000,761653.524114,5472346.500000');
" icon="shared/images/toolbar/zoom_full.jpg" highlight="shared/images/toolbar/zoom_full_selected.jpg"
selectable="false"/>
    <service title="Identify" default="true" cursor="url(cursors/identify2.cur), auto"
icon="shared/images/toolbar/identify.jpg" highlight="shared/images/toolbar/identify_selected.jpg"
locked="true" shape="point">
        <!--url>php/identify.php</url-->
        <url>/cgi-bin/mapserv</url>
        <input type="hidden" name="map"
value="/home/httpd/mapfiles/maps_statedemo/identify.map"/>
        <input type="point" name="mapxy" delim=" "/>
        <input type="extent" name="imgext" delim=" "/>
        <input type="hidden" name="layers" value="CountyMap"/>
        <input type="imagesize" name="imgsize" delim=" "/>
        <input type="hidden" name="mode" value="nquery"/>
        <input type="hidden" name="tolerance" value="0"/>
    </service>

```

```

<service title="WFS Select Feature" default="true" locked="true" cursor="url(cursors/select.cur),
auto" shape="box"
type="inspect-search" icon="shared/images/toolbar/draw_polygon.jpg"
highlight="shared/images/toolbar/draw_polygon_selected.jpg">

```

```

    <url>php/select_wfs.php</url>
    <input type="multipoint" name="coords"/>
    <input type="hidden" name="url" value="http://geoint.lmic.state.mn.us/cgi-bin/mapserv"/>
    <input type="hidden" name="map" value="/home/httpd/html/cap/gubs.map"/>
    <input type="hidden" name="version" value="1.0.0"/>

```

```

    <input type="select" name="typename">
        <option value="mngub06">Municipal GUB 06</option>
        <option value="mngub08">Muni Gub 08</option>
        <option value="mcd2003">MCD 2003</option>
        <option value="mngub08,mngub06,mcd2003">All</option>
    </input>
    <input type="hidden" name="output" value="xml"/>
</service>

```

```

<group title="WFS Demo Layers" expand="true">
    <map title="mngub08" default="true" reference="false" zindex="-1">
        <file>http://geoint.lmic.state.mn.us/cgi-
bin/mapserv?map=/home/httpd/html/cap/gubs.map&REQUEST=GetMap&SERVICE=W
MS&VERSION=1.1.1&LAYERS=mngub08&FORMAT=image/png&TRANSPARENT=TRUE&SRS=EPSG:26915&STYLES=&BBOX=%MINX%,%MINY%,%
MAXX%,%MAXY%&WIDTH=%WIDTH%&HEIGHT=%HEIGHT%</file>
    </map>
    <map title="mngub06" default="true" reference="false" zindex="-1">
        <file>http://geoint.lmic.state.mn.us/cgi-
bin/mapserv?map=/home/httpd/html/cap/gubs.map&REQUEST=GetMap&SERVICE=W
MS&VERSION=1.1.1&LAYERS=mngub06&FORMAT=image/png&TRANSPARENT=TRUE&SRS=EPSG:26915&STYLES=&BBOX=%MINX%,%MINY%,%
MAXX%,%MAXY%&WIDTH=%WIDTH%&HEIGHT=%HEIGHT%</file>

```

```
</map>
<map title="MCD 2003" default="false" reference="false" zindex="-1">
  <file>http://geoint.lmic.state.mn.us/cgi-
bin/mapserv?map=/home/httpd/html/cap/gubs.map&REQUEST=GetMap&SERVICE=W
MS&VERSION=1.1.1&LAYERS=mcd2003&FORMAT=image/png&TRANSP
ARENT=TRUE&SRS=EPSG:26915&STYLES=&BBOX=%MINX%,%MINY%,%
MAXX%,%MAXY%&WIDTH=%WIDTH%&HEIGHT=%HEIGHT%</file>
  </map>
</group>

</mapbook>
```

Appendix D. WFS URLs used in this project

The following URLs were developed and used to test the GeoMoose client:

LMIC WFS URL:

Governmental Unit Boundaries (municipalities)

<http://geoint.lmic.state.mn.us/cgi-bin/mapserv?map=/home/httpd/html/cap/gubs.map&service=wfs&version=1.0&request=GetCapabilities>

DNR WFS URL:

Governmental Unit Boundaries (state parks)

http://deli.dnr.state.mn.us/cgi-bin/wfs?map=DELI_WFS_MAPFILE&service=wfs&version=1.0&request=GetCapabilities

Appendix E. Supporting FGDC Framework Data Layer Publishing Using MapServer

FGDC 2004 Category 5 CAP Grant 04HQAG0165
Final Project Report
December 8th, 2006

<http://www.fgdc.gov/grants/2004CAP/2004FinalReport/165-04-5-MN-FinalReport.pdf>

Appendix F.

ArcGIS Modeling for Updating Governmental Unit Boundaries

Jim Krumrie; LMIC

An ArcGIS Model was created to automate the updating of the governmental unit boundaries layer. This was necessary because the data changes are edited by multiple agencies and then consolidated by LMIC personnel. A process for merging several data sources was developed and captured by the ArcGIS model to facilitate the publication of these changes.

This step involves processing, verifying and combining the Minnesota Municipality and Civil Township geographic datasets into a joint Government Unit Boundary (MNGUB) geographic dataset. Specifically, current annexations were incorporated into the Municipality and Civil Township datasets before they were QA/QC'd and then combined into the MNGUB dataset. In addition, a data process model was developed and documented, entailing the aforementioned procedure.

MNGUB Annexation Update Procedure

The following steps were completed in ArcCatalog 9.2 and ArcMap 9.2 using various tools from ArcToolbox 9.2. The words in bold signify the names of the tools used while italic words signify parameters and settings. These steps were performed manually, however, it was intended that they eventually be incorporated into a model using ESRI's ModelBuilder.

1. **Select** records for new annexation features from Adjusted and Unadjusted boundary layers in ArcGIS SDE. Must select only those records with docket numbers found in the DOT_MISS table and in both the Adjusted and Unadjusted layers. However, the selected records in the Unadjusted layer that are also selected in the Adjusted layer must be deselected. In addition, annexations selected by this process with docket numbers that start with 'D' (deletion) must be separated from all other docket numbers (additions). These D-type features must be copied out to their own docket deletion layer that will be used later.
2. **Make Feature Layer** of each selected set of Adjusted and Unadjusted records. Feature layers are required for the Merge operation in the next step.
3. **Merge** Adjusted & Unadjusted feature layers from step 2 and original Municipal layer into a new combined Municipal layer. Set the field mappings as described below:
 - a. **Output field:** MUNI_NAME **Input fields:** Adjusted.MCD_NAME, Unadjusted.MCD_NAME, Original_Municipal.MUNI_NAME
 - b. **Output field:** FIPS **Input fields:** Adjusted.MCD, Unadjusted.MCD, Original_Municipal.FIPS (set formats so that only characters 5 – 10 are used for the input Adjusted and Unadjusted MCD fields).

4. **Select by Attribute** from appended Municipal file: *FIPS LIKE '□%'* . Some of the merged records from the Adjusted and Unadjusted feature layers may have junk (incorrect and unprintable characters) in their FIPS fields. (If none selected, skip to step 6.)
5. **Calculate** *FIPS = FID*. Replace the junk FIPS of those records selected in step 4 with their unique feature ID field value. Each selected record is thus given a unique FIPS to keep them from being incorrectly dissolved (joined) with other potentially unrelated records that would also have a blank FIPS field in step 6.
6. **Dissolve** on *FIPS* field with all statistic fields (except AREA & PERIMETER) set to *FIRST*. The original and new annexation municipal features with the same FIPS codes are thus joined together. Setting all statistic fields to FIRST ensures that the attributes of the pre-existing municipalities are retained and are not overwritten by their new annexations.
7. **Select** where *[MUN_] = 0*. This query selects all those new features (annexes) that have not already been joined with their respective municipality in step 6. (If none selected, skip to step 18)
8. **Sub-select** where *[MUNI_NAME] LIKE '% City'*. A subset of the unjoined annexes selected in step 7 is chosen whose names end with “City”. These names are most likely incorrect (e.g. Rochester City). (If none selected, skip to step 10)
9. **Calculate** *[MUNI_NAME] = Mid ([MUNI_NAME],1,Len([MUNI_NAME])-4)* . Removes “City” from the end of the names of the selected Muni records from step 8. There are few municipalities whose name actually does end with “City” in Minnesota. It is even more unlikely one of these will receive a new annexation without a FIPS code to join it with its respective municipality in step 6. (Even if this does happen and the “City” is incorrectly removed it will be caught by the error checking in step 17.)
10. **Select** where *[MUN_] = 0*. Select all those new features (annexes) that have not already been joined with their respective municipality in step 6. (If none selected, skip to step 18)
11. **Sub-select** *[MUNI_NAME] LIKE '%St.%'* . A subset of those features selected in step 10 is chosen whose name contains the abbreviation for “Saint”. For the sake of standardization, “Saint” should be spelled out without any abbreviation. (If none selected, skip to step 13)
12. **Calculate** *[MUNI_NAME] = Replace([MUNI_NAME], “St.”, “Saint”)*. Replaces the abbreviation for “Saint” with its spelled out version in those records selected by step 11.

13. **Select** where *[MUNI_NAME]* <> “*Saint Anthony*”. Selects all those features whose name is not “Saint Anthony” (this should be all but 2 records). At this point, Saint Anthony is the only duplicated municipality name in Minnesota.
14. **Dissolve** on *[MUNI_NAME]* with all statistic fields (skip AREA & PERIMETER) set to *FIRST*. Joins all those municipal features selected in step 13 based on their name. Only those named “Saint Anthony” will not be joined.
15. **Select** from pre-step 14 layer where *[MUNI_NAME]* = “*Saint Anthony*”. Selects municipal features named “Saint Anthony”.
16. **Append** “Saint Anthony” features into layer from step 14. Field mapping should be straightforward as both layers have the same fields.
17. **Select** where *[MUN_] = 0*. These are any remaining municipal annex features that have not yet been joined to their respective municipality. These need to be examined and fixed manually before proceeding with step 18.
18. **Erase** new Municipal layer with any deletion docket features found in step 1. These D-type features signify a deletion from the Municipality file and an addition to the Civil Township file.
19. **Merge** docket deletion features found in step 1 with the original Civil Townships layer. For field mapping, ignore all fields from the docket deletion layer except *NAME* where characters 1-4 should be mapped to *TWP_NAME*.
20. **Erase** Civil Townships layer with new Municipal layer to produce new Civil Townships layer. The Erase tool will remove any area of an existing township that is now occupied by a recently expanded municipal feature.
21. **Merge** new Municipal layer with new Civil Townships layer to produce new MNGUB layer. For field mapping, use *FIPS* fields from both layers to produce an output *FIPS* field and the *MUNI_NAME* field from the Municipal layer and *TWP_NAME* field from the Civil Townships layer to produce an output *NAME* field.

Problems Encountered

As previously mentioned, the procedure described above was intended to be incorporated into a model using ESRI’s ModelBuilder application. However, this was not entirely feasible because of deficiencies in the software and time constraints in the project schedule.

In ModelBuilder, the *Field Mapping* components of the *Merge and Append* tools do not retain the field settings chosen by the user but instead change back to default values when the model is validated or saved. This is a known bug (ESRI NIM031376) in ArcGIS 9.2

that was expected to have been fixed in version 9.3¹. Unfortunately, subsequent testing in ModelBuilder 9.3 shows the problem to persist. *Merge and Append* tools are integral to the creation of the necessary data (steps 3, 16, 19 and 21) and though workaround operations (see next section) were attempted, they were not completed due to time constraints of the project.

In step 17 of procedure, the model is intended to stop and inform the user if any remaining municipal annex features have not yet been joined to their respective municipality. Unfortunately, the stop and go capability of ModelBuilder is limited to a single variable that must be false in order to stop the model from continuing. What this means is that if any “unmatched” annex features are found then the result will be incorrectly interpreted by ModelBuilder as a true value that will not stop the model. There is no simple way in ModelBuilder to negate such a positive value to make it false and, again, there was insufficient time to complete any kind of workaround.

Finally, a large number of application lock-ups, failures and inconsistencies impeded progress in developing these procedures. All three of ArcGIS 9.2’s major applications – ArcCatalog, ArcMap, ArcToolbox – exhibited instability and were prone to frequent failures and lock-ups during the development phase of this project. Output data from given ArcToolbox operations often differed between their use in ArcMap and ArcCatalog even though they had used the same input data. These inconsistencies made it difficult and time-consuming to standardize the procedure and model and resulted in a less than fully automated solution for this prototype effort.

Potential Workarounds

To replace the errant *Merge and Append* operations, a Python 2.4 script was written to retain the field map values and prevent them from being changed incorrectly to default values. Unfortunately, this script was not implemented in time for the project completion.

A second Python 2.4 script was written to negate the number of unmatched annexes in the Municipality file and thus force the model to stop to let the user fix them (see step 17). Again, there was insufficient time to complete this workaround.

(See Figure on Page 25)

Figure 4. Process flow diagram illustrating procedures used to create an integrated Government Unit Boundary (MNGUB) geospatial dataset using ESRI’s ModelBuilder application.

¹ From a phone conversation on 7/14/2008 between the author and ESRI Support

MN_GUBS_ANNEX Model

