Automated Map Production Workflows

Aileen Buckley and David Watkins
abuckley@esri.com and dwatkins@esri.com
ESRI, Inc., U.S.A.

Abstract

As noted cartographer Professor Waldo Tobler wrote in 1959, "Automation, it would seem, is here to stay." Fifty years ago, Tobler clearly recognized the advantages that automation offers cartography in terms of increased speed and improved quality. Since then, the infrastructure that supports map making has continued to evolve, and we find ourselves regularly seeking the answers to the same basic questions Tobler asked a half century ago: "What possibilities exist for automation in cartography, and where can these be found?" The answers help us improve the map production process so that it requires less time, provides more consistency (and therefore higher quality), and results in reduced costs. But the answers are also constantly changing as hardware capabilities, software functionality, and other technical constraints and opportunities continue to shift and evolve. In this paper, we review the current state of automation in the context of GIS database-driven cartography, and we provide specific examples of situations in which automation can be capitalized upon in map production workflows.

Introduction

This paper describes automated map production workflows that have been developed or are being developed. Before we start, it would be helpful to clearly describe what we mean by "automated map production workflows". We can do this by parsing the term into its constituent words. "Automation" is the operation or control of equipment, a process, or a system by a machine rather than by hand. "Map production" includes map compilation, or "assembling and fitting together the geographical data you will include in your map", as well as other elements on the page (Robinson, et al. 1995, p. 426. Map production also involves map construction in which the map is placed on the page or multiple related maps are placed on multiple pages, and associated elements like graphs or tables are added to the page or pages. In addition, map production involved the output of a final product. A "workflow" is a process and/or procedure in which certain tasks are completed. So, "automated map production workflows" are machine-driven processes that result in the completion of tasks that relate to the compilation, construction, or output of a map product.

In their description of the map production process, Robinson et al. (1995, p. 426) state, "It's important to compile the map in a way that will make the ultimate map construction as easy as possible." Assuming that automation makes map production easier, having ease of construction as a priority leads to the natural conclusion that automation might be a useful thing to introduce into the map production process at every possible step. In order to do this, the map maker must know a number of things about the map from the outset, including:

- the sources of data and the associated considerations for the data, including spatial accuracy, temporal currency, permissions and copyright, cost, etc.,
- the map projection, extent, scale, features to be mapped, mapping method (e.g., choropleth, isoline, hillshade, etc.),
- the parameters for cartographic data processing and display (e.g., selection, classification, symbolization, generalization), and
- the techniques that will be used for presentation of the final map (Web map, printed map, etc.)

Some level of automation is possible in the map production processes that relate to each of these (see, for example: Buckley, 2005; Eicher et al., 2007; Buttenfield and Frye, 2007; Trainor and Spahlinger, 2008); however we have yet to realize the completely automated map -- the mythical "Make Map" button continues to escape us! While much progress has been made in the automation of certain map making steps, such as label placement, contour generation, generalization, and even in some types of automated feature extraction, we are still unable to fully automate the processes that cartographers engages in with their human intellect and their eye for design. Although progress continues to be made in these more elusive areas, advancements come only slowly over time.

Nonetheless, there are many aspects of map production in which automation can be applied to provide fully computerized results. These areas are usually related to either manipulation of the data as a prerequisite to map compilation, map construction (adding the map to the page or manipulating the elements on the page), and producing the output product. While individual, or "one-off", maps are not usually good candidates for the types of automation currently available, automated workflows can be more seriously considered if the same mapping specifications are applied to more than one map. In these cases, many aspects of the map production process can be automated, including data processing, the construction of map book or atlas pages, as well as exporting and printing the final product.

There are a variety of candidate cartographic products for automated map production workflows, which we summarize as:

- many maps with the same theme but different extents,
- many maps with the same extent but different themes, and
- many maps with the same extent and the same theme, but the data for the theme changes over time.

An example of the first case, a set of maps with the same theme but different extents, is the City of Portland Street Atlas, is shown in *Figure 1*. The maps cover contiguous portions of the city at a scale of 1:10,000. The same data layers are displayed on each map, and a similar layout is used for each page – only the area being mapped changes. Other elements on the pages, such as titles, locator maps, and page numbers could be modified automatically. The second candidate product for map automation is a map series with a single extent but multiple themes. For example, an ecological study for a particular geographic region may include maps with themes for wildlife habitat, hydrography, temperature ranges, vegetation, and more -- all of these for the same area. In this situation, the page layout would stay largely the same, but the layers on the maps would change. In these cases, minor modifications could automatically be made to the map elements on the page, such as legends, titles, and related text or graphics. The third candidate cartographic product for automation is a set of maps with the same extent and the same theme, but the maps show change in the thematic variable over time. In this scenario, the map layers must be revised and the layout for each map might require modification, but little else changes.

For all three of these candidates for automation, a single map laid out on a single page is the starting point. The layout can then be modified, text elements and map features can be updated, and themes can be turned on or off – all through automation. In addition to maps and page layouts, other candidates for automation are reports or publications whose content includes text, graphs, and charts that will accompany the map or maps. Using automation, it is possible to compile the maps, along with one or many of these other types of documents, into a single output file to create a multi-page product. In this scenario, pages with different layouts and content can be combined into a single cartographic product.

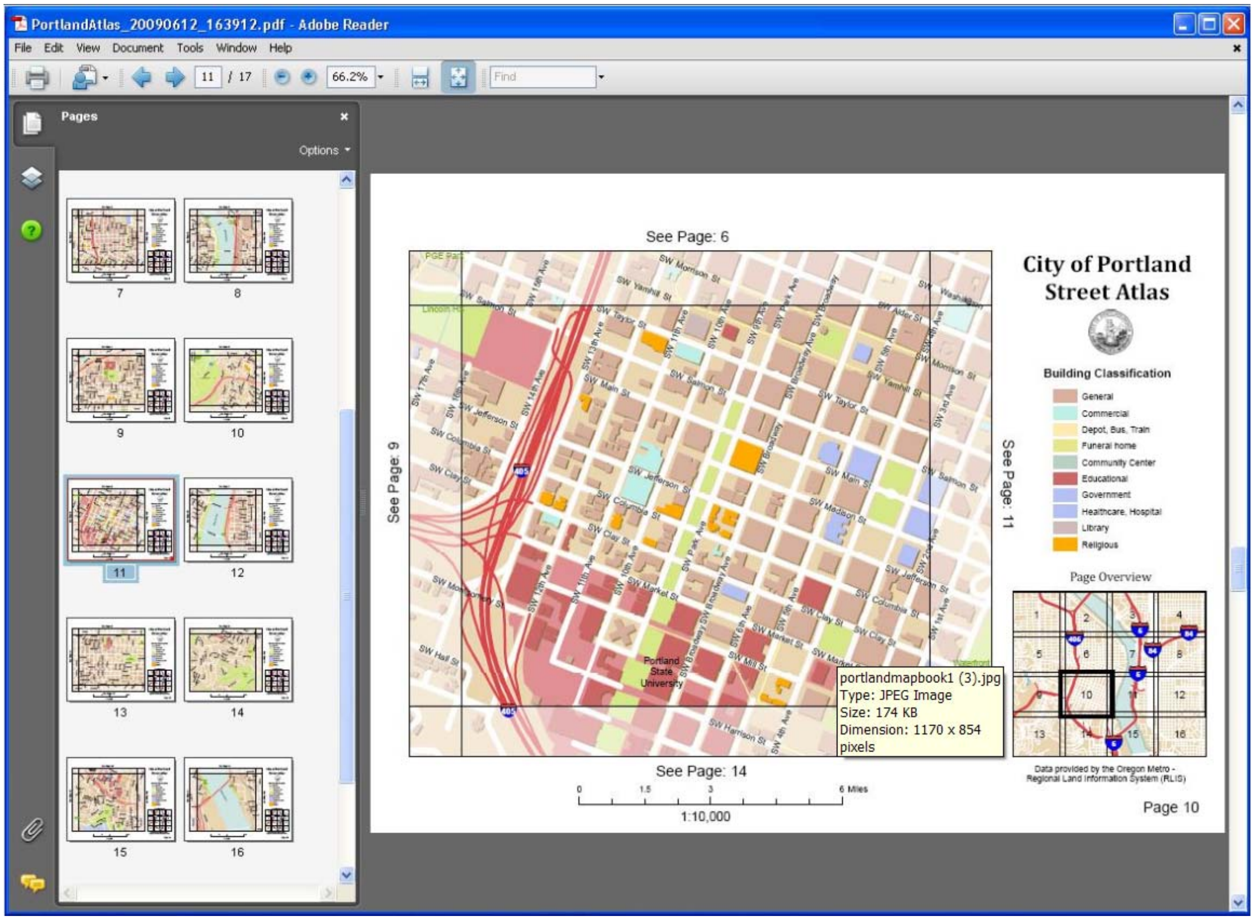

Figure 1. An example of a cartographic product that is comprised of many maps of the same theme with different extents.

In the remainder of this paper we provide more concrete examples of how map production workflows have been or can be automated using software developed for spatial database management and map production. Specifically, we describe automation opportunities using ArcGIS software by ESRI.

Automation with GIS software

Automation is usually achieved through the use of programming or a scripting language. From the release of ESRI's initial GIS software, ArcInfo Workstation, to the current release of ArcGIS and ArcGIS Server (for online maps), the ability to automate cartographic workflows through scripts and programming languages has been supported. ArcInfo Workstation provided automation capability through the Arc Macro Language (AML) – a scripting language that made it possible to automate the compilation of the map, the construction of page layouts, and the output of the cartographic product to a digital file or printing device. AML could be used to build a map product "from the ground up", providing the map maker with the a very high level of specificity (e.g., drawing a line from one x,y position to another on the page.) The AML macro language gave a large amount of control to the person writing the script, but the user experience was not very friendly. The interface was command-line driven, and the AML scripts had to be written as text files. Even drawing a simple map automatically required writing a somewhat lengthy script. Many map makers were willing to overlook these limitations because of the previously-unavailable opportunities for automation. However, what could not be ignored was the insufficient quality required for high-end map production.

ESRI later released ArcView, a Windows-based product designed with a more user-friendly interface. This software included the ability to view the map in a page layout and the ability to manipulate map data and map elements with standard Windows tools and Windows-like toolbars. Users found the initial compilation of the map, which is often an iterative and exploratory process, much easier with the improved graphic user interface (GUI). With ArcView, ESRI also released the Avenue scripting language. This language made it possible to automate limited sets of maps making tasks, such as adding/removing or turning on/off data layers, manipulating some map elements on the page layout, and outputting the map. As a result, mapping with ESRI software took on the character of "compile the initial map, then use a script to automate the production of a replica". While the user experience improved greatly for the compilation of the original map, ArcView with Avenue did not provide nearly as much automation capability as ArcInfo with AML. Additionally, cartographic capabilities required for high-end cartographic products were still missing to a large extent. As a result, ArcView was not adopted by those with serious map production requirements, such as commercial map making companies or national mapping agencies.

Nonetheless, both ArcInfo and ArcView provided cartographers with more capabilities to automate map production workflows than they had with most other software packages. With AML, it was possible to compile the entire map and output the results – for single or multiple maps. With Avenue it was possible to modify the content of compiled maps and page layouts and produce the output. Perhaps the biggest advantage that both provided was the ability to, in many ways, automatically manipulate the data used to make the maps to prepare it for mapping. This was achieved by taking advantage of the GIS database associated with the mapped features – a characteristic that is absent from illustration and CAD software. Lessons learned from ArcInfo and ArcView were incorporated into the development of the current ESRI GIS and mapping software – ArcGIS.

ArcGIS was ESRI's first Component Object Model (COM)-based software product. The main application in ArcGIS is ArcMap, which is used for all mapping and editing tasks as well as for map-based query and analysis. ArcGIS was released with ArcObjects, an object model providing extensive access to the methods and properties used to create the software. This provided the ability for extensive custom automation through any COM-based programming language. Common among these are Visual Basic for Applications (VBA), Visual Basic (VB), C++, C#, and .NET. Many custom tools for automating map production workflows have been written using ArcObjects, including a Map Book developer sample and the Map Production System-Atlas component of the Production Line Tool Set (PLTS). ArcObjects made it possible to create very powerful automation tools, but the object model is large and the programming languages require more knowledge and expertise to use than traditional scripting languages. This put ArcObjects out of reach for many cartographers.

With ArcGIS, ESRI also released Model Builder as part of the geoprocessing framework, which is used to automate GIS tasks and perform spatial analysis and modeling. Model Builder allows users to visually drag/drop and connect geoprocessing tools to create a workflow. These models enable the automation of many processes without the requirement of any programming language, so cartographers once again found themselves drawn to core software capabilities to automate at least some of their map making tasks. Users can also use Python, an object-oriented scripting language, with the geoprocessing tools to create scripts that automate the processes reflected in the models. An existing model can also be saved in Python code to create a script which can then be easily edited and re-run. Python is much simpler than programming languages like C++ and can be learned in a few days, so cartographers do not need to be software developers to write scripts. Also important to cartographers was the introduction of cartographic finishing capabilities – many of the requirements for high-end cartography could now be achieved without having to export the maps to illustration software (Brewer, 2005; Hardy and Lee, 2005).

For cartography, scripts and models can be developed to process the data required to create the maps, and then add the layers to the map so that they can then be symbolized. An example is the Swiss Hillshade Effect model (*Figure 2*) which creates a display of raster data that emulates a hillshading method described by Eduard Imhof (Imhof, 2007; Barnes 2003), as shown in *Figure 3*. A second example is the multi-directional oblique weighting (MDOW) model (as described by Mark, 1992; *Figure 4*), which generates a shaded relief image that represents a combination of hillshades illuminated from four different azimuths (225°, 270°, 315°, and 360°) to emphasize the oblique illumination on any surface. The results of this model are shown in *Figure 5*. A third example is the Bump Map model that creates textures for vegetated areas on a raster hillshaded surface (as described by Jeff Nighbert, 2003, 2006, 2007). The results of this model are shown in *Figure 6*. A fourth example is the Sat-Val-Mod model that replaces the value of a theme overlaid on a hillshaded surface with the lightness or darkness of the hillshade (as described by Viljoen, 2006).

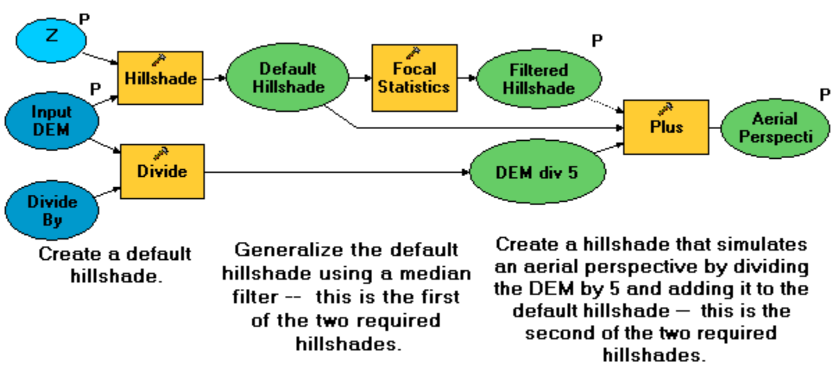

Figure 2. The Swiss Hillshade Effect model.

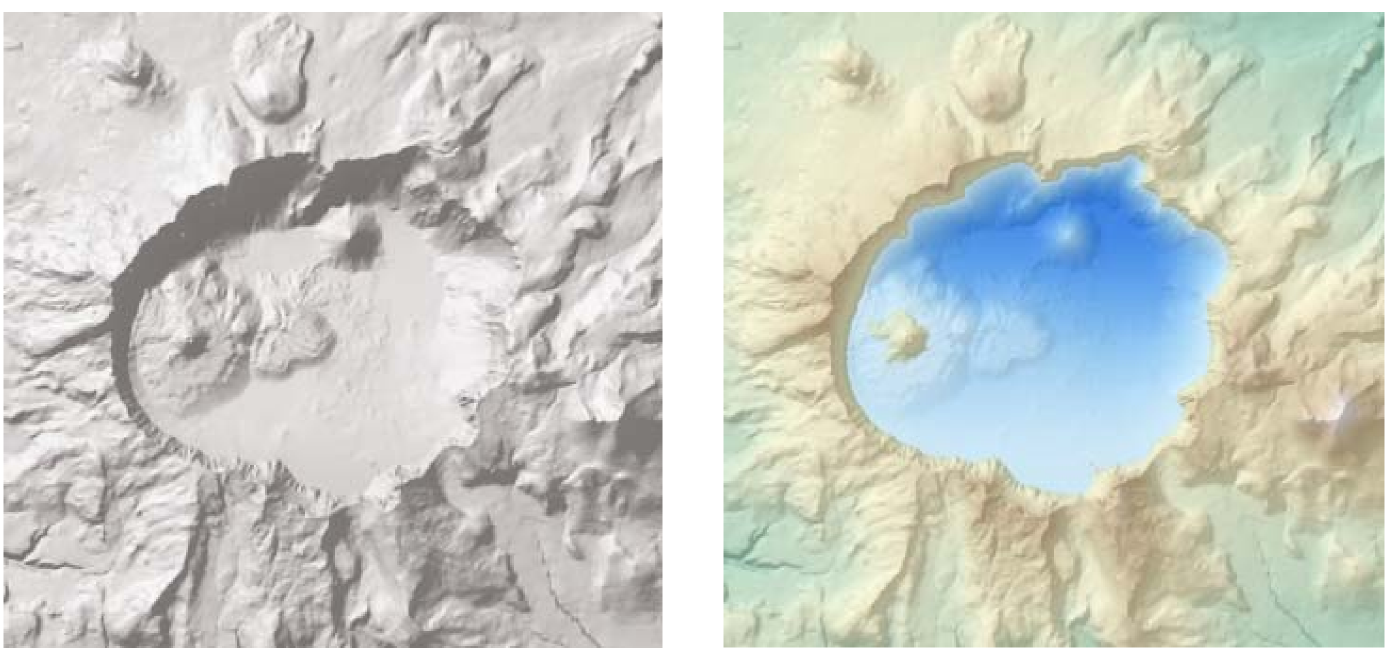

Figure 3. The results of the Swiss Hillshade Effect model, with and without the layer tint. The bathymetry is displayed using the same method.

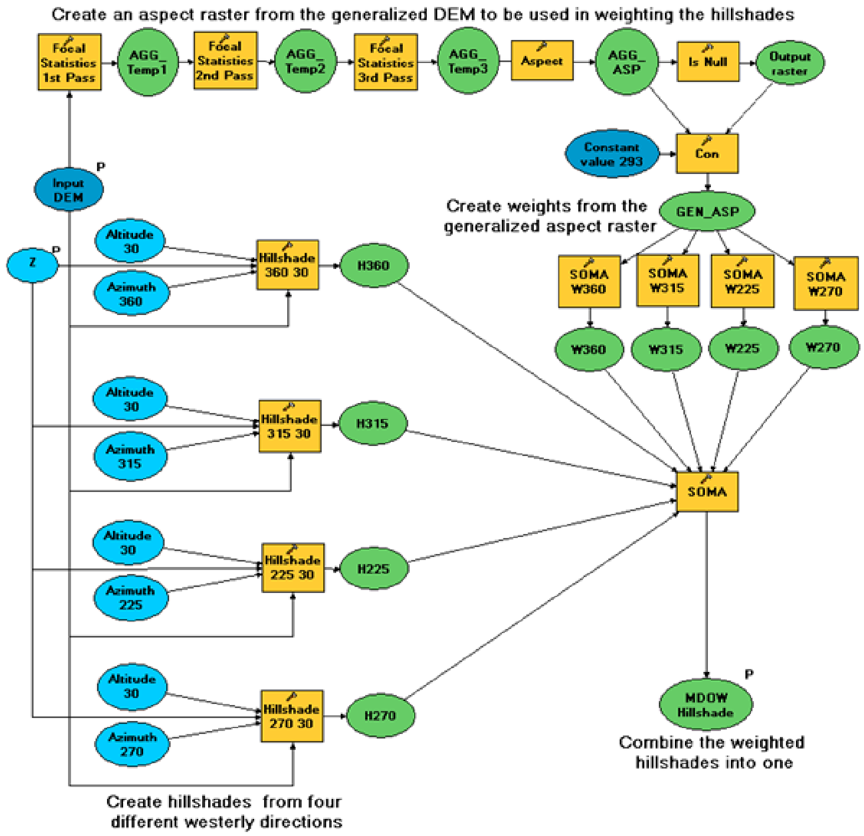

Figure 4. The MDOW model.

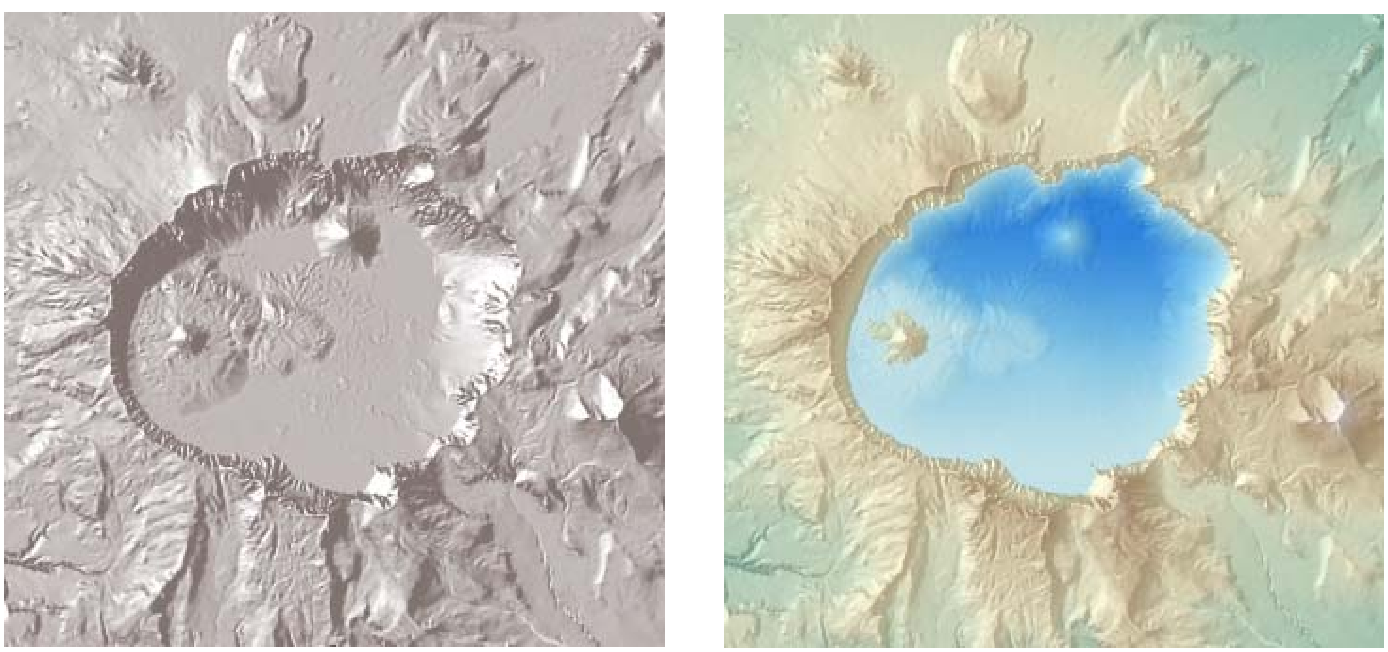

Figure 5. The results of the MDOW model, with and without the layer tint.

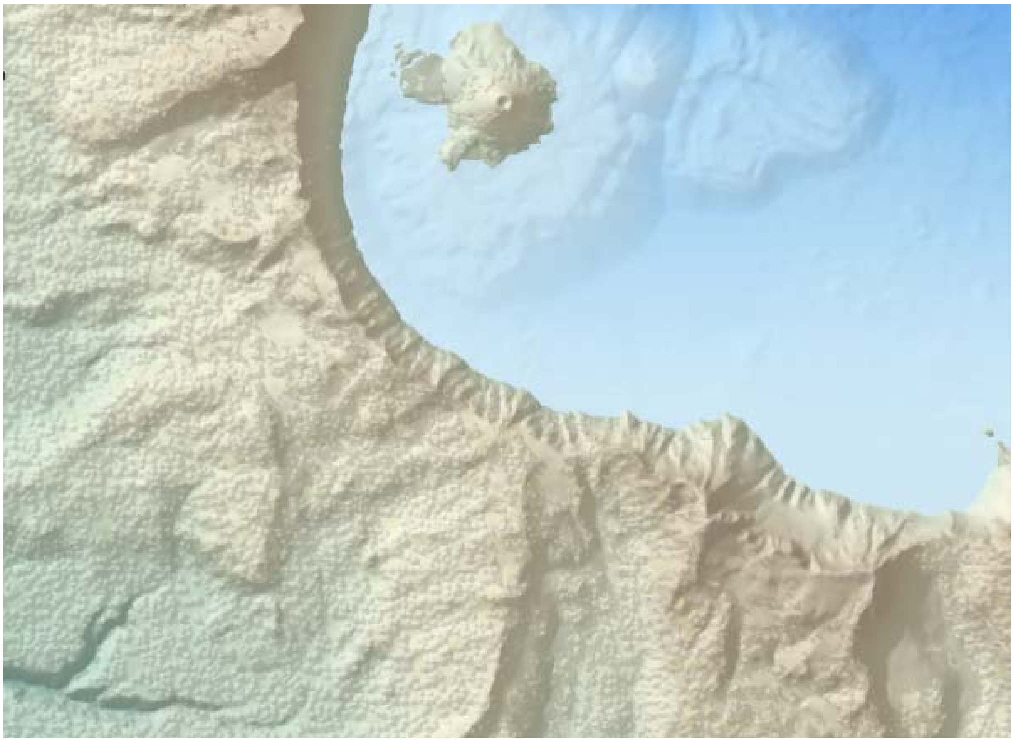
Figure 6. The results of the Bump Map model.

The future release of ArcGIS, ArcGIS 9.4, will include an expansion of the geoprocessing framework to enable the manipulation of map documents, map layouts, and layer files (files that reference the geographic data and assign the symbology in the display) through a new Python site package called "arcpy". A site package is essentially a software development kit (SDK), which is a set of development tools that allows a user to create applications to perform a particular set of tasks. The arcpy site package will provide a new set of map automation capabilities through the arcpy.mapping module - the portion of the arcpy SDK that relates to the mapping. For example, using a Python script, a cartographer can automatically open a map document, then update a string of text on the page, change a data layer on the map, modify the symbology used, and export the map to a digital output, such as a PDF file.

The functions in an arcpy.mapping script closely mimic the actions that would be performed by the map maker using the ArcMap application. Consider the following simple ArcMap workflow:

1. Open the map document located at "C:\GIS\ParcelAtlas_2009.mxd".
2. Find any text elements on the page layout that include "Date: 7/29/2009" and change them to "Date: 7/30/2009".
3. Export the updated map document to a PDF file.

An arcpy.mapping script expresses these steps in the map production workflow as:

```
mapDoc = arcpy.mapping.MapDocument(r"C:\gis\ParcelAtlas_2009.mxd")
for textElement in arcpy.mapping.ListElements(mapDoc, "Text"):
   if textElement.text == "Date: 7/29/2009":
      textElement.text = "Date: 7/30/2009"
arcpy.mapping.ExportToPDF(mapDoc, r"C:\gis\ ParcelAtlas_2009.pdf")
```

The results are shown in *Figure 7*.

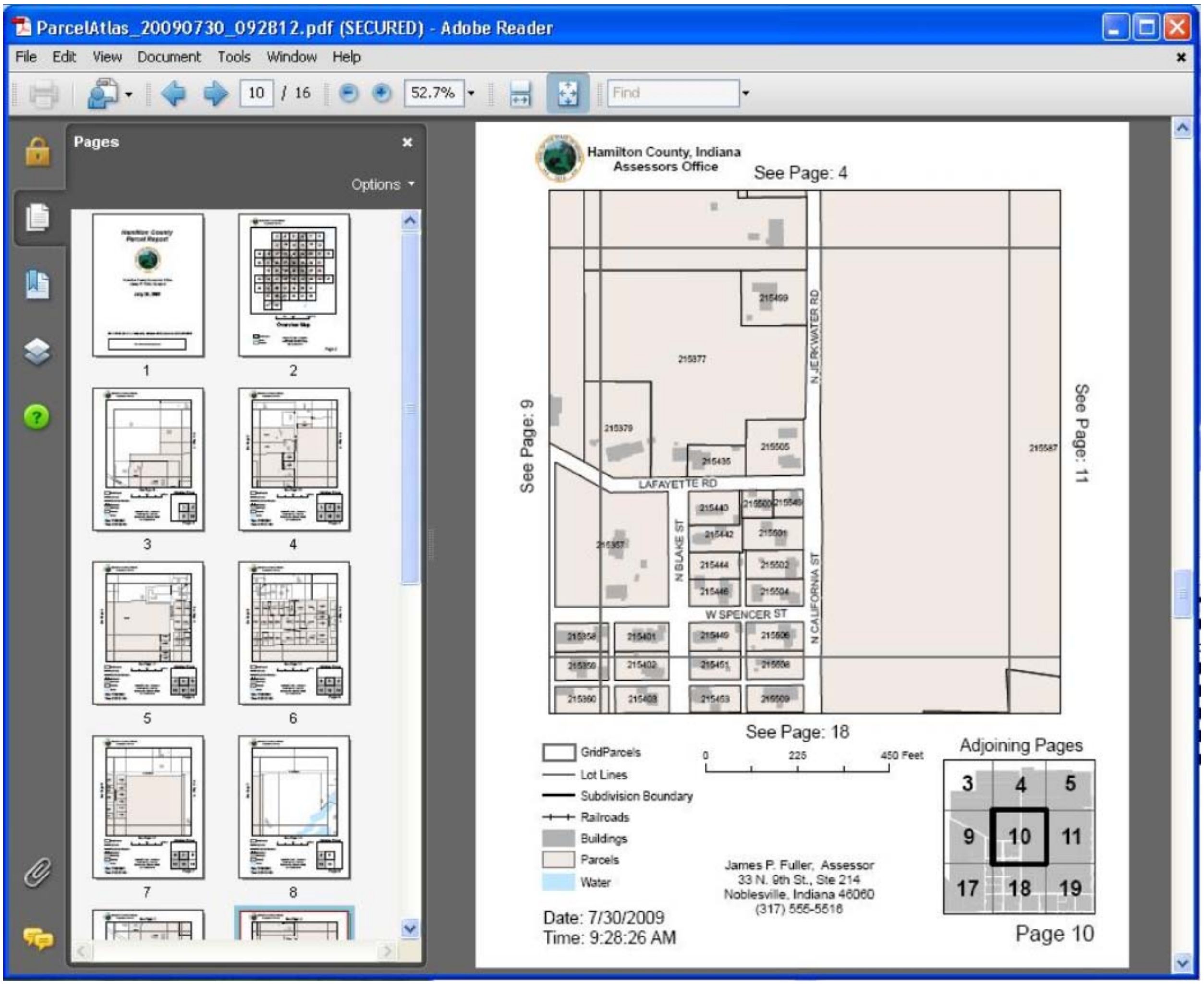

Figure 7. The updated map pages reflect the directions provided in a simple script.

The mapping functions provided in arcpy focus on modifying existing map layers and map elements in the page layout – they are not designed to provide a complete map production system. As with ArcView, the interactive and visual ArcMap environment is used to compile the original maps and page layouts in map documents. This allows mapmakers to not only take advantage of the friendlier interface, they can also use the advanced software tools that support of high end cartography. Though Model Builder and Python scripting, much of the data manipulation and some of the map compilation can be automated through geoprocessing. The data layer and layout element modification capabilities of arcpy.mapping can then be used to automatically create variant map products and to output them. Because Model Builder is user-friendly and the Python scripting language is fairly simple, cartographers need not be software developers to automate production workflows that relate to cartographic data management, map compilation, and printing and exporting (output).

Recall that Robinson and his colleagues advise cartographers that they should compile maps in such a way that the ultimate map construction will be as easy as possible. Being able to successfully use these new GIS software capabilities to automate map production workflows requires careful map construction on the part of the cartographer. The goal is to produce the desired map product so that it contains the map and page elements required to construct variations of the map. In essence, the initial map is “mined” to create variants through instructions contained in a Python script that direct the map production modifications. If the map is constructed to enable this scripting, then arcpy.mapping and ArcMap together provide a much more complete automated map production “system” than cartographers have been offered to date (*Figure 8*).

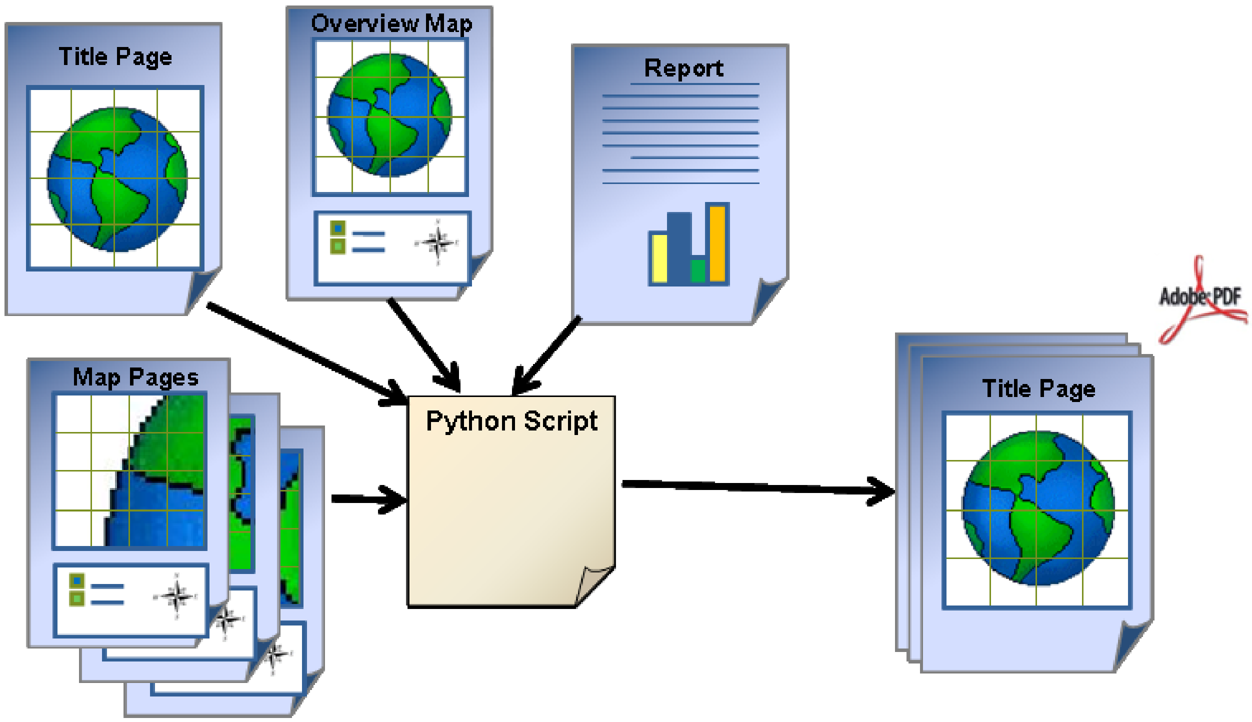

Figure 8. The latest software release provides a more automated map production system to produce multiple maps, along with reports, graphs, and charts that accompany the maps. The system also manages the output of the final product to a digital file or a printing device.

Conclusion

While some level of automation has existed for decades to support map projection workflows, the software limitations or advanced level of training required by the map maker have led to the conclusion that automated map production capabilities have been noticeably deficient. The fully automated map continues to be elusive; however, new software developments offer map makers more opportunities to introduce automation in the map production process than in the past. This, coupled with ease of use and the capability to achieve high quality cartographic results, should make the refined software more attractive to map makers.

We conclude that opportunities do exist for automation of map production workflows to ease and speed the map making process, though they are necessarily different from those that existed previously due to advancements in hardware, software, data, and cartographic knowledge. The paragon of map automation continues to elude us – until the fully automated map becomes a reality and the “Make Map” button appears on a toolbar, the bar high will remain high for software developers, and cartographers no doubt will make sure that it stays up there.

References

Barnes, David. 2003. "Using ArcMap to Enhance Topographic Presentation." *Cartographic Perspectives*, Volume 42, Spring 2002, 5-11.

Brewer, Cynthia A. and Charlie Frye. 2005. "Comparison of GIS and Graphics Software for Advanced Cartographic Symbolization and Labeling: Five GIS Projects." *Proceedings of the International Cartographic Association Conference 2005*, A Coruña, Spain.

Buckley, Aileen R. and Charlie Frye. 2005. "An Information Model for Maps: Toward Cartographic Production from GIS Databases." *Proceedings of the International Cartographic Association Conference 2005*, A Coruña, Spain. Imhof, Eduard. 2007. (English translation first published in 1982). *Cartographic Relief Presentation* (from the original German published in 1965, *Kartographische Gelandedarstellung*). Redlands, CA: ESRI Press, 399 pages, ISBN-10: 1-58948-026-0, ISBN-13: 978-1-58948-026-1.

Buttenfield, Barbara P. and Charlie Frye. 2007. "Re-Thinking Best Practices in Cartographic Data Capture and Data Modeling." *Proceedings of the International Cartographic Congress 2007*, Moscow, Russia.

Eicher, Cory, Barbara Schneider, Markus Bedel and Dieter Neuffer. 2007. "How Not to Cut Yourself on the Bleeding Edge: Experiences from Implementing a Cartographic Production System Based on Commercial GIS Software." *Proceedings of the International Cartographic Congress 2007*, Moscow, Russia.

Hardy, Paul and Dan Lee. 2005. "ArcGIS Tools for Professional Cartography." *Proceedings of the International Cartographic Association Conference 2005*, A Coruña, Spain.

Mark, Robert. 1992. "Multidirectional, oblique-weighted, shaded-relief image of the Island of Hawaii", U.S. Geological Survey Open File Report No. 92-422, http://pubs.usgs.gov/of/1992/of92-422/of92-422.pdf.

Nighbert, J. 2003. "Characterizing Landscapes for Visualization through "Bump Mapping" and Spatial Analyst". Paper presented at the 2003 ESRI International User Conference, http://gis.esri.com/library/userconf/proc03/p0137.pdf.

Nighbert, J. 2006. "The Western Oregon Plan Revision Infrastructure: Design and Implementation." Paper presented at the 2006 ESRI International User Conference.

Nighbert, J. 2007. "Making Noise with ArcGIS." Paper presented at the 2007 ESRI International User Conference, http://gis.esri.com/library/userconf/proc07/papers/abstracts/a1343.html.

Robinson, Arthur H., Joel L. Morrison, Phillip C. Muehrcke, A. Jon Kimerling and Stephen C. Guptill. 1995. *Elements of Cartography*, Fifth Edition. New York, NY: John Wiley & Sons, Inc., 674 pages, ISBN 0-471-55579-7.

Trainor, Timothy and Stephanie Spahlinger. 2008. Complex Automated Map Production Workflow Supporting the 2010 Decennial Census. *Proceedings of AutoCarto2008: The 17th International Research Symposium on Computer-based Cartography*, Shepherdstown, West Virginia, USA. September 8-11, 2008.

Viljoen, David. 2006. "SatValMod." http://nrd:imanrd2@ftp.gis.nrcan.gc.ca/viljoen/downloads/satvalmod/.