

A Strategy for Integrating Open Source GIS Toolboxes for Geoprocessing and Data Analysis

Yang Cao^a and Daniel P. Ames^a

*Geospatial Software Lab, Idaho State University, Idaho Falls, ID
Corresponding Author: Yang Cao, caoyang@isu.edu*

Abstract: This study presents an object-oriented and interface-based geospatial and hydrologic data processing and modeling toolbox and strategy for extension by third parties to include complete suites of new tools through the implementation of simple interfaces. The resulting software implementation includes both a toolbox and workflow designer or “model builder” constructed using the underlying open source DotSpatial library and MapWindow desktop GIS. The key contribution of this research and software development activity is in the creation and use of an extensibility architecture for both individual custom tools and batches of existing tools. This concept is introduced to allow for seamless integration of geoprocessing and hydrologic data analysis tools from various sources – including online sources – within a single user interface. In this way developers can “wrap” large existing collections of analytical code without having to re-write it for interoperability. Additionally, developers do not need to design the user interfaces for loading, displaying or interacting with their specific tools, but rather can simply implement the provided interfaces and have their tools and tool collections appear in the toolbox alongside other tools.

Keywords: Workflows, GIS modelling, spatial analysis, open source GIS, DotSpatial, MapWindow.

1 INTRODUCTION

Today’s fast-paced software development world is becoming increasingly characterized by taking advantage of existing software resources to the degree possible, rather than developing new code for every purpose. This is particularly true in the world of open source software. Currently, there are thousands of free software packages that can be accessed via open source project hosting websites such as SourceForge.net and CodePlex.com. Free and open source software supports and encourages software reuse based on specific open source licenses, which generally allow the third party developers to use, modify, and redistribute the code. Reuse of existing free software has become a hot topic in the software community because of its potential benefits, including reduced code implementation time, cost, and improved software quality. Ramel (2005) proposed that the most frequent type of open source software reuse is component integration. Software component integration is the technology that combines software components into an overall system so as to provide overarching functionality. Component integration falls into three categories: full integration, loose coupling and tight coupling (Andrade Neto et al., 2005).

Although extensive academic and practical research has explored GIS software integration with respect to loose coupling and tight coupling mechanisms (Bao et al., 2000; Eldrandaly et al., 2005; Sui and Maggio, 1999; Whiteaker et al., 2012), much less research has investigated component integration with the full integration

technology (Rubio, 2003). Full integration is direct incorporation of a third party toolkit fully within a GIS (Goodchild, 1992). Although full integration does require significant programming effort, this approach allows the different software packages seamlessly to integrate together into a unified software package without needing to recode all of the analytical tools in the third party software.

This paper presents a full system integration approach within a GIS environment that is accomplished by integrating the existing spatial analysis tools with interface-based programming. Such an approach makes the full integration easy by the implementation of simple interfaces, thereby significantly supporting the effective incorporation of batches of existing tools from a third party into the GIS tools. The full integration approach is illustrated by the way in which Whitebox tools is closely integrated into the GIS libraries called DotSpatial using interface-based programming. Furthermore, this approach shows a way to incorporate object-oriented third party tools into their own application.

2 DotSpatial

The geospatial analysis tool component used in this research comes from the open source DotSpatial library (<http://dotspatial.codeplex.com/>). The DotSpatial project has been under development by members from the Geospatial Software Lab at Idaho State University, Idaho Falls, Idaho, USA, as well as an international open source volunteer community and members of the MapWindow project (<http://mapwindow.org/>) since April 2010. Since its first release, DotSpatial has been downloaded over 40,000 times and it currently receives approximately 200 downloads per day by users and developers for exploring the free and open source alternatives for GIS enabled custom software.

DotSpatial aims to provide a free, open source, set of libraries for the .NET, Silverlight and Mono platforms that is consistent and dependable. This set enables developers to easily incorporate spatial data, analysis, and mapping into their applications thereby unleashing the massive potential of GIS in solutions for organizations and communities of all types in a nonrestrictive way. DotSpatial is being used by a lot of open source GIS development including LiDAR applications (Yang et al, 2011) and water resource management and hydrological modeling (Tarboton et al, 2011). It is being developed by members of the MapWindow GIS open source team, and the broader OSGeo DotNet developer community (Ames et al., 2007).

The DotSpatial engine used by in this project comprises two layers of structure (Figure 1) including:

Windows Forms User Interface Layer: this layer comprises basic forms and resources which include Shape Editor application, Modeling Forms, Symbology Forms, Data Forms, Projections Forms and Positioning Forms. These contribute to the construction of a variety of GIS functions.

Business Layer: the business layer appears as a series of libraries which including the function from modeling, symbology, data manipulation, GDAL extension, topology, projection to positioning and serialization.

Currently, the 30 tools available in the DotSpatial library are classified into 9 categories: Analysis, Terrain Analysis, Conversion, Spatial Reference, Vector Overlay, Raster Overlay, Interpolation, Statistics and Generalization. The DotSpatial's toolbox control is an extensible tree-view of geoprocessing functions that have been wrapped in the ITool interface. It is able to recognize ITool implementations in the "Tools" directory in the executable's directory. In order to

add your own tools, just implement the ITool interface, build your dll and place that library in the Tools directory.

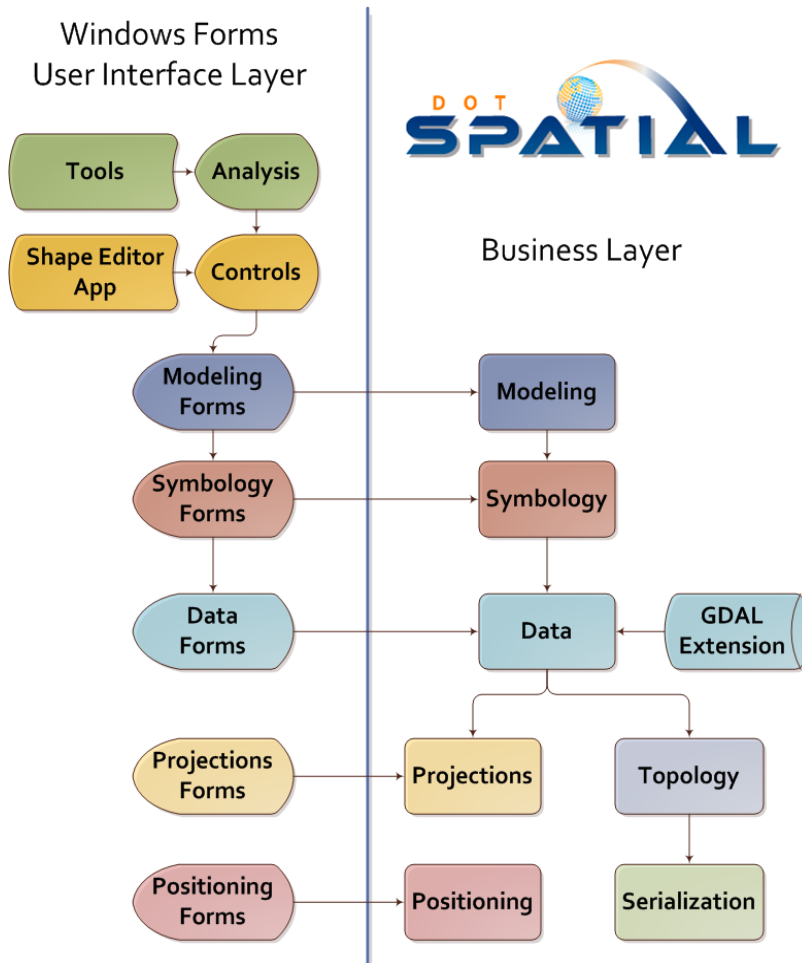


Figure 1. DotSpatial Structure (source: <http://dotspatial.codeplex.com/>)

3 WHITEBOX GAT

Whitebox Geospatial Analysis Tools (Whitebox GAT) is an open source GIS and Remote Sensing software package . It has been developed at the University of Guelph's Centre for Hydrogeomatics under the GNU General Public License. Whitebox GAT is based on Window's .NET framework and programmed in the Visual Basic and C#. Whitebox GAT is able to read and process much larger spatial datasets, such as LIDAR data. It allows the developer to create new plugin tools and extensions using built-in Python scripting.

Currently, there are more than 180 tools available in Whitebox GAT in 11 categories: Conversion Tools, Data Import/Export, GIS Analysis, Hydrological Tools, Image Processing Tools, LiDAR Tools, Mathematical Analysis, Raster Creation, Statistical Analysis, Stream Network Analysis and Terrain Analysis.

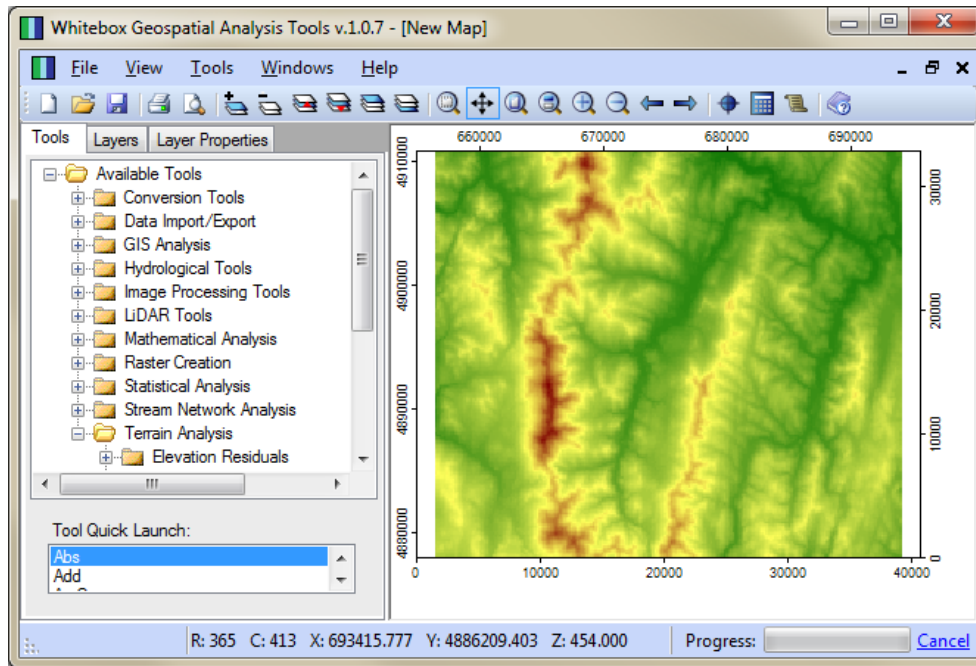


Figure 2. Whitebox GAT Graphic User Interface showing eleven tools categories (source: <http://www.uoguelph.ca/~hydrogeo/Whitebox/>)

4 FULL INTEGRATION STRATEGY

The full integration strategy presented here is based on implementing the simple interfaces within DotSpatial. In this case, two interfaces are used: *ITool* and *IToolProvider*. *ITool* allows the developers not to create a graphic user interface when they create a tool in DotSpatial. It also helps to maintain the tool interoperability. *IToolProvider* generates the graphical user interface automatically when the DotSpatial tool is instantiated.

DotSpatial's toolbox is greatly enriched by incorporating Whitebox GAT to introduce more tools, such as complex map algebra, stream and watershed extraction, interpolation, and digital terrain analysis.

The process of integrating Whitebox GAT into DotSpatial entails three phases:

- 1) Reference all Whitebox plug-in dlls into DotSpatial
- 2) Implement Class *WhiteboxToolProvider* which specifies which tools to load in DotSpatial
- 3) Implement Class *WhiteboxTool* which implement *ITool* interface

There are a total of 20 library assemblies of Whitebox GAT that need to be referenced in DotSpatial, including: *GISTools.dll*, *Interfaces.dll*, *LiDAR_tools.dll*, *MathAnalysis.dll*, etc. By referencing the library assemblies, DotSpatial know the information of the Whitebox GAT's tools.

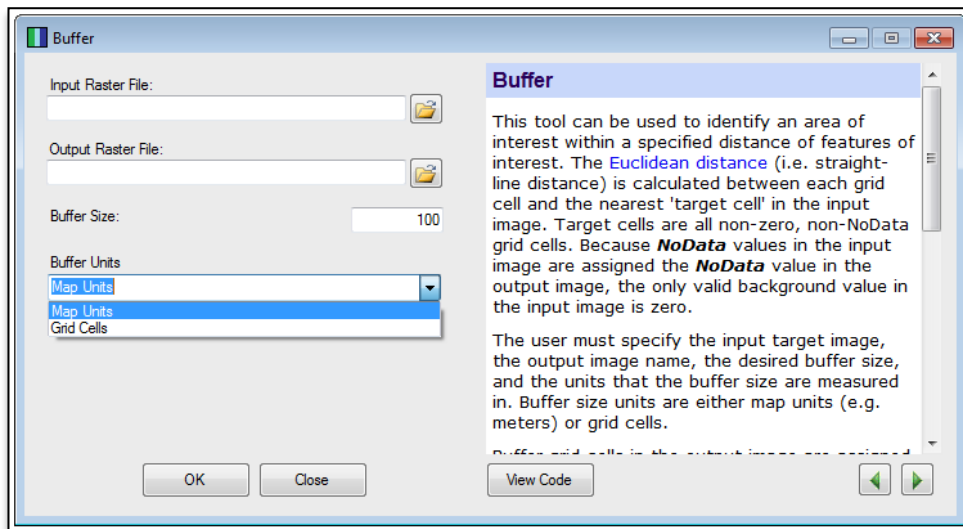
By simply adding one line of code in the *GetTools()* method in the *WhiteboxTool* Class, the developer can integrate one Whitebox GAT tool into DotSpatial. For example, the aim of the one line code shown below is to incorporate the *Buffer* tool from Whitebox GAT to DotSpatial.

```
yield return new WhiteboxTool(new GISTools.Buffer());
```

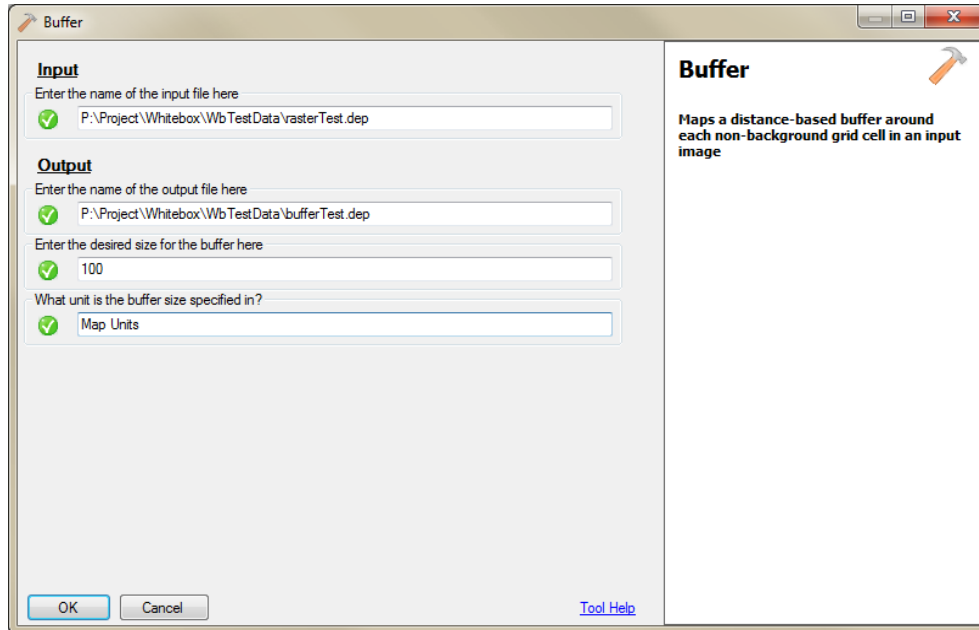
Currently, the 136 Whitebox GAT tools are accessible in the Online tab of DotSpatial Extension Manager. The end user who has the DotSpatial application can utilize the Whitebox GAT Tools with its Extension Manager.

The following is an example of the code written to implement ITool (partly shown because of the space):

```
public class WhiteboxTool : ITool
{
    private Interfaces.IPlugin _plugin;
    private Parameter[] _inputParam;
    private Parameter[] _outputParam;
    ...
    public bool Execute(DotSpatial.Data.ICancelProgressHandler
cancelProgressHandler)
    {
        BackgroundWorker worker = new BackgroundWorker();
        _wbHost.ProgressHandler = cancelProgressHandler;
        _plugin.Execute(_wbHost.ParameterArray, worker);
        ...
        return true;
    }
    ...
}
```



a) Whitebox GAT's Buffer tool shown in Whitebox GAT



b) Whitebox GAT's Buffer tool shown in DotSpatial

Figure 3. Illustrates the difference between the Buffer tool shown in the Whitebox GAT form and the Buffer tool in the automatically generated in DotSpatial form.

5 CONCLUSIONS

This paper briefly introduces the full integration of two geospatial analysis tools using software interfaces. This is illustrated by the implementation of the integration of Whitebox GAT and DotSpatial. This full integration provides an exciting approach for the seamless integration of GIS software which is also highly efficient and allows for the developers to conduct the software integration easily. The implementation of Whitebox GAT and DotSpatial in this study can serve as an example to demonstrate how the full integration of GIS software could be integrated.

It is noted that the implementation speed of the full integration may not be as fast as the loose coupling or tight coupling due to software design issue, because different software packages have their own readable data format. To deal with the problems, one solution is to use common standard GIS data format. We are looking forward to such a GIS software development environment that allows more interoperability for facilitating system integration.

REFERENCES

- Ames, D.P., C. Michaelis, and T. Dunsford, , *Introducing the MapWindow GIS Project*,. *OSGeo Journal*, Vol. 2: 8-10, 2007.
- Andrade Neto, P. R., P. J. Ribeiro Jr, , and K.D. Fook, , Integration of statistics and Geographic Information Systems: the R/TerraLib case. VII Brazilian Symposium on Geoinformatics, GeoInfo2005 Campos do Jord.ao, Brazil, November 20-23, 2005.
- Bao, S., L. Anselin, D. Martin, and D. Stralberg, *Seamless integration of spatial statistics and GIS: The S-Plus for ArcView and the S+Grassland Links*, *Journal of Geographical Systems*, 2: 287–306,2000.

- Cao Y., D.P. Ames, and P. Yang, Development of an Extensible Geospatial Analysis Toolbox, paper presented at American Water Resources Association (AWRA) Spring Specialty Conference Proceedings, Water Integ Track, New Orleans, Louisiana, USA, March 26 - 28, 2012.
- Eldrandaly, K., N. Eldin, D. Sui, M. Shouman, and G. Nawara, , *Integrating GIS and MCDM using COM technology. The International Arab Journal of Information Technology* 2(2):162-167, 2005.
- Goodchild, M., R. Haining,, and S. Wise., *Integrating GIS and spatial data analysis: problems and possibilities. International Journal of Geographical Information Systems*, 6(5):407-423, 1992.
- Ramel, S, Software Reuse in Free Software: State-of-the-Art. ODETTE project report, Centre de Recherche Public Henri Tudor, 2005.
- Rubio, V. G, RARcInfo: using G.I.S. data with R. paper presented at the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), Vienna, Austria, March 20-22, 2003.
- Sui, D.Z., and R.C. Maggio, *Integrating GIS with hydrological modeling: practices, problems, and prospects. Computers, Environment & Urban Systems*, 23, 33-51, 1999.
- Tarboton, D.G D. R. Maidment, I. Zaslavsky, D. P. Ames, J. Goodall, R. P. Hooper, J. S. Horsburgh, D. Valentine, T. Whiteaker, and K. Schreuders, Data Interoperability in the Hydrologic Sciences, paper presented at the Environmental Information Management Conference, Santa Barbara, CA, U.S.A., September 28-29, 2011.
- Whiteaker,T., Y. Cao, and A.M. Castronova, 2012. Using hydrodesktop to share hydrologic data and analyses. paper presented at American Water Resources Association (AWRA) Spring Specialty Conference, New Orleans, Louisiana, USA, March 26 - 28, 2012.
- Yang P., D.P. Ames, and R.Shrestha, Spatiotemporal analysis of stream network structure based on snow-on and snow-off LiDAR, Abstract EP41A-0582 presented at American Geophysical Union (AGU) Fall Meeting , , San Francisco, California, U.S.A, December 5-9, 2011.