BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi" din Iaşi
Tomul LV (LIX), Fasc. 2, 2009
Secția
HIDROTEHNICĂ

EXTENDING THE ARCGIS PLATFORM THROUGH OBJECT ORIENTED PROGRAMMING

BY

DOMNIȚA MATEI

Abstract. This work presents the possibilities of extending the ArcMap application and the other applications in the ArcGIS platform through Object Oriented Programming (OOP). Using OOP a developer can create new features or functions based on the existing ones. The main objective of the work is creating a command for ArcMap that adds a new function to ArcMap version 9.2. The new function offers the possibility of exporting a set of vector or raster layers in KML format for data viewing in open source and free software. ArcGIS version 9.2 and lower does not offer the possibility of exporting geographic data in KML format (or in other open formats), so users that need to view the data must use software provided by ESRI or oe of the open source GIS alternatives. But exporting data in the KML format allows the user that only needs to view the data to be able to do this in free applications like Google Earth or directly on the internet in Google Maps.

Key words: GIS, programming, applications, software.

1. Introduction

A Geographic Information System (GIS) integrates hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information, analyzing spatial relationships, and modelling spatial processes. A GIS allows users to view, understand, question, interpret, and visualize data in many ways in the form of maps, globes, reports, and charts [1].

A GIS helps users answer questions and solve problems by looking at data in a way that is quickly understood and easily shared (the digital map). The GIS can offer the information needed to solve certain problems, allows viewing the relations and connections between data and modelling the geographic space and events that happen within it. For example, a GIS can be used by a transportation company to estimate transport costs and timings between two locations, can choose and create a map of the best route. Also, a GIS can perform other types of operations like estimating avalanche risk on a mountain slope, creating an evolution model for a forest fire and any other problems that can be related to geographic location of elements.

ArcGIS is an integrated collection of software products, providing a standards-based platform for spatial analysis, data management, and mapping. ArcGIS can be used via the Web, mobile devices, and desktop applications and can also be integrated with other enterprise systems such as work order management, business intelligence, and executive dashboards [3].

The ArcGIS desktop applications can be accessed using three software products, each providing a higher level of functionality. All the three software products share the main applications (ArcMap and ArcCatalog), just the functionality is different:

a) ArcView® provides comprehensive mapping and analysis tools, along with simple editing and geoprocessing tools.

b) ArcEditor includes the full functionality of ArcView, with the addition of advanced editing capabilities.

c) ArcInfo extends the functionality of both to include advanced geoprocessing [3].

The main application for working with maps in the ArcGIS Desktop products is ArcMap. ArcMap lets users create and interact with maps. In ArcMap, users can view, edit, and analyze the geographic data. Users can query the spatial data to find and understand relationships among geographic features. Users can symbolize the data in a wide variety of ways [2].

Different users use the applications from the ArcGIS platform in different ways, so the platform offers different levels of customization. From creating shortcuts or toolbar buttons for widely used commands to creating new complex tools that offer functions different from the existing ones, the possibilities of customizing the platform are presented in chapter **2**.

2. Customizing the ArcInfo Platform

The first and simplest level of ArcInfo customization involves no programming knowledge. All users can easily change the look and feel of the ArcInfo applications using standard user interface capabilities. For example, toolbars can be turned on and off using the customize dialog or buttons can be added to a toolbar and many application properties can be changed using simple actions like this.

The second level involves using the in-built Visual Basic® for Applications (VBA) application scripting capabilities to add new menus, tools, and work flows to the ArcInfo applications. The ArcMap and ArcCatalog applications include a VBA development environment. With VBA, it is possible to create applications using existing functions and data that run within the ArcMap and ArcCatalog application frameworks. It is not, however, possible to create new custom feature classes or build applications that run outside ArcMap and ArcCatalog. VBA is a very good choice for small-to-medium-size applications that use or extend the existing ArcInfo applications or functions.

Serious software developers who want to create reusable software building blocks, new applications, or custom feature additions to the geodatabase object model will prefer to work with ArcObjects directly using Visual Basic, or, especially in the latter case, Microsoft® Visual C++® or Delphi® [5]. This level of customization involves knowledge of Object Oriented Programming and ArcObjects, which will be presented later.

The application presented here will use the third method of customizing the ArcGIS Platform to create an extension to the ArcMap application. To understand the way this extension is created, the concepts of Object Oriented Programming and details about ArcObjects are presented first.

3. Object Oriented Programming

Object-oriented programming (OOP) can trace its roots to the 1960s. Researchers studied ways to maintain software quality and developed OOP to address common problems and improve reusability of program parts. The methodology focuses on data rather than processes, with programs composed of self-sufficient modules (objects) each containing all the information needed to manipulate its own data structure.

OOP may be seen as a collection of cooperating *objects*, as opposed to the conventional model, in which a program is seen as a list of tasks (subroutines) to perform. Each object is capable of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an entity with a distinct role. The actions (or "operators") on the objects are closely associated with the object.

We will use a simple example to present the basic concepts of OOP. As OOP uses objects to create a programming model, we will consider a car as the main object on which the examples are based.

The basic concepts are the following:

Class. The root word of classification is *class*. Forming classes is an act of classification, and it is something that all human beings (not just programmers) do. For example, all cars share common behavior (they can be steered, stopped, and so on). Different behaviors appear as different *methods* in a program. A method is a series of instructions that describe a behavior of an object. Also, cars share common *attributes* (they have four wheels, an engine,

and so on). You use the word *car* to refer to all of these common behaviors and properties [4].

Object. The word *car* means different things in different contexts. Sometimes we use the word car to refer to the general concept of a car: we speak of car as a *class*, meaning the set of all cars, and do not have a specific car in mind. At other times we use the word car to mean a specific car. The term *object* or *instance* is used to refer to a specific car.

The three characteristics of identity, behavior, and state form a useful way to think about and understand objects.

Identity is the characteristic that distinguishes one object from all other objects of the same class. For example, two cars created in the same factory in the same year with the same parts appear identical. The thing that makes them different is the serial number on the chassis.

Behavior is the characteristic that makes objects useful. Objects exist in order to provide behavior. Most of the time you ignore the workings of the car and think about its high-level behavior. Cars are useful because you can drive them. The workings exist but are mostly inaccessible. It is the behavior of an object that needs to be accessible.

State refers to the inner workings of an object that enable it to provide its defining behavior. A well-designed object keeps its state inaccessible. For example, the driver should not have access to the inner workings of the engine.

Inheritance. Inheritance is a relationship that is specified at class level. A new class can be derived from an existing one. For example, a car looks and acts the same whether it's an AllRoad vehicle, a town car or a limousine. Each of these have four wheels, a steering wheel, brakes and other known characteristics. But each of these types of cars have new features added to the standard features of a car. Instead of rewriting every method or attribute that already exists at a car, the programmer can inherit all the characteristics of the car and only add new properties or behavior.

Interfaces. In a typical class hierarchy, the operation (the name of a method) is declared in the base class, and the method is implemented in different ways in the different derived classes. The base class exists solely to introduce the name of the method into the hierarchy. In particular, the base class operation does not require an implementation. An interface contains no implementation of any kind; An interface contains only operations (the names of methods). Interfaces are important constructs in object-oriented programs. When you derive from an interface, it is said that you *implement* that interface. When you derive from a non-interface (an abstract class or a concrete class) it is said that you *extend* that class.

4. The ArcObjects Library

***ArcObjects*™** is a collection of components that form the foundation of ArcInfo™ software (Fig. 1). Developers can use the ArcObjects framework to

enhance and extend ArcInfo programmatically. With ArcObjects, developers can, for example, add new tools or work flows to the ArcInfo, ArcMap™, and ArcCatalog™ applications, or extend the ArcInfo data model by adding new custom feature types. These are only two examples of the many ways in which developers can build on, embed, or extend ArcInfo [5]. This set of components includes more than 1,200 objects that may be used to customize, extend, or construct GIS applications.

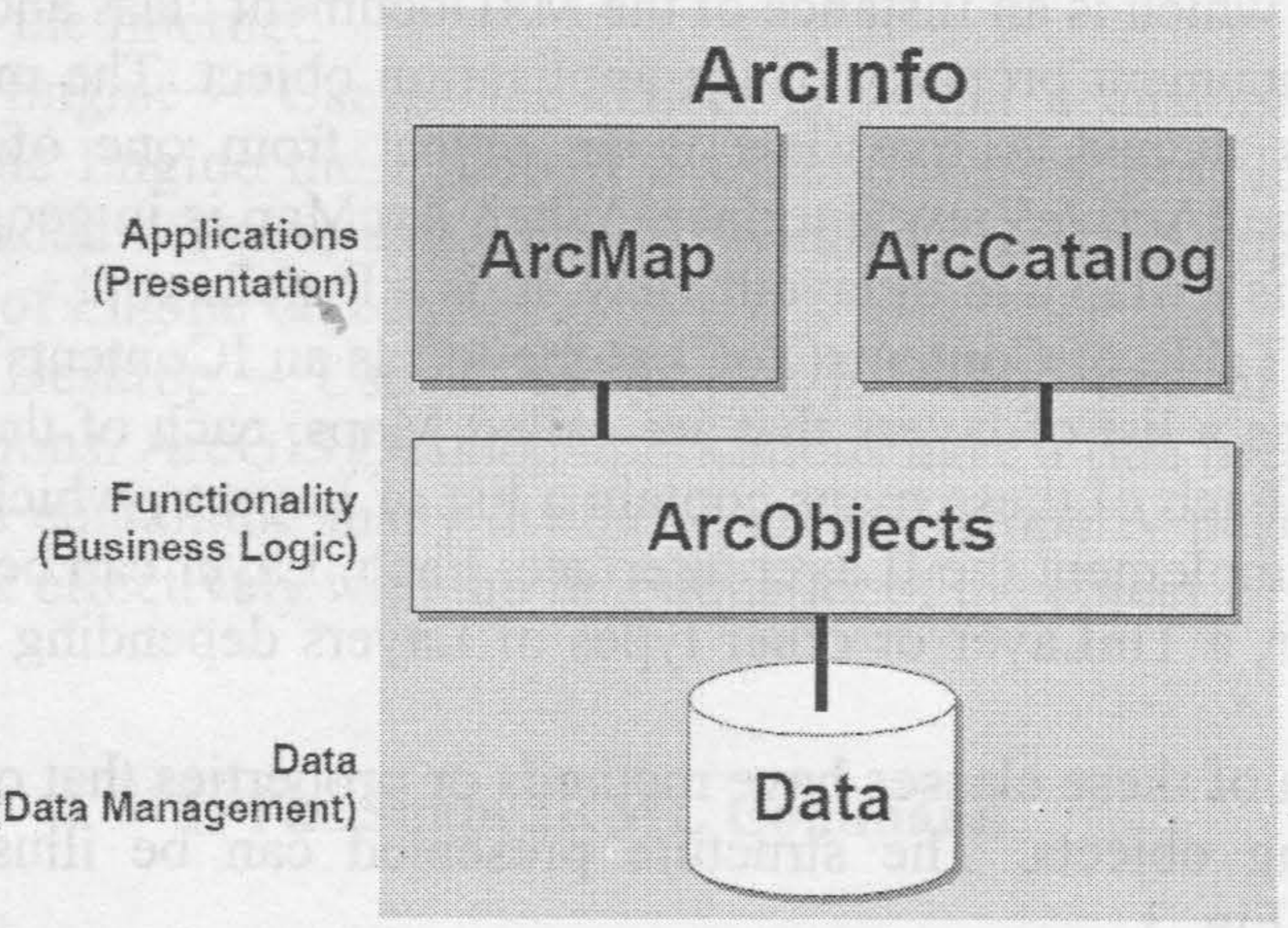

Fig. 1 – The place of ArcObjects in the ArcGIS platform.

ArcObjects is not sold separately; it is included with ArcInfo software like Avenue™ is an integral part of ArcView® 3 software. To build applications using ArcObjects, users must obtain a copy of ArcInfo and any derivative applications require a fully licensed ArcInfo seat [5].

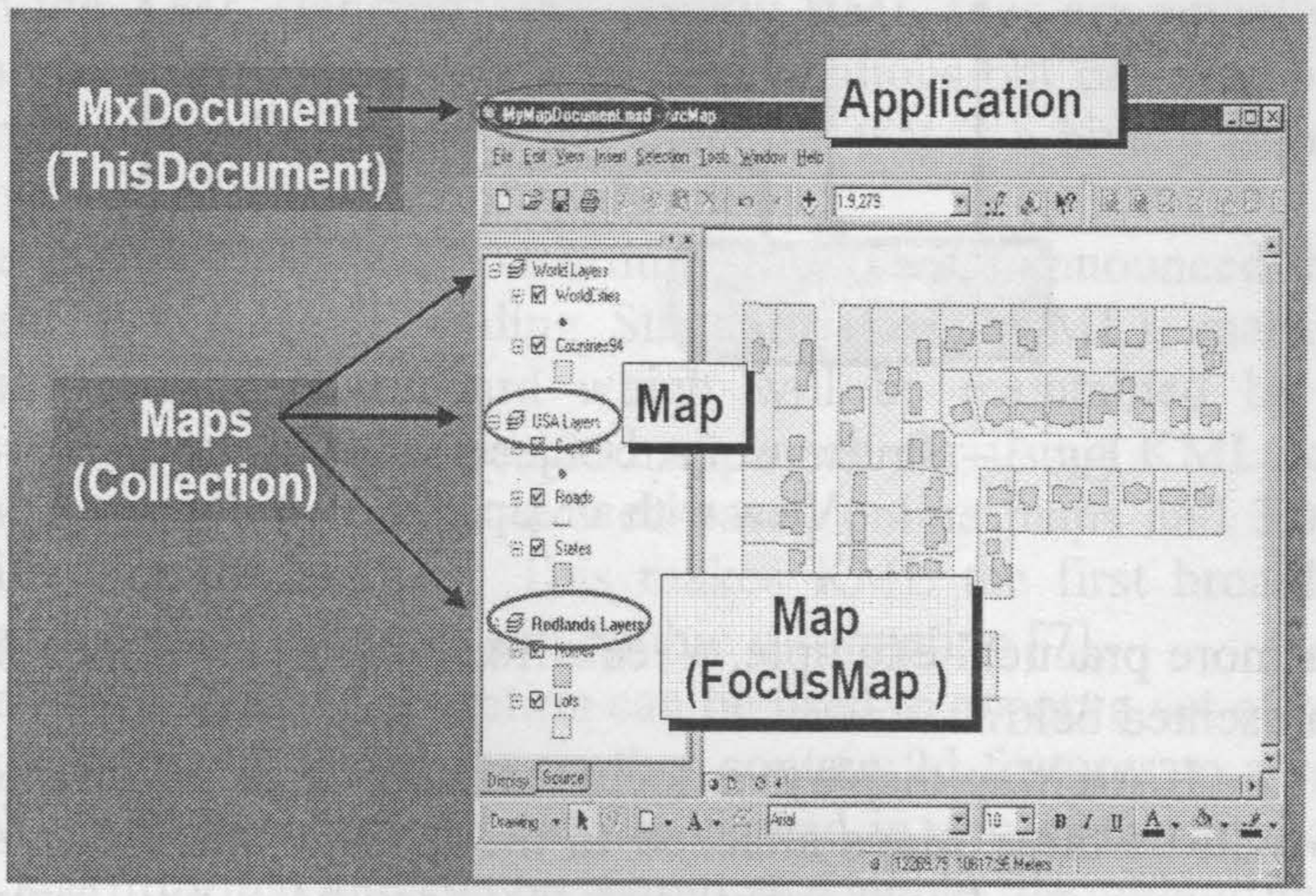

Fig. 2 – The ArcMap application and the main objects within it.

The main objects in ArcObjects built for the ArcGIS Desktop applications are structured similar to the applications in ArcGIS Desktop, so programming using ArcObjects is only harder until the programmer understands the structure and the relationships between these objects. The usual ArcMap window has a lot of objects that a programmer can use from it. Some of these objects are presented in Fig.2, and their classes are also presented.

The object that references all the other objects that a programmer can use is the Application object. This application can have a map document opened (a .mxd file) which is an instance of the MxDocument class and can be accessed using the Document property of the application object. The main part with the active layers named ActiveView is an object from one of the classes that implement the IActiveView interface. When ArcMap is in geographic view, the ActiveView is a Map and in layout view, it is a PageLayout.

The Table of Contents(ToC) at the left is an IContentsView object. The ToC contains a list of layers that are called Maps, each of these maps being a Map object. Each of these maps contain a list of Layers , which are instances of classes that implement the ILayer interface. Each Layer can be a RasterLayer, a FeatureLayer, a TinLayer or other types of Layers depending on what the user needs.

Each of these classes have methods or properties that offer access to the corresponding objects. The structure presented can be illustrated using the hierarchy in Fig. 3.

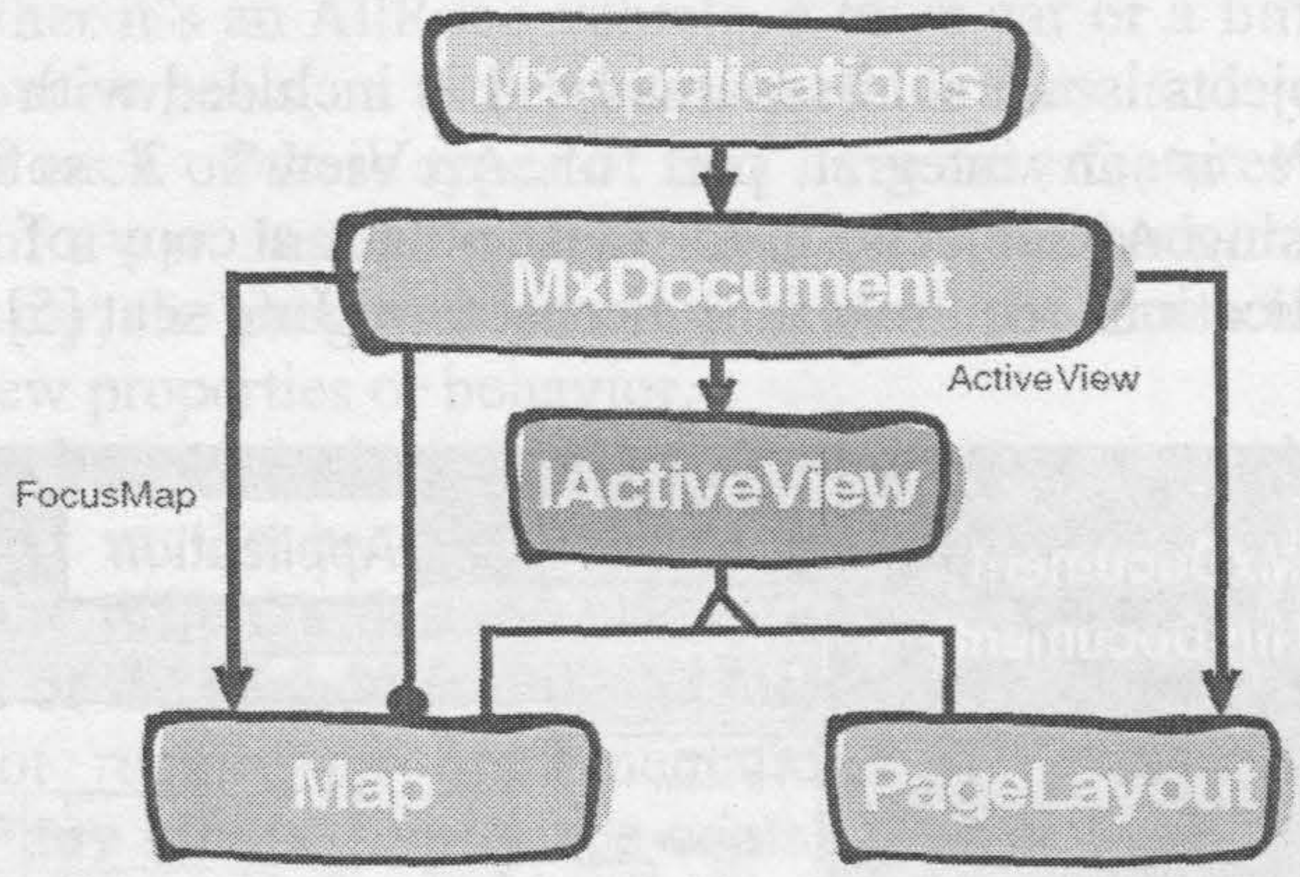

Fig. 3 – Structure of ArcObjects for a Layout View with a Map.

For a more practical example, a real-life situation corresponding to Figs. 2 and 3 is presented below.

If ArcMap displays a document with three data frames in layout view, there will be three Maps attached to the MxDocument, the ActiveView property will return the PageLayout, and the FocusMap will return the Map corresponding to the data frame that is highlighted [6].

Programming with ArcObjects is not limited to applications that work on top of the ArcGIS Desktop product family. Although the application presented here will only work with an existing ArcMap application and an ArcInfo license, any product created using ArcObjects will fall into one of the ArcGIS product families:

ArcGIS Server – The object is used within the server framework, where clients of the object are most often remote. The remoteness of the client can vary from local, possibly on the same machine or network, to distant, where clients can be on the Internet.

ArcGIS Engine – Use of the object is within a custom application. Objects within the Engine must support a variety of uses; simple map dialog boxes, multithreaded servers, and complex Windows desktop applications are all possible uses of Engine objects.

ArcGIS Desktop – Use of the object is within one of the ArcGIS Desktop applications. ArcGIS Desktop applications have a rich user experience, with applications containing many dialog boxes and property pages that allow end users to work effectively with the functionality of the object.

5. The Shape2KML Command

Shape2KML is an ArcMap command that was created using ArcObjects desktop development and can be used to export geographic data from ArcMap Layers to a KML document. This command has a user interface and a complex function, so it will be referred to as the Shape2KML application from now on.

KML is a open standard file format used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard. This is why KML files are actually formatted text that can be interpreted by different applications but can also be inspected using a simple text editor, unlike the shapefile that is a binary format and can only be viewed in a limited list of applications.

The Open Geospatial Consortium, Inc. (OGC) announced the approval of the OpenGIS® KML Encoding Standard (OGC KML), marking KML's transition into an open standard which will be maintained by the OGC. Developers will now have a standard approach for using KML to code and share visual geographic content in web-based online maps and 3D geospatial browsers like Google Earth®. This makes KML the first broadly accepted standard for the visualization of geographic information [7].

The Shape2KML application can be used to export a set of raster layers of type jpg and png or vector layers that contain 2d features to a single KML document. This KML document can be viewed in any of the applications that support the KML format and can be shared as a single file with all the data. The application interface is presented in Fig. 4 and it's very straightforward.

The left of the form has the list of Layers visible in the ToC, from which the user can select the ones that are exported to the KML file. The chosen layers appear in the Selected Layers dropdown list. For a vector layer, the user can choose where the altitude attributes can be extracted from. The altitude can be relative to ground or absolute, according to the KML specifications.

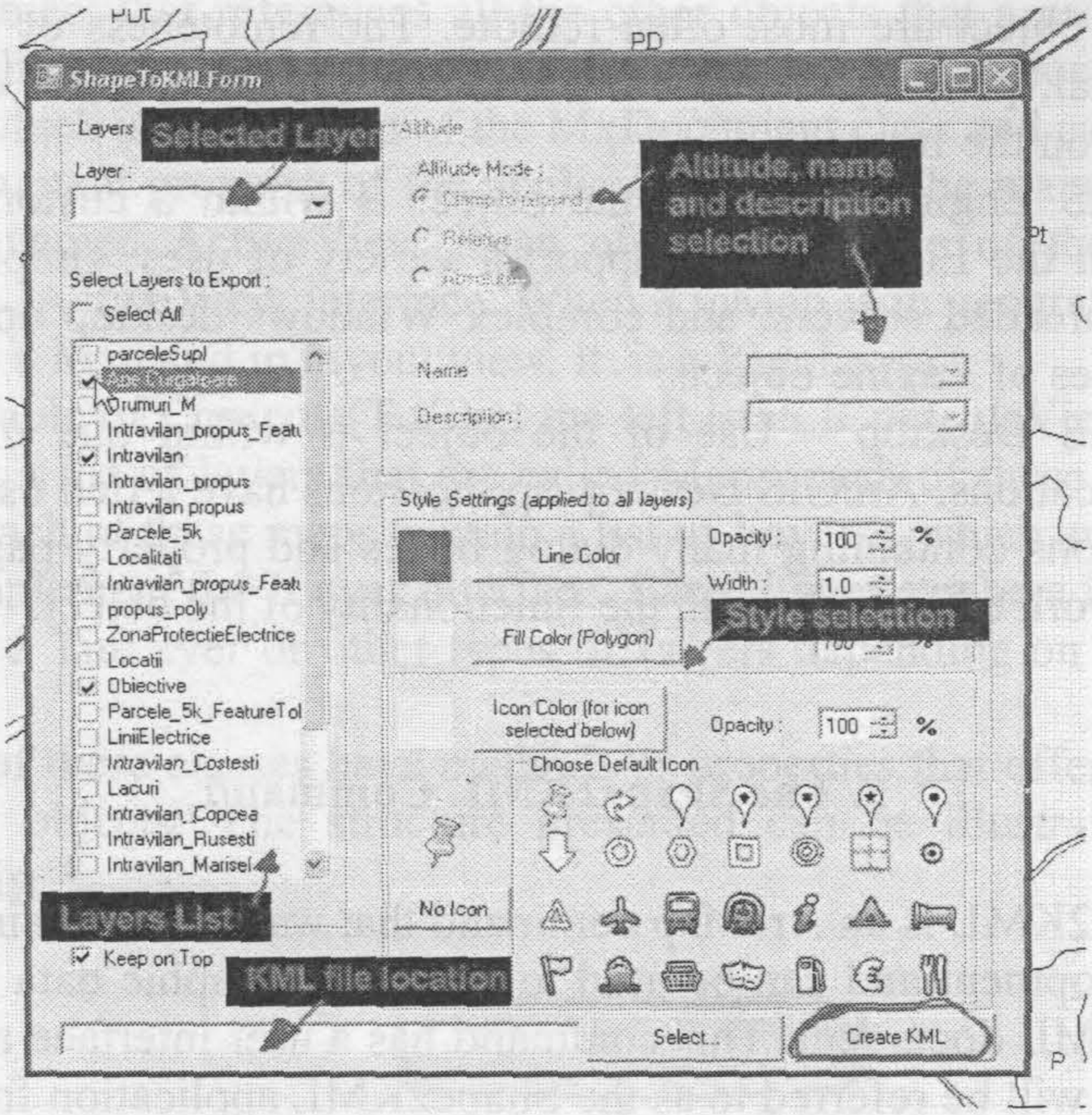

Fig. 4 – The Shape2KML user interface.

The lower right part of the form allows users to choose a style for the exported features. The style is composed of a color and other characteristics like width for a line or a specific icon for the point. The user chooses a location for the file and exports it. The application shows the progress in exporting the file.

6. Conclusions

The The exported file can be viewed in Google Earth ,viewed on the web in Google Maps or opened using any of the programs that support the KML format. Fig. 5 presents one of the files exported to KML representing contour lines for a terrain with the altitude extracted from the attribute table of the shapefile. As the example shows, the lines are exported correctly and they fit well on the elevation model generated by Google Earth from SRTM data.

The application also allows exporting raster layers at their correct location on the Earth surface for studying data from these layers related to data from Google Earth.

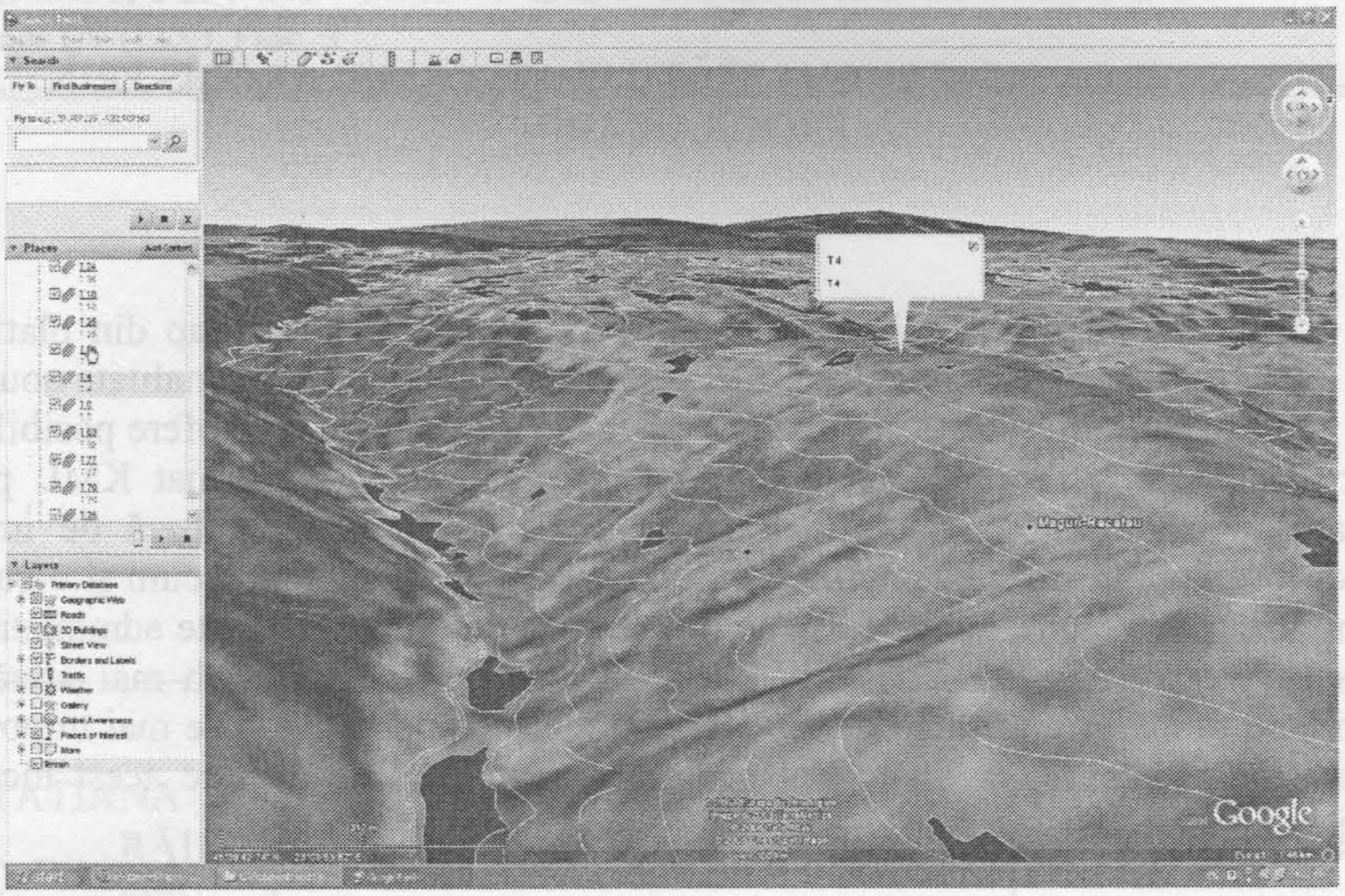

Fig. 5 – Conversion results.

Building this application allowed me to understand the ArcGIS object model and the way a developer can use ArcObjects to extend the ArcGIS desktop applications. Using ArcObjects more complex applications can be built on top of the ArcGIS applications or even independent from these.

The application and the source code were published on the ESRI developer network website to allow potential users to use it or study the source code and extend it or modify it by their needs.

Received: January 04, 2009

Babes-Bolyai University,Cluj-Napoca,
Faculty of Geography,
e-mail: mdomnita@gmail.com

REFERENCES

1. *** www.GIS.com Guide to Geographic Information Systems
2. Booth B., Mitchell A., *Getting Started with ArcGIS.*
3. *** ESRI ArcGIS web page http://www.esri.com/software/arcgis/
4. *** Msdn training - Module 7: *Essentials of Object-Oriented Programming* , 2001, Microsoft Corporation.
5. *** *Developing Applications with ArcInfo.: An Overview of ArcObjects.* An ESRI® Technical Paper . April 2000.
6. Jackson J., *Exploring ArcObjects* , www.esri.com, 2000.
7. *** OGC Approves KML as Open Standard http://www.gis-news.de/news/2008_04_16_ogc_kml.htm

EXTINDEREA FUNCȚIONALITĂȚII PLATFORMEI ARCGIS PRIN PROGRAMARE

(Rezumat)

Sunt prezentate posibilitățile de extindere a aplicației ArcMap din platforma ArcGIS prin Programare Orientată pe Obiecte pentru a obține o funcționalitate nouă.

Obiectivul lucrării este crearea unei extensii ArcMap care să ofere posibilitatea exportării unui set întreg de Layere din documentul curent în format KML pentru vizualizarea acestor date în Google Earth. Google Earth oferă o bază de imagini complexă formată din imagini satelitare obținute din diverse surse, precum şi o bază de date complexă pentru unele zone formată din rețeaua de drumuri, limite administrative, date despre vreme, date despre trafic. Pe lângă acestea Google Earth mai oferă şi o experiență 3d unică prin afişarea de modele tridimensionale la clădirile mai importante şi prin afişarea terenului ca un model tridimensional, dacă se doreşte acest lucru, în majoritatea zonelor.

În data de 14 aprilie 2008 formatul KML a fost acceptat de OGC (Open GeoSpatial Consortium) ca fiind un format "open" de descriere a datelor geografice, care poate fi folosit de către oricine fără a fi necesară plata unei taxe sau alte demersuri. De asemenea, acceptarea KML ca un standard în industria geospațială pentru descrierea datelor geografice măreşte popularitatea acestui format, permițând utilizarea lui în foarte multe aplicații Desktop sau Web care lucrează cu date geografice.

ArcMap nu oferă posibilitatea de a exporta datele în format KML, iar exportarea în acest format ar permite vizualizarea datelor folosind o mulțime de aplicații gratuite, precum şi vizualizarea lor pe internet fără a fi necesară instalarea vreunei aplicații. De asemenea, suprapunerea fişierelor KML peste datele existente deja în Google Earth poate oferi vizual diverse informații importante fără a fi necesară cumpărarea vreunui produs.

Extensia creată extinde funcționalitatea aplicației ArcMap prin adăugarea acestei operații de exportare care permite trimiterea informațiilor geografice pentru vizualizare în format KML către orice utilizator de calculator. Informațiile pot fi vizualizate folosind aplicații gratuite uşor de instalat sau interfața web Google Maps care nu necesită nici o instalare pe calculator. Astfel, datele pot fi transmise unor utilizatori care nu au o pregătire în domeniul GIS sau care nu au programele necesare pentru a deschide formatele proprietare care aparțin ESRI.