CROSS-PLATFORM GIS DEVELOPMENT

B. Gültekin ÇETİNER [1] *and Bahadır ZEREN* [2]

Istanbul Technical University, Faculty of Aeronautics and Astronautics, Istanbul, Turkey

[1] cetinerg@itu.edu.tr [2] zerenb@itu.edu.tr

Geographical Information Systems (GIS) allow employment of database technologies together with spatial based information. Therefore, they present a suitable platform to spatially control and display information for many organizations such as Local Governments and software houses developing Vehicle Tracking or similar systems. For Client/Server type of Geographical Information Systems, desire of multiple operating systems necessitates different approaches to GIS development. In this paper, a software component namely TGISMap is introduced to use GIS functions both under Windows and Linux environments. The importance of the approach is that the same source code for the GIS applications can be transferred between two aforementioned operating systems with little or no change. The descriptions of the cross-platform GIS development are provided from the point of database design and necessary tools.

1. INTRODUCTION

When geographic information systems (GIS) were introduced in the 1950s, its early use was limited to a small group of researchers. Botanists, meteorologists, and transportation planners began automating the process of thematic mapping. These researchers' efforts represent the early attempts at computerized cartography. As being one of the fastest growing technologies, GIS has emerged as a powerful and sophisticated means to manage vast amounts of geographic data.

The use of geographic information tools has received overwhelming acceptance by local, state, and federal government organizations as well as the private sector. This technology is enabling organizations to consider more effective ways of doing business. Towns, cities, and counties throughout the world are now embracing GIS. Many local governments have embraced GIS with high expectations. However, after absorbing the cost of building large databases, only a few organizations have made GIS available to all staff. The vast majority of local government officials are unable to take advantage of GIS for their daily duties. Many examples exist of the technology being too costly, accompanied by a lack of user-friendly software. On the other hand, the existence of different computer operating systems makes cross-platform solutions difficult for GIS development. GIS have therefore been arguably restricted to a few specialists within an organization.

GIS has grown out of a number of technologies including cartography, information management, computer science, photogrammetry, and remote sensing. Advancements made in these fields correspond to advancements in GIS. This technology, therefore, consists of computer software and hardware designed to organize spatial data for analysis, assessment, and cartographic depiction. It provides a mechanism by which information on a feature's location, spatial interaction, and geographic relationship can be assessed and viewed in moments. It provides an opportunity to efficiently view and access geographic data to improve the decision-making process.

GIS relies on the integration of three areas of computer technology: a relational database management system to store graphic and non-graphic data; cartographic capabilities to depict, graph, and plot geographic information; and spatial analytical capabilities to facilitate manipulation and spatial analysis [*E. J. Han*, 2002]. Three distinct areas are shortly:

a) Graphic capabilities, **b)** Relational database and **c)** Spatial analysis

Most researchers organize data in such a way that they can be envisioned as digital layers or coverages of information. Maps contain the layers, and layers contain the geographical datas. Geographical data can be expressed as geometric shapes such as points, lines and polygons. In most of the development tools, layers usually contain one kind of theme such as soil (polygon), river (line), road (line), well (point) etc. Based on the three requirements above, a software component is introduced herein to develop GIS applications both under Windows and Linux operating systems. Architecture of the component is designed in such a way that the same source code for GIS applications can be employed on both platforms with little or no change. The paper outlines main features of the cross-platform GIS component.

2. CROSS-PLATFORM GIS COMPONENT (*TGISMap*)

While Cross-platform means the ability of using piece of software under different platforms (usually different operating systems), a computer programming language (and usually a visual development environment) that is capable of working with multiple operating systems is required. This language has to provide extensive capabilities for cross-platform GIS development. One way of achieving this might be to use a platform independent language such as Java or C. However, demanding requirements for GIS development suggest different alternatives since Java is slow as being a run-time interpreter and C is lacking extensive graphical and relational database libraries needed for cross-platform GIS functions. An ideal alternative is a development environment specifically designed for cross-platform application development with extensive graphical, relational database design capabilities. This alternative should also have a common object-oriented programming language for spatial analysis purposes [Raper, J., and Livingstone, D, 1995; Panchal V. K.,1996]. Delphi† and Kylix† are chosen for this purpose. Both Delphi and Kylix are object-oriented, visual programming environments for rapid development of highly efficient cross-platform applications with a minimum of manual coding. They provide a common comprehensive class library called the Borland Component Library for Cross Platform (CLX) and a suite of Rapid Application Development (RAD) design tools with extensive graphic and database capabilities. The Cross-Platform GIS component described herein makes use of this CLX library components and objects in a truly object-oriented fashion. Figure 1 shows some of the main classes available in CLX library.

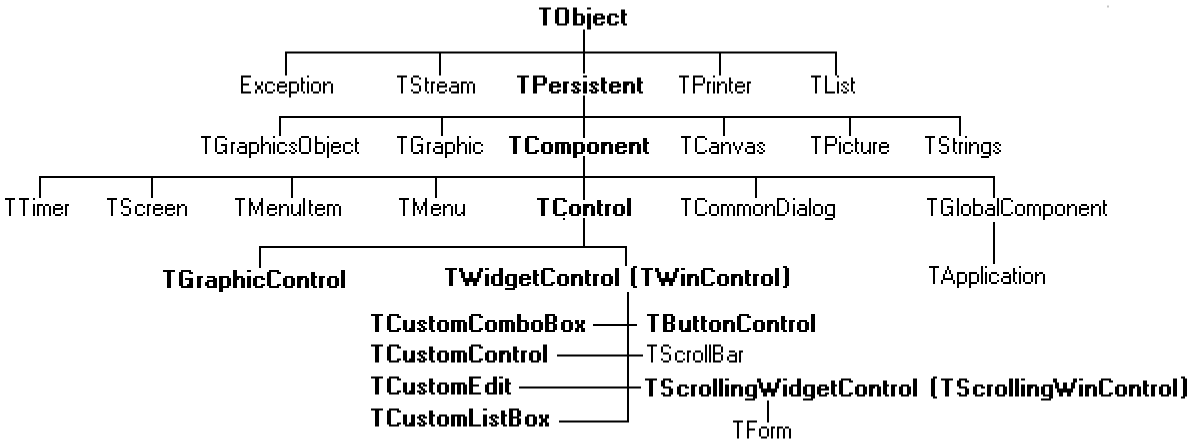

Figure 1: General schema of CLX Library

Some of the important classes used in development of cross-platform GIS component are as follows:

a) TGraphicControl: It is a base class that GIS component will be set on. Drawing and showing 3D data are all handled in this base class.

b) TCanvas: It is the base class providing drawing. Transfer of the spatial data on window and drawing operations by the user in an interactive fashion are maintained in this class.

c) TGraphic: Assembled Vision Transfer or store of raster data in memory are all handled by using this class and sub-classes (TBitmap, for example).

d) TThread: With this class which is inherited from TObject, multi-threading is achieved. This class makes you draw the things simultaneously while you are reading and writing data in memory or file.

e) TStream: This class and sub-classes inherited from it (for instance, TFileStream, TBlobStream) are responsible for data transfer between the GIS component and metafiles containing maps.

For relational database functions of GIS component, DataCLX library (Cross-Platform Library for Database) is used. Without concentrating on different relational database systems such as InterBase, Oracle, DB2, MySQL and unspecified database systems using ODBC connection, developer can have the functionalities of database security, transactions, referential integrity, stored procedures and triggers provided by DataCLX. DBExpress database components in DataCLX library are used for all database functionalities for different database systems. Main database components in DataCLX library used in GIS component are as follows:

a) TSQLConnection: It is the class that provides the connection to the Database-Server. It can execute SQL queries directly.

b) TSQLTable: This class provides connection to the database as a database table. It connects to database using its SQLConnection class. This component returns a unidirectional dataset to the user from a table desired.

c) TSQLQuery: This class provides connection to the database as SQL queries. It connects to database using its SQLConnection class.. This component returns a unidirectional dataset to the user as result of a SQL query specified in its SQL property.

d) TSQLStoredProc: It is the component to execute stored procedures in database. It connects to database using its SQLConnection class. This component returns a unidirectional dataset.

e) TSQLClientDataset: This class provides connection to the database either as database table or SQL query. It connects to database using its SQLConnection class. It returns a bidirectional dataset through which data is not only read but also can be updated. In contrast to the unidirectional datasets, all data as a result of the query is transferred to the memory of the client machine and can be manipulated locally.

f) TClientDataset: It is the file-based dataset component that works on XML and XML-like data.

For GIS system, when components explained above are considered, typically two types of database structure can be established:

- **Client/Server GIS System:**

In Client/Server model (Figure 2) of *TGISMap*, after a geographic database created, the connection to the server computer is made by the *TSQLConnection* and after that the user can query data by using *TSQLQuery* component. The *TGISMap* component working on client-side loads the map-data, which are obtained as a result of the query, to the memory of the computer. Geographical data then can be manipulated locally and sent back to the server.

- **Local GIS System:**

In local model, geographical data is stored and manipulated locally on a flat-file basis choosing from XML, dBase, Paradox systems. *TClientDataset* component can be used to query data from the file on the local machine. Data manipulation is maintained same as in Client/Server model. In case of local GIS model, multiple users are not allowed.

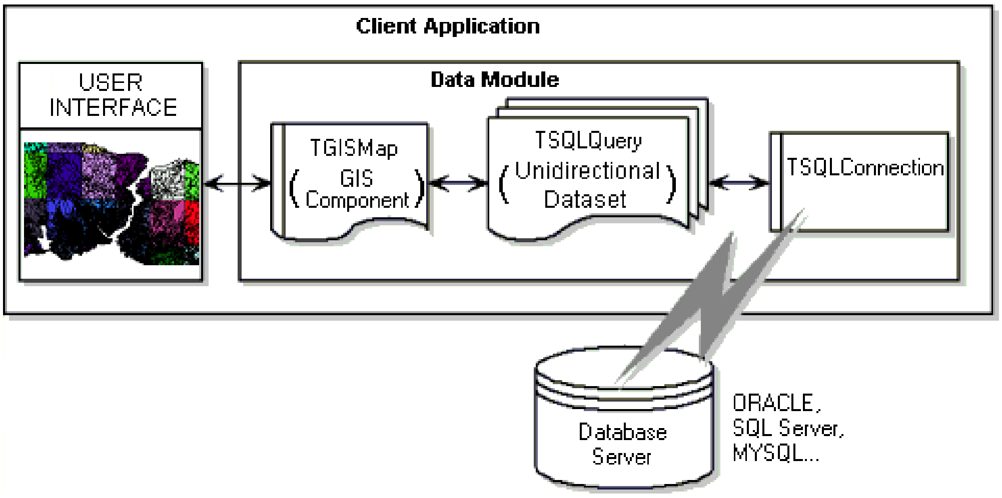

Figure 2: Client/Server GIS model

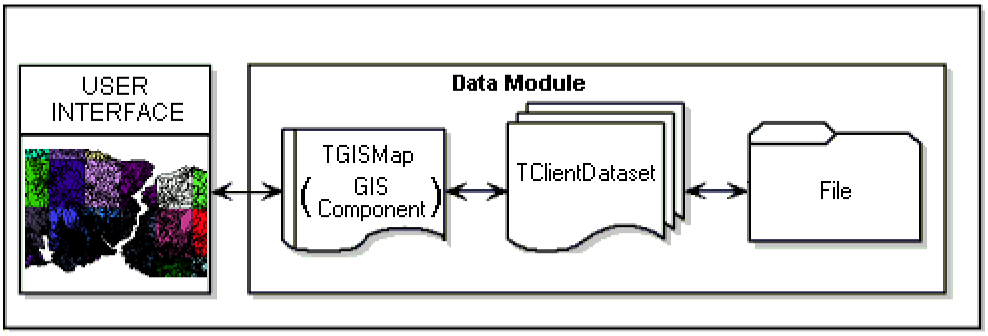

Figure 3: Local GIS model

3. ARCHITECTURE OF GIS COMPONENT (*TGISMap*)

The main parts of the GIS system are the layers including various themes and the shapes that representing the geographic data in these layers. General structure consists of layers each typically containing one type of shape (points, polygons or lines) as seen in Figure 4.

The layer structure in *TGISMap* component is outlined below:

```
TLayer = record
    LayerCode     : Integer;
    Name          : String;
    LayerObjects : TPObject;
    .
    .          Other Attributes(Layer Boundaries, Attributes
    .          relevant to the drawing, etc…)
end;
```

Here *LayerObjects* shows the address of the first object on the layer. All other objects can be tracked following the addresses in the linked list. The component allows many layers which are stored in a dynamic array specified in object Pascal as (`Layers : array of TLayer;`). Adding and removing layers and objects to the component is very flexible.

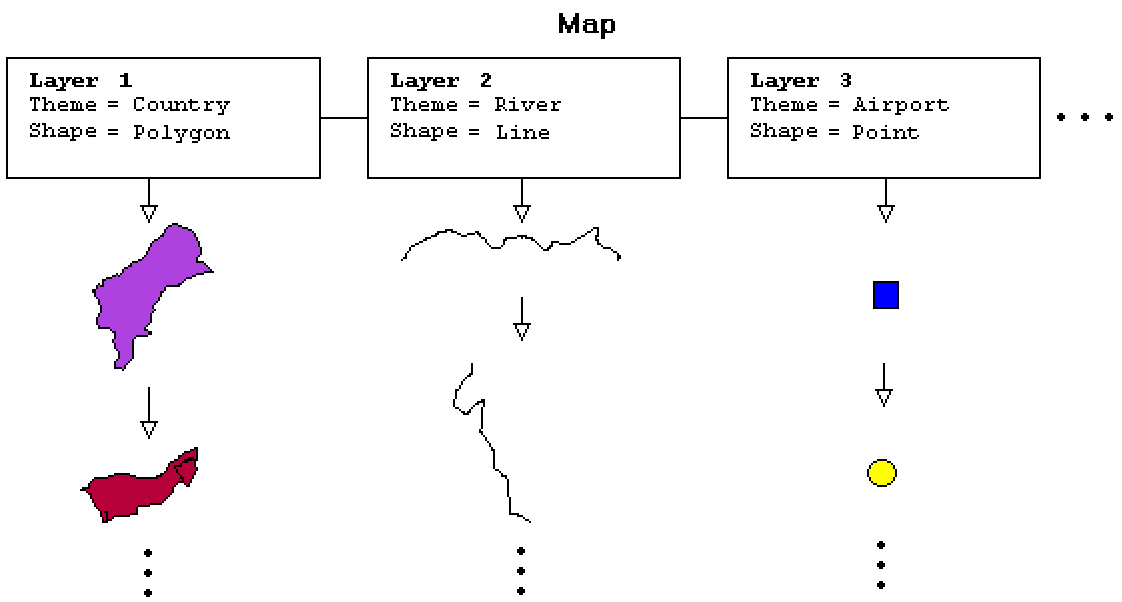

Figure 4: Schema of basic GIS System

The structure is as follows for the objects on the layers;

```
TPObjects = ^TObject;

TObject = record
    ObjectCode   : Integer;
    Type         : TObjectType;
    LayerCode    : Integer;
    Coordinates  : Pointer;
    NextObject   : TPObject;
    .
    .
    .          Other Attributes (Number of points, the number
    .          items etc…
    .
end;
```

Linked lists of the layer objects are maintained in a dynamic and flexible fashion (Figure 5). Coordinate information is stored in memory buffers using the addresses pointed by the *CoordinateInformation* attribute.

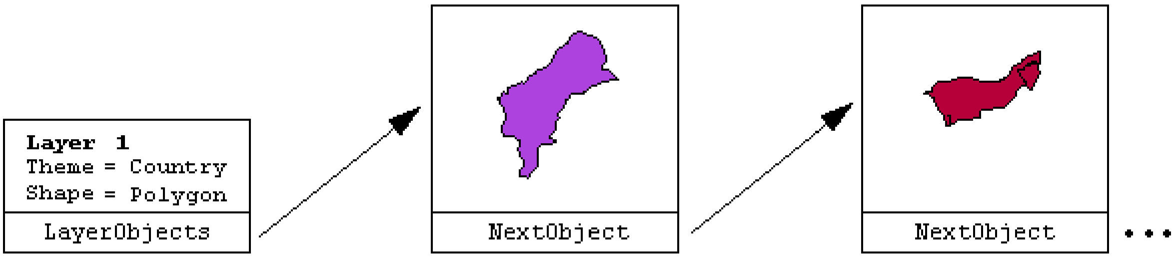

Figure 5: Linked list schema on GIS System

Geographic data of *TGISMap* is saved in Cartesian coordinate system with double precision values. Every operation on the screen (zooming, picking of object etc…) is converted to the Cartesian coordinate system by a transfer function (Figure 6).

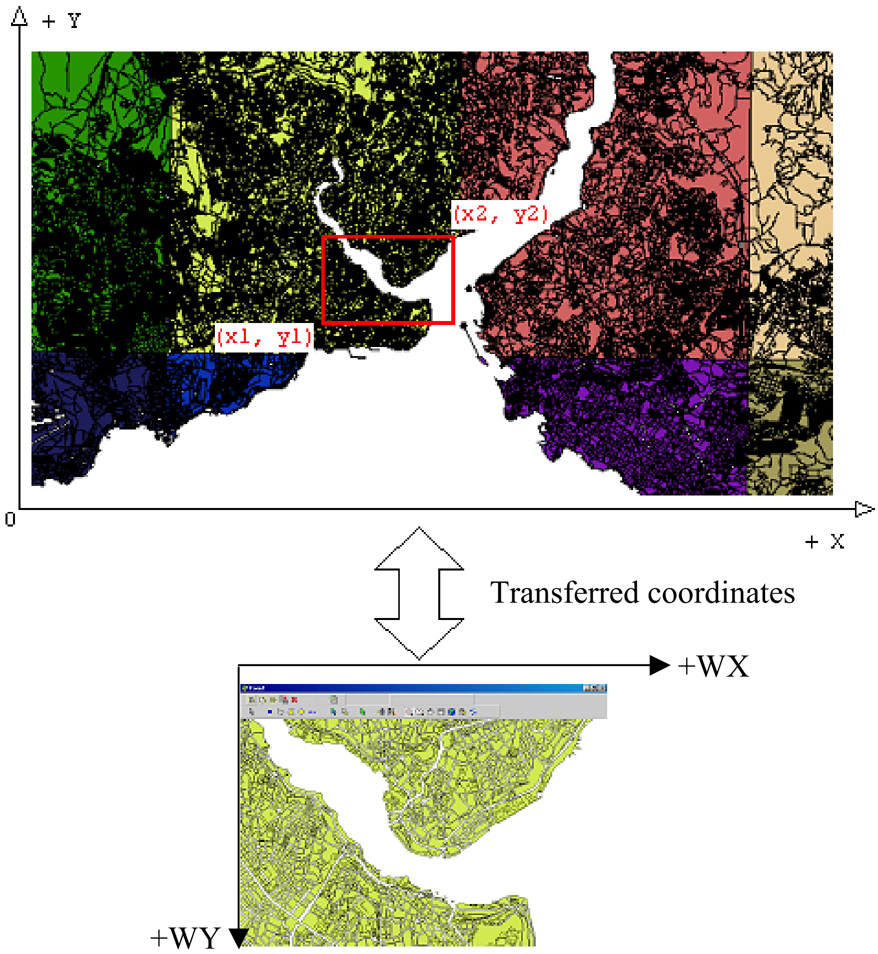

Figure 6: Transferring geographical data to user interface

4. DATABASE STRUCTURE FOR GIS COMPONENT

There are many different data models suggested for GIS systems using relational or object-oriented notations [*Worboys, M. F. et al,1990*]. A sample of the data model constructed for *TGISMap* component is illustrated in Figure 7. The sample data model is created using the data modeling application called ERWin‡ and IDEF1X data modeling notations.

Layers, objects inside these layers, maps and drawing settings for geographical data are saved in any database system supported by DataCLX library. The main entities in data model are *LAYER*, *MAP* and *OBJECT_TABLEs*. The many-to-many relationship between *LAYER* and *MAP* is resolved using the intersection entity called *MAPXLAYER*. This structure allows a layer (for example country) to be included in different maps and different attributes (drawing, color etc.) to be set for each map and layer combination.

OBJECT_TABLEs are created for each layer to store the objects in it. These tables may also be used to add new extra attributes (columns) to the tables where the layer objects are on. Objects' coordinate informations are stored in blob fields named *CordinateInformation* in these tables. Those fields contain a specific structure similar to shape files defined by ESRI [*ESRI, 1998*]. Each object table contains only one type of object information selected from points, polygons or lines.

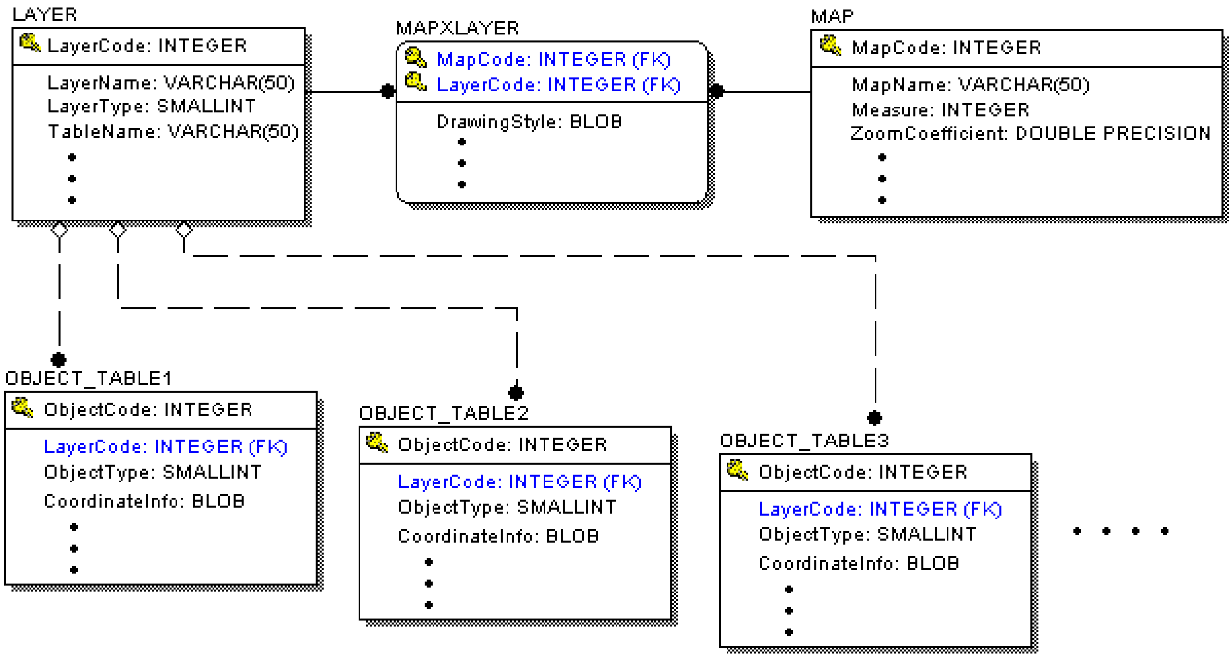

Figure 7: Sample data model showing the structure of GIS component

5. CONCLUSIONS

In this paper, a software component namely *TGISMap* is introduced to use GIS functions both under Windows and Linux environments. The component is developed in an object-oriented language using powerful cross-platform application development libraries suitable for GIS. The importance of the approach is that the same source code for the GIS applications can be transferred between two aforementioned operating systems with little or no change. The descriptions of the cross-platform GIS development are provided from the point of database design and necessary tools. Extending GIS component to Internet is very demanding for many purposes [*Teerayut Horanont, 2002*]. The *TGISMap* component currently does not provide Internet support and study continues to extend the capabilities of the component for cross-platform Internet development.

REFERENCES

E. J. Han, "Architecture Construction for For/With Component in GIS", The 2nd Annual International Conference on Computer and Information Science (ICIS '02), (2002)

Environmental Systems Research Institute Inc. (ESRI) (www.esri.com), "Shape file technical description", (1998)

Panchal V. K., Shina, L. K. Prusty, B.G. Chansarkar, R. A. Object-Oriented Design to Terrain Analysis. Indo-US Symposium-workshop on Remote Sensing & its Applications, Mumbai, India, (1996)

Raper, J., and Livingstone, D. Development of geomorphological spatial model using object-oriented design. International Journal of Geographical Information Systems 9., (1995)

Teerayut Horanont, Dr.Nitin Kumar Tripathi, Dr.Venkatesh Raghavan, "A Comparative Assessment of Internet GIS Server Systems", (2002)

Worboys, M. F., Hearnshaw, H. M. and Maguire, D. J., Object-oriented data modeling for spatial databases. International Journal of Geographical Information Systems 4., (1990)

TRADEMARKS

† *Delphi® and Kylix® are Trademarks of Borland Corporation.*

‡ *ERWin® is Trademark of Computer Associate (CA) Corporation.*