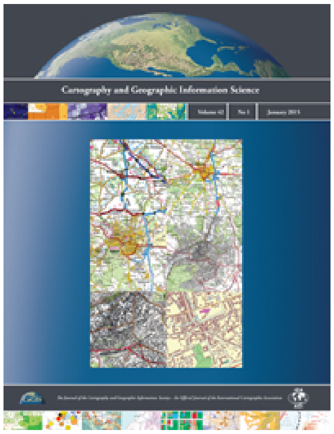

Cartography and Geographic Information Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcag20

A method based on graphic entity for visualizing complex map symbols on the web

Jinqu Zhang[a] & Yunqiang Zhu[b]
[a] Computer School, South China Normal University, Guangzhou 510631, China
[b] State Key Laboratory of Resources & Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China
Published online: 14 Nov 2014.

To cite this article: Jinqu Zhang & Yunqiang Zhu (2015) A method based on graphic entity for visualizing complex map symbols on the web, Cartography and Geographic Information Science, 42:1, 44-53, DOI: 10.1080/15230406.2014.981586

To link to this article: http://dx.doi.org/10.1080/15230406.2014.981586

Cartography and Geographic Information Science, 2015
Vol. 42, No. 1, 44–53, http://dx.doi.org/10.1080/15230406.2014.981586

A method based on graphic entity for visualizing complex map symbols on the web

Jinqu Zhang[a] and Yunqiang Zhu[b]*

[a]*Computer School, South China Normal University, Guangzhou 510631, China;* [b]*State Key Laboratory of Resources & Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China*

(*Received 29 October 2013; accepted 25 September 2014*)

A method based on graphic entity for map symbol visualization in vector-based web mapping is proposed. The method is especially effective for complex geological symbols. It has three parts: graphic entity library, symbol synthesizer, and synthesis specifications. The design process is described as the application of the method to symbols representing point, line, and polygon features in web mapping. The method works with Flash, SVG, or other mapping technologies. We chose Flash to visualize geological disaster information for the Guangdong province in China. The case study demonstrates that this graphic entity-based method can meet the demands for interactive symbolization and complex symbol visualization in Internet web mapping. It produces good results, is fast, and is easy to operate. This makes it especially suitable for web-mapping applications, especially vector-based and hybrid WebGIS applications which display complex symbols.

Keywords: map symbols; interactive symbolization; flash; Internet mapping

Introduction

Map symbols are used to represent objects or geographic phenomena in the real world; they play an increasingly important role in web-mapping applications (Kresse and Danko 2011). With the development of computer and information technology, geographic information system (GIS) has been widely used in various fields. Different fields of applications require different symbol libraries, and existing GIS symbol libraries may not contain the specific symbols used by professionals in various fields. The need to expand GIS symbol libraries, coupled with the existence of popular WebGIS-rich applications and an increasing availability of spatial data, leads to an urgent demand for more efficient symbolization methods for vector-based web mapping.

As in the icon-based visualization technique (Zhang and Pazner 2004, 2007), two types of map symbols – static and dynamic symbols – are used. For both types, the use of complex symbols is unavoidable. A complex symbol can be dynamic or static and cannot be drawn simply by using points, lines, and polygons. Usually, symbols are pre-defined in a desktop GIS and then published using an Internet Map Server. Although this method can display complex symbols, it has problems on the client side when it is used for interactive symbolization. In the past, interactive map symbol visualization was attempted by using SVG technologies (Zaslavsky 2000; Wang, Li, and Yin 2007; Zhang, Li, and Yin 2012). The complex description of symbols in SVG hinders the application and development of SVG in web mapping (Antoniou and Morley 2008).

However, the Internet and the wide use of "rich Internet applications (RIAs)" (O'Rourke 2004) made it possible to apply and change map symbols dynamically according to users' wishes. Map application programming interfaces (APIs) were released by third parties to meet the demand for interactive symbolization at the client. For example, the Environmental Systems Research Institute (ESRI) offers various APIs for web application development on various platforms, including an ArcGIS API for JavaScript, ArcGIS API for Flex, and ArcGIS API for Silverlight. Each of these APIs can retrieve vector-based data; render them on the fly; and enable the user to create highly interactive, visually rich, and expressive web mappings (Schnabel and Hurni 2009). However, visualization of complex symbols remains problematic. GIS software currently on the market provides a rich variety of map symbols. Yet, they often do not meet the needs of professional fields, such as geological disaster mapping. Consequently, many mapmakers have created their own map symbols, including dynamic symbols for animated maps of aggregate data about acquired immune deficiency syndrome incidence (MacEachren and DiBiase 1991), cartographic symbols used for planetary mapping (Nass et al. 2010), and situation symbols in emergency thematic cartography (Wang, Wu, and Gong 2006; Li et al. 2011). These thematic applications above are dependent on the GIS software and lack interactivity in symbol visualization in web applications. Robinson et al. (2013) developed a

*Corresponding author. Email: zhuyq@igsnrr.ac.cn

web-based symbol store and an interactive tool for sharing point symbols, which provided a good solution for sharing map symbols. Auer et al. (2011) developed the symbology and a symbol-scaling method for representing spatial–temporal data in the client, but the method still could not handle complex map symbols.

Currently, Internet-based maps are the major form of spatial information delivery (Peterson 2008). They are especially useful in delivering geological hazard information due to their intuitive, interactive, and dynamic appearance which guides and assists the user in solving complex geospatial analysis problems (Andrienko and Andrienko 1999). Geological disasters such as landslides, ground fissures, and floods have endangered human life for a long time. Many complex symbols may be needed to map different geological disaster points.

In this article, we present a method based on graphic entity for visualizing map symbols. The method was developed to (1) simplify the design process of vector-based complex symbols; (2) promote the sharing and reuse of map symbols, and (3) improve the efficiency of online interactive symbolization.

Method based on graphic entity

There are three main traditional methods for drawing map symbols: (1) using a vector data rendering engine embedded in Internet Explorer (IE); (2) using custom bitmap patterns or icons for point symbols and as a fill pattern for polygons; and (3) using an API provided by HTML5 canvas or a third party, such as ESRI's ArcGIS API for Flex or the Google Maps API. The first method is limited to drawing simple symbols; if complex symbols are attempted, this becomes a very complicated and time-consuming task. Custom bitmaps (method 2) can express complex symbols; however, the symbols will have serrated edges when the user zooms in. With regard to the third method, we were unable to identify a better solution for changing symbols interactively. This section presents the method based on graphic entity and its use for complex symbols' mapping. Different from the graphic primitives described with the coordinates, a graphic entity is a compound of such basic graphic objects as polylines, rectangles, and arcs. It can also be in the form of a recursive set of entities. A graphic entity can be used as a symbol independently or can be used for synthesizing much more complex symbols. The method based on graphic entity has three parts, including a graphic entity library, symbol synthesizer, and symbols specifications (Figure 1).

Graphic entity library

Making complex map symbols is time-consuming. The graphic entity library provides a mechanism for sharing graphic entities. Different from step-by-step drawing, the complex map symbols are created by a combination of graphic entities retrieved from the library. Graphic entities produced using external visual tools are stored in a library, with each entity having a unique number. The entities in the library are shared and can be reused. Different complex symbols can use the same graphic entity, which reduces the need to reproduce the same symbol.

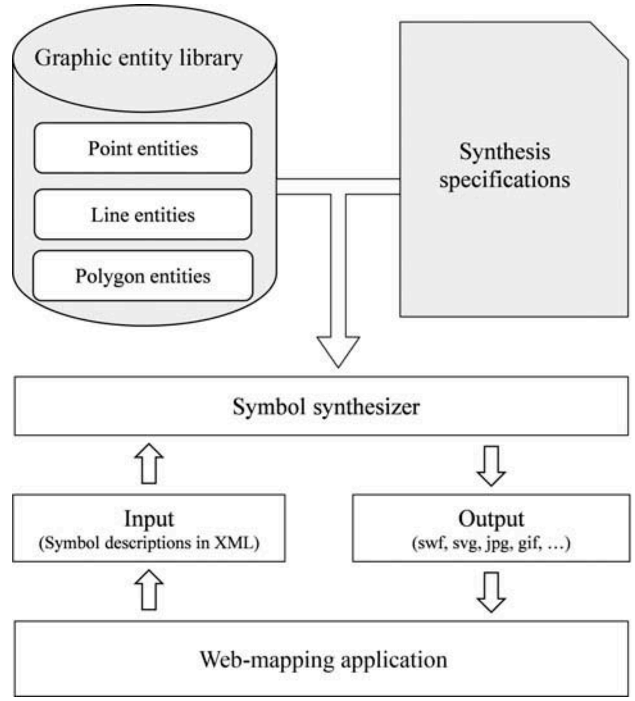

Figure 1. Framework of the method based on graphic entity.

Symbol synthesizer

Symbol synthesizer is a function box with interfaces providing capabilities of combining multiple graphic entities into one compound map symbol (Figure 1). This map symbol may be in Shockwave Flash (swf), Scalable Vector Graphics (svg), jpg, gif, or other format that can be directly used in a web-mapping application. The synthesizer performs the following tasks: (1) retrieval of graphic entities from the library and locating them in the right place and order; (2) rendering graphic entities with appropriate colors; (3) adjusting the display size of each graphic entity; and (4) adding animation effects to dynamic map symbols. The symbol synthesizer receives synthesis specifications from the client user and creates the compound map symbols. Synthesis specifications are defined in the following section.

Synthesis specifications

This part defines the arrangement rules for graphic entities constituting a complex map symbol. The rules are input parameters which are written in XML format with pre-defined XML tags. Some previous studies have pointed out the feature attributes included in a map symbol, such as size, color, and direction (MacEachren and Kraak 2001; Tinghua 1998). Our method is not limited to these symbol elements; it provides a mechanism to extend these elements by using XML tags.

The method based on graphic entity uses a XML tag named state to describe the components and display style of a symbol. There is only one tag for a static map symbol but two or more for a dynamic symbol, with each tag representing a display style. Each state tag contains at least three elements: duration, transparency, and entities. The entity tag is a parent node, with at least one entity tag as a child node. Table 1 lists the common tags with their meanings and optional values.

The following is an example of synthesizing a complex map symbol using synthesis rules. It consists of two graphic entities arranged together according to specified rules. When the symbol synthesizer receives a request, it first creates a blank scene object and then retrieves the graphic entities with ids of 1 and 2. When the entities are in the right place and rendered according to the rules, the symbol synthesizer generates a compound symbol object and returns it to the application. Figure 2 shows the example of a compound symbol generation.

```
<state>
  <duration>0</duration>
  <transparency>1</transparency>
  <entities>
      <entity>
    <id>1</id>
    <scale>1<scale>
    <location>0,0</location>
    <color>0x000000</color>
    <rotation>0</rotation>
      </entity>
        <entity>
    <id>2</id>
    <scale>1<scale>
    <location>0,0</location>
    <color>0x000000</color>
    <rotation>0</rotation>
      </entity>
  </entities>
</state>
```

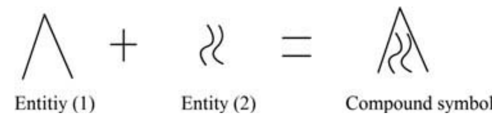

Figure 2. Generation of a compound symbol.

Implementation

The method based on graphic entity can be realized using Flash, SVG, or other technologies. Because we are familiar with Flash, we chose this technology to demonstrate the implementation of the method from symbol design to symbol synthesis.

Graphic entity design and construction

Adobe provides a visual design environment, Flash Professional, for editing FLA (extension of Flash Movie Authoring File) files and conveniently compiles them into swf files. The swf file format is an efficient delivery format for displaying vector graphics on the web. A swf file is composed of a header followed by a number of tags called definition tags and control tags. The display of a swf file frame is a three-stage process:

(1) Objects are defined with definition tags, such as DefineShape and DefineSprite. Each object is given a unique ID, called a character, which is stored in a repository called the dictionary.

Table 1. XML tags and their optional values.

Tag name	Meaning and available values
state	Stands for a complete display status.
duration	Represents the duration of the current state of a dynamic symbol in seconds and zero for the static symbol.
transparency	The transparency of the displayed symbol with value between 0 and 1.
Entities	This tag contains all the graphic entities used for the complex map symbol and they are arranged in the order, with the first entity located at the bottom and the last at the top.
Entity	This is an element contained in the entity tag and it has a series of sub-tags that could be used to describe this entity.
Id	The unique identity number in the graphic entity library.
Scale	Specifies the display size relative to the original size of the entity.
location	Specifies the location of the entity with value of coordinate pair in the form of (x, y) in pixel.
color	Specifies the color of the entity using hex color string.
rotation	Specifies the rotation of the entity in degree.
……	Self-extended properties.

(2) Selected characters are copied from the dictionary and placed on the display list, i.e. the list of characters that will be displayed in the next frame.
(3) Once complete, the contents of the display list are rendered on the screen with ShowFrame.

The most important aspect of this process is that one swf file can be loaded and displayed by another swf file, and the displayed objects can be controlled and selected. According to this characteristic, the designed graphic entities can be stored in a swf file which serves as the graphic entity library. The graphic entities in the library are further retrieved and applied by the Flash-based mapping application.

A graphic entity can be any object or combination of objects, including graphics, text, sound, video, and animation. In our approach, graphic entities can be created by performing the following two steps in Flash Professional:

(1) Choose the Insert menu and select New Symbol. In the Symbol dialog box, choose movieclip and give the movieclip a symbol name. Flash will then create a new movieclip symbol, a graphic entity in the Library, and display the timeline of the new movieclip. In the timeline, one can draw any graphic needed for a map display, just as one would put a graphic into the timeline of the main movie. The names of the entities are made up of the type of the entity and serial number, such as pnt1 for the first point entity, poly1 for the first polygon entity, and line1 for the first line entity. The next graphic entities are named sequentially.
(2) After drawing a graphic entity, right-click on the name of the entity and choose Properties. In the window that then opens, click Advanced to access the options in the advanced zone. Check the "Export for ActionScript" button. This will store the Symbol in a class (whose name is added to the field "Class") and make it possible to use the entity in the ActionScript codes. Flash automatically adds the name of the entity to the Library as the class name. The name can then be modified, although it is usually helpful to have the same name for the class and the object. This name also appears in the "Linkage" column in the Library. Once the "Export for ActionScript" option is enabled and the class name is set, click OK. The class that represents this entity object has been created and can be used in an ActionScript code.

By repeatedly executing the above two steps, the needed graphic entities, including both simple and complex entities, are created in the library.

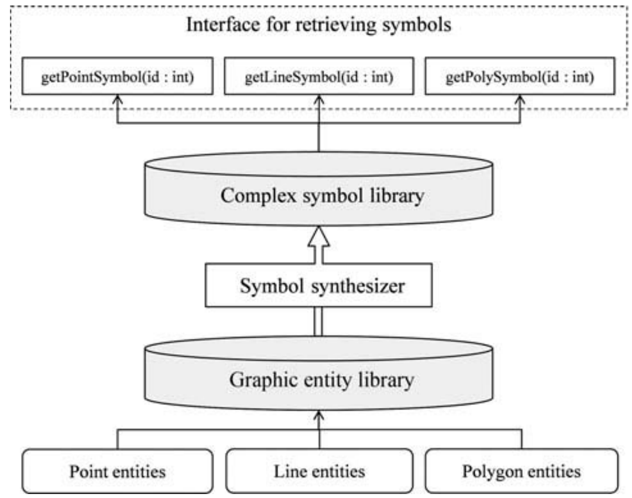

Figure 3. Symbol synthesizer and interfaces.

Symbol synthesizer development

In contrast to step-by-step symbol drawing, graphic entities can easily be designed with the assistance of Flash Professional. A virtual complex symbol library and an interface of retrieving functions facilitate convenient symbol retrieval. Web-mapping applications can directly retrieve complex symbols by calling interface functions (Figure 3).

Methods of applying symbols to a map layer

Symbolization methods are slightly different for point, line, and polygon feature types. The main symbolization processes of our method are shown in Figure 4.

Point symbolization

A point represents a geographic location in space, and point symbol mapping is a fairly simple process. When mapping complex symbols in a Flash-based web-mapping application, the loop program will handle each point until no additional point features remain. In each loop, the program first determines the symbol ID using point record attributes and acquires the symbol instance by sending its ID to the interface function. The map symbol instance can then be directly displayed by adding it to the display list assigned to the location coordinates of the point. The symbol ID parameter can be assigned in advance and stored in an XML configuration file. In interactive symbolization, the symbol ID can be acquired by clicking the symbol list on the web page.

Line symbolization

The symbolization of line features follows a slightly different process than does point feature symbolization. Line

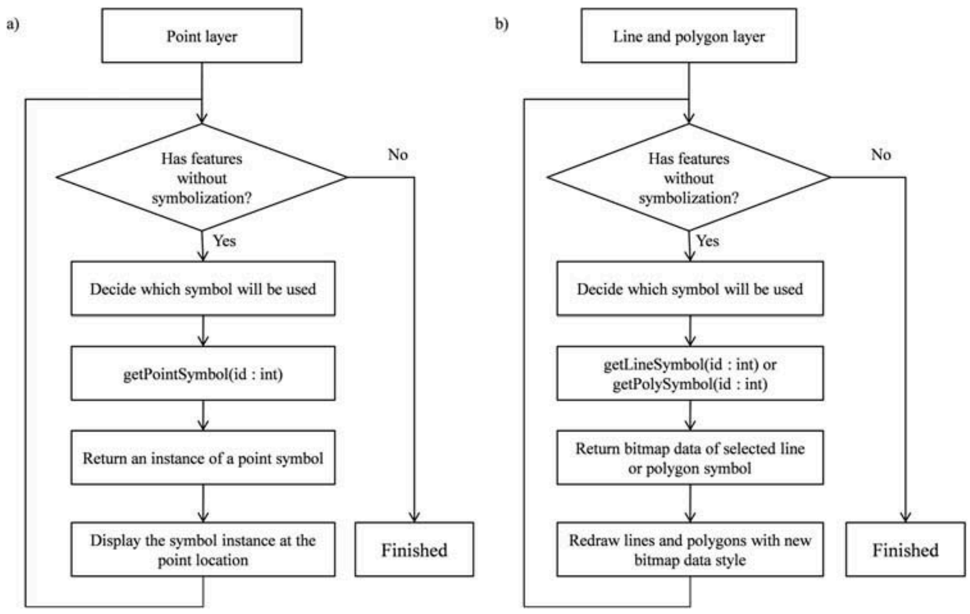

Figure 4. Workflows of map layer symbolization.

symbols are used to draw linear data such as transportation networks, water systems, boundaries, zoning, and other connective networks. In ArcGIS, five line symbol types are available, including simple, cartographic, hash, marker, and picture symbols. Usually, complex line symbols are created by combining more than two line symbols using a combined redrawing method, which slows down the display performance.

With the enhancement of Flash ActionScript 3.0, four functions — lineStyle(), lineShaderStyle(), lineGradientStyle(), and lineBitmapStyle() — can be used to express different line symbols. The use of these functions is described in detail in the online documentation for Adobe (Noble and Anderson 2008). We found the first three functions difficult to use in rendering complex symbols. Therefore, we used the fourth function, lineBitmapStyle(), to draw complex line symbols. All complex line symbols should be created in advance in Flash Professional. When applying a symbol to a line feature, the interface function getLineSymbol(id:int) returns the bitmap data for the selected line symbol. The bitmap data are used as a parameter of lineBitmapStyle(), and then, drawing line with line API functions completes the line symbolization.

Polygon symbolization

The symbolization of a polygon is divided into two parts: border styles and fill patterns. The symbolization of border styles is similar to the symbolization of lines; we now focus on the symbolization of fill patterns. In vector-based WebGIS, symbolization may be applied to the initial drawing and when making changes to the fill pattern. After the data are transmitted, the vector coordinates are manipulated, and various drawing actions are performed to display a polygon. Typically, three steps are used: (1) setting a fill pattern with the functions beginFill(), beginGradientFill(), or beginBitmapFill(); (2) drawing line segments using coordinates that form a closed polygon; and (3) completing the polygon and the application of the fill pattern. This process is time-consuming, especially when changing the fill pattern interactively. In the following section, a mask-based method will be proposed for interactive symbolization of polygons.

Interactive symbolization for a thematic map layer

Interactive symbolization is in high demand, especially by web users interested in thematic mapping. Traditionally, the symbols and the layer are bound together. When a layer is retrieved and displayed, all the symbols on this layer are pre-rendered. Unlike traditional interactive mapping, interactive symbolization has three basic characteristics: (1) Map symbols are independent of the layer and each symbol is a separate entity; (2) Map symbols on a layer can be replaced in whole or in part in real time; (3) The display style of given map symbols can be changed online without sending another request to the server. Interactive symbolizations of point, line, and polygon features differ in their interactive display and implementation methods. Line features are usually symbolized by redrawing. In the following subsections, we focus on interactive symbolization of point and polygon features.

Interactive symbolization of point features

The interactive display of a point symbol display has two aspects: one is replacing the symbol and the other is changing the styles of the current symbol, including size, color, rotation, transparency, location, and other properties. The method based on graphic entity performs both steps quickly and efficiently. In interactive point symbolization, point symbols available in the library are listed in advance. The user can choose one and apply it to the selected features. For existing symbols on a layer, the styles can be changed by modifying the synthesis specifications described above. These specifications are fully implemented on the client.

Mask-based method for polygon symbolization

We propose a simple mask-based approach to enhance the drawing efficiency and interactivity of polygon symbolization. A mask is an object that limits the visibility of another object to the shape of the mask applied to it. A mask can be a filled shape, a type object, an instance of a graphic symbol, or a movie clip. Masks are traditionally used to achieve embellishing effects, such as spotlight effects, and transitions in web animations. In this study, we apply this function to the interactive symbolization of polygons. Mask effects can be achieved manually in Flash Professional and by coding in ActionScript. ActionScript programming is the main technical approach used for the mask-based approach to polygon symbolization in Flash-based mapping.

Figure 5 illustrates the basic steps of the mask-based approach used in polygon symbolization. First, the coordinates of a bounding rectangle are calculated using the getBounds() or getRect() functions in ActionScript. Second, the bounding rectangle is drawn with a fill pattern generated by the synthesizer using specific entities acquired from the symbol library. Third, a mask is applied to the rectangle by selecting the polygon as the mask property of the rectangle. Figure 5(d) and (e) illustrate the process before and after applying the mask.

The rectangle fill patterns are generated from polygon symbols in the library that were previously created in Flash Professional. When applying one symbol to a polygon feature, the interface function getPolySymbol(id:int) returns the bitmap data for the selected polygon symbol. The bitmap data are then used as a parameter of beginBitmapFill(), which completes the rectangle symbolization. By applying a mask to the rectangle, we produce a symbolized polygon.

Geological disaster information mapping in Guangdong province, China

Located in southeast China, the Guangdong Province occupies a land area of 178,000 km^2. Most of the area has a subtropical monsoon climate, with an annual precipitation of approximately 1770 mm per year (Guangdong Provincial Statistics Bureau 2011). The heavy rainfall, recent tectonic movement, ground surface weathering, and human engineering activities have caused many geological disasters in the region, including a severe collapse, landslides, mudslides, and other geological disasters (Lu, Qing, and Fan 2006). Therefore, it is of utmost importance to develop a geological disaster information distribution system for Guangdong Province (Fu, Zhang, and Li 2006). One of the most important tasks is to produce geological symbols and use them quickly and easily. The method based on graphic entity was used to

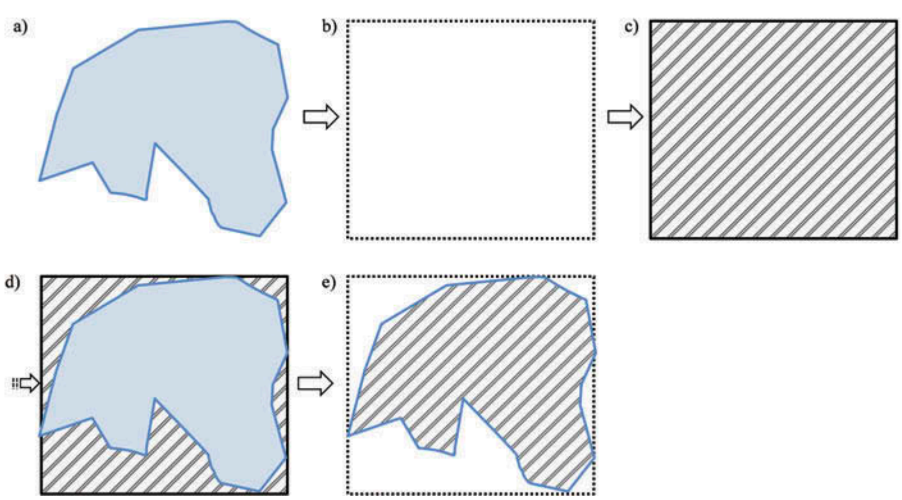

Figure 5. Mask-based approach for polygon symbolization. It has steps from (a) a polygon waiting for symbolization to (e) setting bounding rectangle with polygon as mask, accompanied by (b) getting minimum bounding rectangle, (c) applying the rectangle with fill pattern, and (d) bounding rectangle and polygon without mask.

Table 2. Geological disaster information records.

No.	Region name	Warning level	Disaster type	Longitude	Latitude	Observer	Date and time
1	Pingtang town	3	Landslide	111.37	22.489	Dong Liu	2 December 2012 20:00:00
2	Silun town	3	Landslide	111.315	22.722	Hong Lin	2 December 2012 20:00:00
3	Sihe town	3	Collapse	111.582	22.466	Zu Yuan	2 December 2012 20:00:00
4	Fucheng town	3	Collapse	111.57	22.782	Shan Lin	2 December 2012 20:00:00
…	…	…	…	…	…	…	…

publish the geological disaster information for the province.

The sources of the data were Geological Environment Monitoring Stations in the Guangdong Province. Currently, there are 13 monitoring stations covering the entire province. The geological disaster monitoring data are stored in a two-dimensional table. The data structure is shown in Table 2. The geological disaster information is stored in a database managed using Microsoft SQL Server 2005. These geological disaster point data are mapped on a base map with administrative region boundaries.

Graphic entity design

The map showing geological disasters in Guangdong Province is comprised of complex point symbols indicating collapse, landslides, mudslides, ground subsidence, ground fissures, and land subsidence. These geological symbols had to be designed using symbol definition tools in GIS software. However, it is better to use dynamic symbols for disaster points. Based on the design specifications for geological disaster symbols in China, 88 symbols were selected and made into a Flash swf symbol library. Figure 6 shows some of the

Figure 6. Part of geological map symbols created with Adobe Flash Professional CS3 software.

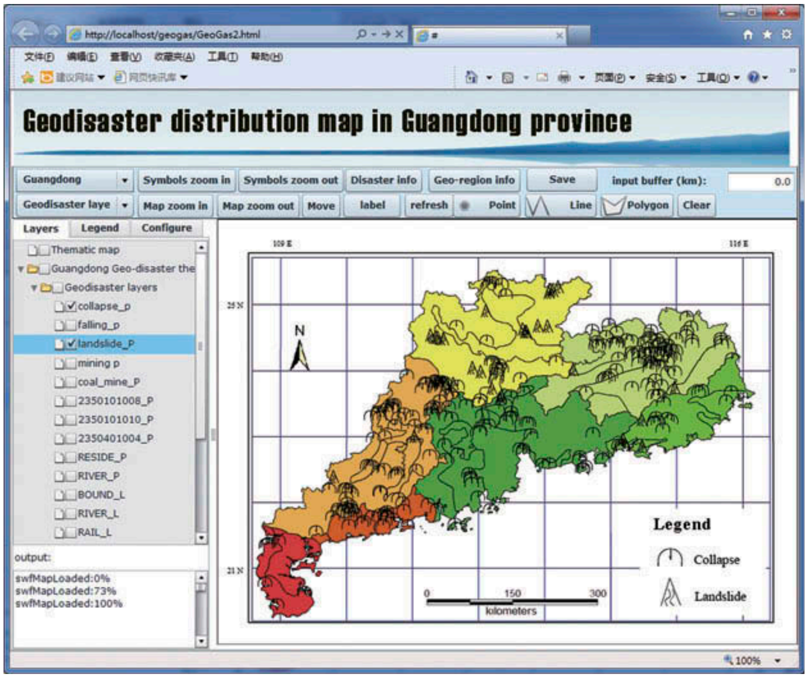

Figure 7. Geological disaster information visualization for Guangdong Province.

geological map symbols created using the method based on graphic entity.

Visualizing the geological disaster information

The geological disaster point symbols created are loaded into a Flex map viewer and stacked on the base map to show the geological disaster distribution information. As described above for point symbol visualization, the geological disaster records are visualized and released to the public using Flash technology. The map viewer is an swf file embedded in the IE. All of the functions in the map viewer are written in ActionScript 3.0 language. When the user navigates to the URL address of the map in IE, the map viewer first loads the administrative region layer from a swf file and then automatically sends a request to the server to retrieve the latest geological disaster records from the database using an Active Server Page program. The map viewer will then extract the needed, type-specific geological disaster symbols from the symbol library and visualizes the symbols in the appropriate locations based on the location fields stored in the record. Figure 7 shows web-based geological disaster information visualization in the Guangdong Province. Besides zooming the map, the user can zoom selected point symbols and interactively change fill patterns for the polygons in the application.

Discussion and conclusion

The design of disaster symbols and their visualization are the two most important challenges for creating a geological disaster map. The traditional methods are, for various reasons, unable to provide interactive visualization of dynamic points such as incidences of geological disasters. One of the most egregious shortcomings is that the symbols must be redrawn point by point, which reduces drawing efficiency. Given these shortcomings and restrictions, a method based on graphic entities was used to publish geological disaster information.

The method was successfully applied to mapping geological disaster information in Guangdong Province in China. In the study, Flash Professional was used to visualize complex geological disaster symbols. Through a unified naming convention and programming processing, these complex geological disaster entities can be retrieved and integrated into one compound symbol by a symbol synthesizer. The complex compound symbols created are then applied to a map layer. The case study shows that our method is fast and user friendly and that it can improve the efficiency of online interactive symbolization. The method based on graphic entity meets the demands of mapping geological disaster information. Complex map symbols and interactive symbolization are easily implemented. Although in our study we use Flash technology, the method can be implemented with other technologies too.

To sum up, the method based on graphic entity described here has been shown to be of assistance in designing complex symbols and improving the symbolization process for vector-based Internet mapping. It benefits the symbolization process in at least three aspects:

(1) It simplifies the complexity of the production process for complex map symbols. The designed graphic entities can be easily integrated into the application, which in turn may be used for rendering map layers in real time.
(2) It supports both static and dynamic symbols applied to a map layer in the application. Both static and dynamic symbols can be used to symbolize a map layer. The difference is that a static map symbol has only one display state, whereas a dynamic symbol has at least two states forming a tween animation.
(3) It provides a mechanism for the sharing and reuse of map symbols.
(4) When using the method based on graphic entity, users will usually be provided with a list of graphic entities to work from. Note also that while our method used Flash Professional, other technologies can learn from our approach and develop similar applications.

With the development of WebGIS applications, demand for custom, dynamic map symbols will increase. However, tools for Internet-based map symbol design and interactive symbolization are still scarce. This article presents a map symbol design and application process for point, line, and polygon features. Our research shows that the method based on graphic entity presented here is more efficient compared with the traditional methods. Mechanisms and methods for sharing map symbols based on graphic entities will be investigated in the future research.

Acknowledgements
This study was financially supported by the Natural Science Foundation of Guangdong Province, China [grant number S2013010014097]; the Research Plan of LREIS [grant number O88RA900KA]; and the Key Project for the Strategic Science Plan in IGSNRR, CAS [grant number 2012ZD010].

References
Andrienko, G. L., and N. V. Andrienko. 1999. “Interactive Maps for Visual Data Exploration.” *International Journal of Geographical Information Science* 13 (4): 355–374. doi:10.1080/136588199241247.

Antoniou, V., and J. Morley. 2008. “Web Mapping and Webgis: Do We Actually Need to Use SVG?” In *SVG Open 2008: 6th International Conference on Scalable Vector Graphics*, Nuremberg, August 26–28, 1–14. http://www.svgopen.org/2008/papers/82-Web_Mapping_and_WebGIS_do_we_actually_need_to_use_SVG/

Auer, T., A. M. MacEachren, C. McCabe, S. Pezanowski, and M. Stryker. 2011. “HerbariaViz: A Web-Based Client–Server Interface for Mapping and Exploring Flora Observation Data.” *Ecological Informatics* 6 (2): 93–110. doi:10.1016/j.ecoinf.2010.09.001.

Fu, C., X. Zhang, and Z. Li 2006. “The Design of Framework of Geo-Hazard Early-Warning in Guangdong Province.” [In Chinese.] *The Chinese Journal of Geological Hazard and Control* 17 (1), 51–55.

Guangdong Provincial Statistics Bureau. 2011. *Guangdong Statistical Yearbook*. Guangzhou: Guangdong Yearbook Press.

Kresse, W., and D. M. Danko. 2011. *Springer Handbook of Geographic Information*. Berlin: Springer.

Li, L., J. Qiu, Z. Cheng, and S. Chen. 2011. “Modeling and Implementation of Situation Symbols in Emergency Thematic Cartography.” 2011 19th International Conference on Geoinformatics. doi:10.1109/GeoInformatics.2011.5980683.

Lu, X., Z. Qing, and T. Fan. 2006. “Study on Prediction of Geological Disaster Distribution in Guangdong Province.” *Chinese Journal of Rock Mechanics and Engineering* 25 (Suppl. 2): 3405–3411.

MacEachren, A. M., and D. W. DiBiase. 1991. “Animated Maps of Aggregate Data: Conceptual and Practical Problems.” *Cartography and Geographic Information Science* 18 (4): 221–229. doi:10.1559/152304091783786790.

MacEachren, A. M., and M.-J. Kraak. 2001. “Research Challenges in Geovisualization.” *Cartography and Geographic Information Science* 28 (1): 3–12. doi:10.1559/152304001782173970.

Nass, A., S. van Gasselt, R. Jaumann, and H. Asche. 2010. Implementation of Cartographic Symbols for Planetary Mapping in Geographic Information Systems. *Planetary and Space Science*. doi:10.1016/j.pss.2010.08.022.

Noble, J., and T. Anderson. 2008. “Flex 3 cookbook [online].” Accessed January 31, 2014. http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/flex/pdfs/ora_flex3_ckbk_ch21.pdf

O’Rourke, C. 2004. “A Look at Rich Internet Applications [Online].” *Oracle Magazine*, 7, 1–4. Accessed January 31, 2014. http://wolfpaulus.com/wp-content/uploads/2010/06/richinetapp.pdf

Peterson, M. P. 2008. *International Perspectives on Maps and the Internet*. Berlin: Springer.

Robinson, A. C., S. Pezanowski, S. Troedson, R. Bianchetti, J. Blanford, J. Stevens, E. Guidero, R. E. Roth, and A. M. MacEachren. 2013. “Symbol Store: Sharing Map Symbols for Emergency Management.” *Cartography and Geographic Information Science* 40 (5). doi:10.1080/15230406.2013.803833.

Schnabel, O., and L. Hurni 2009. “Cartographic Web Applications – Developments and Trends.” Proceedings of the 24th International Cartography Conference, Santiago, November 15–21.

Tinghua, A. I. 1998. “Dynamic Symbol and Dynamic Map.” [In Chinese.] *Journal of Wuhan Technical University of Surveying and Mapping* 23 (1), 47–51.

Wang, C., L. Li, and Z. Yin. 2007. “The Area Symbols Design and Implementation Based on SVG.” [In Chinese.] *Science of Surveying and Mapping* 32 (3), 168–145.

Wang, L. Y., X. Wu, and J. Gong. 2006. "Modeling and Application of Situation Symbol in GIS." *Journal of Geomatics* 31 (2): 4–6.

Zaslavsky, I. 2000. "A New Technology for Interactive Online Mapping with Vector Markup and XML." *Cartographic Perspectives* 37: 12–24.

Zhang, H. S., Y. Li, and Z. P. Yin. 2012. "An Approach on Interactive Allocated Map Symbol System for SVG Webgis." [In Chinese.] *Science of Surveying and Mapping* 37 (6), 130–132.

Zhang, X., and M. Pazner. 2004. "The Icon Image Map Technique for Multivariate Geospatial Data Visualization: Approach and Software System." *Cartography and Geographic Information Science* 31 (1): 29–41. doi:10.1559/152304004773112758.

Zhang, X., and M. Pazner 2007. "An Icon-Based Co-Geovisualization Technique for Multivariate Geospatial Data." In *Proceedings of the7th International Workshop on Geographical Information System Conference*, 310–315. Beijing: IGSNRR.