AutoCAD Map Conversion Based on Teigha.NET and ArcGIS Engine

Muyao Shi1,a, Xiaoyu Wang1,b, Wang Xin1,c, Gao Hui1,d, Mei Nan1,e, Huafang Li1,f, Shi Song1,g

1Beijing Municipal Institute of Labor Protection, Beijing 100054, PR China

ashimuyao@bmilp.com, bemail: xiaoyu_19@163.com, cemail: wx19891001@126.com, demail: gh@bmilp.com, eemail: meinan @bmilp.com, femail: lhf@bmilp.com, gemail: ss@bmilp.com

Keywords: AutoCAD, Conversion, Teigha.NET, ArcGIS

Abstract. In this study, we attempted to realize the conversion of CAD data such as commercial plans into ArcGIS. Based on actual requirements for the conversion of CAD DWG data to ArcGIS in engineering practices, a CAD data conversion tool was developed on the basis of the open source C# class library Teigha.NET. CAD data are pretreated and then sent to this tool after the established graph specifications are satisfied. Graphic conversion is performed and the graphics are sent to the map issuance system after topologic check and geometric error repair by ArcGIS. This tool has been applied to convert data of commercial plans, and the conversion results indicated that the proposed conversion method exhibits high graphic conversion efficiency, satisfies the requirement for map display, saves working hours, and improves work efficiency.

Introduction

With the development and popularization of the GIS and Internet technologies, demands for map visualization, such as the display of commercial plan data with the ultimate aim of displaying thematic maps through the Internet, have increased. However, a problem during implementation is that original plan drawings are mostly created using AutoCAD in the DWG format. DWG files are binary graphic files with a closed format because of which common users cannot read their contents directly. Thus, these DWG drawings must be first converted into GIS data for map issuance of commercial plan data. Generally, information from DWG data is acquired through data conversion interfaces provided by various platforms such as ArcGIS, AutoCAD, and FME.

This study formulated detailed CAD drawing specifications and presented corresponding conversion methods for different types of graphic elements, mainly for commercial planning data. It also solved the reading of coordinate and attribute information from DWG data using Teigha.NET and wrote these information in the Shapefile format with related interfaces in the ArcGIS Engine. The converted GIS data can be imported into the WebGIS system for map display.

Issues regarding Data

Differences between CAD and GIS. It is difficult to convert DWG data into GIS data. Firstly, the data model of CAD is very different from that of GIS. CAD data focus on the design of drawings and often neglect the attribute data of graphics. Furthermore, CAD graphics contain only drawing information such as line width, color and line type. In contrast, GIS data consider spatial relations, such as adjacent objects. In short, CAD data contain only graphics information while GIS data contain spatial databases. Secondly, graphics in CAD layers are very complex, and one CAD layer can be drawn with block references, open polylines (PLine), closed PLines, and text, while those in GIS layers are very simple, and one layer represents only one type of element, such as...
point, line, and plane. Finally, CAD graphics are much more precise than GIS data on large-scale maps, but they often adopt coordinate systems different from that of GIS. Plane coordinate systems are often adopted in CAD while projection coordinate systems, such as UTM, are frequently adopted in GIS. In addition, metadata information, such as the author’s intention, source, range, precision, and other important information about graphics, are seldom saved in CAD during drawing, whereas detailed metadata information of maps are preserved in GIS. Therefore, the first step of converting CAD data into GIS data is to establish CAD drawing specifications to guide the pretreatment of data or the drawing of new drawings.

**CAD Drawing Specifications.** For the conversion of commercial plan data from DWG, CAD drawing specifications must be established to make requests for the pretreatment of CAD data and the drawing of new drawings. The specifications should take into account the demands for data display from users, namely, the specific type of data to be displayed. For instance, in engineering, the elements of graphical display include the shop, overall framework of buildings (wall), red line range, partition wall of shop (counter map), column, window (transparent curtain wall area), staircase, escalator, elevator, equipment room (including air-conditioner room, monitoring room and water pump house), bathroom, functional areas inside and outside the red line range (including loading area, square and parking lot), internal generatrix, empty area, entrance (primary and secondary entrances of the gate, garage entrance), and other functional areas (VIP room, customer service center, personal office, warehouse and staff canteen), in which the shop is the main part and the other elements are supplementary, which must be consistent with CAD drawings. Thus, the specifications were formulated as follows:

1) Layer division should be elaborate and not unrefined. Each layer should be of a single type of element and different thematic elements should be placed in different layers. The layers should be named normatively, i.e., each layer should contain only points, lines, or planes, and only one type of thematic element.

2) Planar ground features should be drawn by closed PLines, and existing open planes should be edited and closed.

3) To facilitate the conversion of elevators or stairs, their boundaries should be drawn into closed polygons. Steps should be scattered in straight lines and placed in different layers from boundaries, such as an elevator boundary layer or an elevator step layer. The two layers represent the elevator. It is suggested that the arrows on elevators or stairs should also be drawn in a new layer to better present elevators in GIS in the future.

4) According to the demands of users, it should be determined whether to scatter other block reference objects (corresponding to the point elements in GIS), except for elevators.

5) Text notes should also be layered.

6) Drawing files should be organized according to floors, i.e., each drawing should be drawn with the ground objects of only one floor.

In particular, the layer division should be clear and distinct with the names in your native language; the graphics in each layer of points (also called block reference in CAD), lines, or planar ground objects for the elements in GIS also fall primarily under these three types. Planar ground objects should be drawn using the closed PLine command. All the planar ground objects meeting the requirements of conversion are drawn by closed PLines, except for those with sides of straight lines. In such cases, the plan for modifying the original drawing is as follows: The original data of the user (such as A-original building walls, new walls, and finished surfaces) can be reserved and a new CAD layer will suffice. For example, the layer can be named as “shop”, and
then other layers are drawn out by closed PLines. The conversion tool can convert all the original walls into GIS layers and also the new CAD data meeting the requirements into GIS data.

In general, the original data of the user can be modified slightly and appropriately, such as closing all the data that can be closed (usually shops and counters). Other ground objects, such as drawing frame, column, door, window, T-molding line, and elevational line, should be kept consistent with CAD drawings by the user during display in GIS under most circumstances. In this manner, graphics originally drawn by line elements should be reserved, and if it is difficult to modify them, a new layer can be created and these lines can be drawn again. Regarding block references, they should be scattered and converted as much as possible into basic graphic elements, such as straight lines, circles, arcs and elliptic arcs, so as to display them more realistically in GIS.

**Method Design**

*Technologies for Reading DWG Data and Writing GIS Data.* DWG data is read by Teigha.NET, which was previously known as OpenDWG and DWGDirect. Teigha.NET is an object-oriented class library designed and developed by Open Design Alliance. This class library supports multiple versions of DWG and can read DWG files directly to acquire data. Programs developed based on this class library can operate independently with high efficiency and security without the AutoCAD platform\(^1,2\). The Teigha.NET class library can be used in the .NET platform by Microsoft with high development efficiency and powerful functions. The version adopted in this study is Teigha.NET 4.00, and the core class library is in the file TD_Mgd_4.00_10.dll.

GIS data is written by ArcGIS Engine, and the conversion results of CAD data are stored in the Shapefile format of ESRI ArcGIS, which is a format of the vector data structure of non-topological entity describing the geometrical and attribute characteristics of spatial data. The Shapefile format usually comprises the following files: the main file (*.shp) for storing geometrical data, the index file (*.shx) for storing the offset of the corresponding main file record contained in each record to the main file header, dBASE table file (*.dbf) for storing attribute data, and spatial reference file (*.prj) for storing the spatial reference information of data\(^3\).

*Data Conversion*

**Process Design.** Based on the characteristics of CAD data, Teigha.NET and ArcGIS Engine were utilized in this study to convert commercial plan data in the CAD format. A flow chart of the conversion process is shown in Fig. 1.

![Figure 1. Process of CAD data conversion.](image-url)

Firstly, geometrical data and attribute data of CAD drawings are read by Teigha.NET, and then corresponding shapefiles are created by ArcGIS Engine to store the geometrical and attribute data of graphics. GIS elements can be classified into point, line, and plane; it should be noted that text notes are stored by points. The geometrical information comes from the inserted base points of CAD texts and the attribute information comes from the text content. GIS data is displayed in the form of labels during graphical display. After treatment should be performed on the obtained GIS
data by geometrical check and repair tools. Subsequently, the data is input to GIS spatial database and displayed graphically in the GIS software[4].

Data Conversion. Various information in DWG files are saved in a database in the memory in the form of a series of objects, and each example of the database class stands for a simple database, which is composed of 9 symbol tables and an object dictionary, namely BlockTable, LayerTable, LinetypeTable, TextStyleTable, DimStyleTable, RegAppTable, UCSTable, ViewportTable, and ObjectDictionary. Each symbol table and object dictionary contains multiple records. For example, BlockTable contains multiple BlockTableRecords. Graphic entity data, such as points, straight lines, PLines, splines, arcs, and segments, are contained in BlockTableRecord. LayerTable contains multiple LayerTableRecords (as shown in Fig. 2), and LayerTableRecord stands for the layer of CAD[5].

![Figure 2. Database structure of Teigha.NET for .DWG.](image)

According to the structure of the database, information in DWG, such as entity data and layers, can be accessed by corresponding classes and methods. The entity data information in DWG is read by traversing the DWG database and information (e.g., the coordinates of a corresponding entity is read according to different entity types), and then corresponding GIS files (Shapefiles) are created one by one according to the types including point, line, and plane for writing the read entity information. Thus, it is crucial to establish the correspondence between the CAD primitive and the ArcGIS primitive as shown in Table 1. The conversion is performed according to CAD entity type after establishing the correspondence of the type with the process of reading the CAD entity data, thereby establishing Shapefile and writing geometrical and attribute data of the entity. The conversion of each type of entity should refer to the attributes and methods of the entity types of Teigha.NET and ArcGIS Engine. Despite the differences between the two class libraries, the attributes and methods of graphic entity objects can correspond with each other to realize the conversion of DWG data (as shown in Fig. 3).
Table 1. Correspondences of geometrical types.

<table>
<thead>
<tr>
<th>ID</th>
<th>Teigha.NET class</th>
<th>ArcGIS Engine class</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BlockReference</td>
<td>PointClass</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Line</td>
<td>PolylineClass</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Arc</td>
<td>CircularArcClass</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Polyline</td>
<td>PolylineClass</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Polyline</td>
<td>PolygonClass</td>
<td>Open</td>
</tr>
<tr>
<td>6</td>
<td>Circle</td>
<td>CircularArcClass and PolygonClass</td>
<td>Closed</td>
</tr>
<tr>
<td>7</td>
<td>MText or DBText</td>
<td>PointClass</td>
<td>Storing base points and text contents</td>
</tr>
<tr>
<td>8</td>
<td>Ellipse</td>
<td>EllipticArcClass and PolylineClass</td>
<td>Only for elliptical lines</td>
</tr>
</tbody>
</table>

```csharp
public void Convert(string dsFolderPath, string strSaveFolder, string[] szLayers) {
    var aEat = File.ReadAllLines(dsFolderPath); // Load all data into a string array
    // Process each line
    foreach (var strLine in aEat) {
        // Process each line
        foreach (var strClass in szLayers) {
            // Check if the line contains the class name
            if (strLine.Contains(strClass)) {
                // Perform specific conversion logic
                // Example: Convert the line to a specific class
            }
        }
    }
}
```

Figure 3. Program for the detailed conversion of CAD data into GIS data.

Check and Repair of Graphics. The newly converted Shapefile may contain some geometrical errors; for example, the planar ground objects may not be closed. Therefore, GIS layers should be checked and repaired through several tools, namely ArcToolBox ➔ Data Management Tools ➔ Features ➔ Check Geometry ➔ Repair Geometry, and then they can be imported to the spatial database after all the errors are repaired.
Implementation of Tools

The conversion tool CAD2GIS was implemented in the development platform Visual Studio 2010 by Microsoft and C# was adopted as the programming language. DWG data can be read and converted by programming after introducing the powerful class libraries Teigha.NET for .DWG and ArcGIS Engine. The esriSRProjCSType.esriSRProjCS_World_Mercator projection coordinate system was adopted for GIS data to facilitate the graphical display of data by ArcGIS Server. CAD2GIS is completely independent from the AutoCAD platform, easy to operate, and powerful in function. Drawing specifications and this conversion tool can customize various utility functions according to the requirements of actual projects to improve their production efficiency.

This study conducted a CAD data conversion experiment using data of a commercial plan. Fig. 4 and Fig. 5 show the CAD data and converted GIS data, respectively. The converted results indicated that the conversion method and tools designed in this paper can realize data conversion while retaining complete information so as to meet the requirements of users.
Conclusions

Based on practical project requirements, this study first formulated CAD drawing specifications and then developed the graphic conversion tool CAD2GIS with the powerful DWG read-and-write library Teigha.NET, ArcGIS Engine, which features outstanding processing capacity of geographic data, and C# language to realize the conversion as well as check and store CAD data to improve the production efficiency of projects. Commercial plan data were used as an example in the conversion. It is significant because this tool can serve as the basis of graphical display of CAD data using GIS technology. The drawing specifications presented in this conversion tool can also be referred to in other types of CAD data, such as for the conversion and storage of CAD pipeline data.

References