The Design and Implementation of a Reusable Viewer Component

Hong Gab Kim, Young Jae Lim, Kyung Ok Kim

Spatial Information Technology Center, Computer & Software Laboratory, ETRI
161 Kajong-Dong, Yusong-Gu, Taejon 305-350, KOREA
Phone : +82-42-860-5327
Fax : +82-42-860-4844
E-mail : { gabbi, yjlim, kokim }@etri.re.kr

Abstract

This article outlines the capabilities of a viewer component called GridViewer, and proves its reusability. GridViewer was designed for the construction of the image display part of GIS or remote sensing application software, and consequently it is particularly straightforward to closely couple GridViewer with access to very large images. Displaying is performed through pyramid structure, which enables to treat very large dataset up to several gigabytes in size under the limited capability of PC. GridViewer is free from responsibility to handle various formats of raster data files by taking grid coverage, which is designed by OGC to promote interoperability between implementations done by data vendors and software vendors providing analysis and grid processing implementations. GridViewer differs from other such viewer by allowing for clients to extend its function and capability by using small set of methods originally implemented in it. We show its reusability and expandability by applying it in developing application programs performing various functions not supported originally by the GridViewer COM component.

1. Introduction

A viewer, image display part of image processing software including GIS and remote sensing software, fundamentally visualizes raster datasets on the screen. When a viewer has only responsibility to display images, it is easy to develop and maintain it. Recently, a viewer plays not only display part but also communication part between a user and image processing programs. For example, many operations, such as creating region of interest, displaying and editing vector data, adding annotation, and collecting ground control points, including basic functions such as zooming and panning, are accomplished on the viewer. The more functions executed on the viewer, the less reusability of the viewer, because the viewer becomes complicated and depend on the other part associated with the added functions.

The goals of GridViewer are to provide a reusable viewer component for development of various applications used in GIS and remote sensing field. GridViewer can display very large datasets up to several gigabytes in size and is free from various formats of raster data files by taking advantage of grid coverage[1][2].

2. Architecture of GridViewer

GridViewer contains five managers in it as shown in figure 1. Layer manager plays an important part in managing input grid coverages and accessing to raster datasets through grid coverages. Draw manger displays raster datasets according to the zoom ratio and displaying region and tool manager handles user's action to the viewer. Coordinate conversion manager converts coordinate unit among screen, grid(image) and georeferenced coordinates. Cache manager enables fast display by reducing frequency of access to the large datasets.

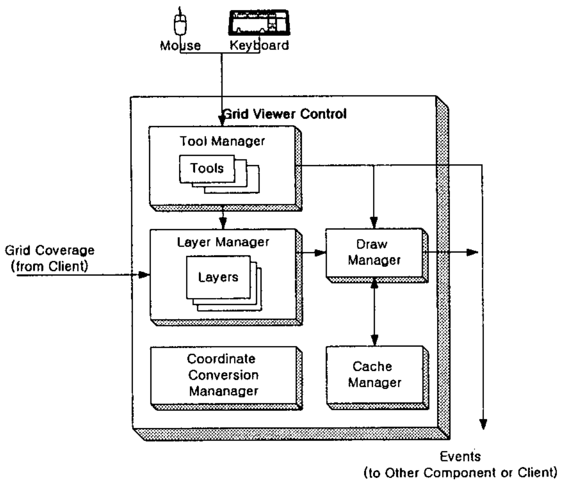
Mouse
Keyboard
Grid Viewer Control
Tool Manager
Tools
Layer Manager
Layers
Draw Manager
Coordinate Conversion Mananager
Cache Manager
Grid Coverage (from Client)
Events (to Other Component or Client)

Figure 1. Architecture of GridViewer

2.1 Layer Manager

Layer manager plays a fundamental part to manage grid coverages fed into GridViewer and access raster datasets through them when draw manager requests a block of raster dataset to draw. Only layer manager has right to access and read datasets and all modification of raster dataset is performed out of GridViewer. This restriction facilitates persistent synchronization of raster data.

2.2 Cache Manager

It is impossible to load whole raster dataset on the memory and display it, because the raster dataset can be very large up to several gigabytes in size. GridViewer only fetches a block of raster dataset displayed on the window from the whole raster dataset. GridViewer makes image tiles from a fetched data block and draw them on the window. When image tiles no more need to be displayed, cache manager puts them into the cache instead of destroying them instantly. Later, draw manager does not request a block of raster dataset to the layer manager but uses the cached tiles, if the image tiles exist in the cache. With reducing frequency of access to the datasets, GridViewer improves panning and zooming speed. The cache replacement strategy used in cache manager is on the basis of aging concept that the oldest image tile must be replaced when local cache is full. Cache manager increases the ages of all image tiles according to their proximity-distances. The aging of all cached image tile increases in proportion to the distance between each image tile and region displayed on the window[3][4].

2.3 Tool Manager

Tool manager activates or deactivates the tools, such as zooming tool and panning tool, contained in GridViwer. The activated tool executes their tasks according to the user's action occurring on the viewer. When all tools in the GridViewer are deactivated, user's actions to the viewer, for example, moving a mouse and clicking the left button of a mouse, are notified to the client by firing events. The other tools that are not supported by GridViewer can be embodied in the client side with these events.

2.4 Draw Manager

Draw manager calculates the region to be displayed on the window, requests blocks of raster dataset to layer manager or cache manager and displays them on the window. Draw manager fires an event to the client whenever it finishes drawing fetched image tiles. The client can embody virtual layers by using the event, on which client can draw anything for itself.

2.5 Coordinate Conversion Manager

Coordinate conversion manager can perform various coordinate unit conversion among window, image and georeferenced coordinates.

3. Implementation and Applications

The developed GridViewer COM component is available under rapid application development environment, Visual Basic, Power Builder, JavaScript and even script language such as JavaScript and VBScript, because it supports dual interface. This component has small set of methods. The methods not supported by it can be embodied by using its original methods in the client or other components working together with it.

3.1 Magnifier Window

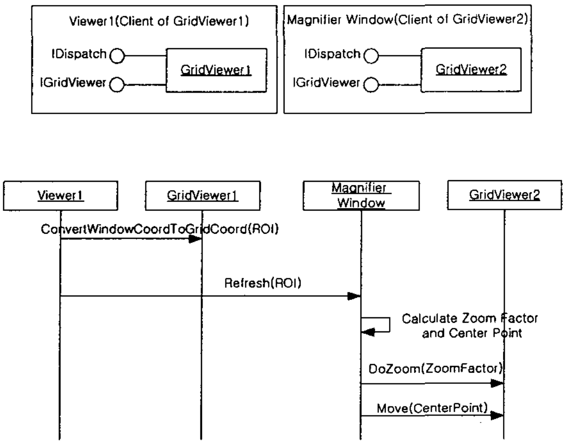

Figure 2. Structure and message diagram for a magnifier window

The magnifier window is a viewer to display the specified region on the original viewer at different zoom ratio. GridViewer does not have this function, but it can be developed easily with GridViewer component. As shown in figure 3, if the region to be magnified is changed, viewer1(client of gridviewer1) calculates the image coordinate of the region by coordinate conversion methods of gridviewer1 and passes it to the magnifier window(client of gridviewer2). The magnifier window displays the requested region by calling the zooming and panning methods of gridviewer2. Figure 3 shows the developed magnifier window.

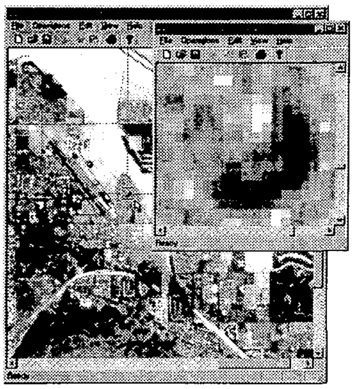

Figure 3. A sample of magnifier window

3.2 Vector Overlay

Vector overlay, to draw vector data on an image, is can be easily implemented by using coordinate conversion method of GridViewer. GridViewer fires an event to the client when it finishes drawing process. The client converts coordinate unit of vector data to window coordinate unit and draws them on the device context given as one of parameters. Figure 4 shows a sample client performing vector overlay. Any other functions can be embodied easily in the client in similar manner as explained above. Figure 5 shows an example client of GridViewer. It can perform additional operations such as creating region of interest and collecting ground control points.

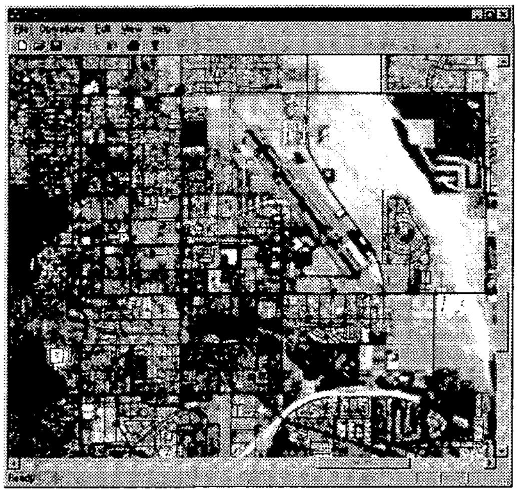

Figure 4. Vector overlay

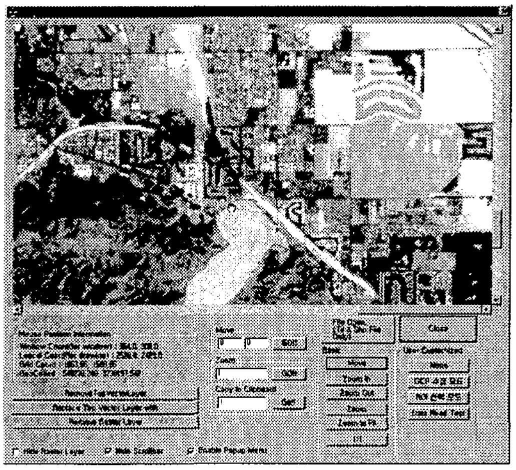

Figure 5. A sample client with GridViewer

5. Conclusion

GridViewer, a viewer component for remote sensing and GIS application software is proposed and developed. Displaying is performed through pyramid structure, which enables to treat very large dataset up to several gigabytes in size under the limited capability of PC. GridViewer is free from responsibility to handle various formats of raster data files by taking grid coverage. It has a small set of mandatory methods, but the functions not supported by them can be added by using its original methods in the client program or other components working together with it. It is easier to maintain the viewer module because the viewer does not have to be changed in order to add other functions. The developed viewer component has been applied to various applications, such as a magnifier view, vector overlay, and creating region of interest, so as to prove its reusability and extensibility.

6. Reference

[1] OpenGIS Consortium Technical Committee, *The OpenGIS Guide,* 1998

[2] OpenGIS Consortium, *OpenGIS Implementation Specification: Grid Coverage, Revision 1.0,* 2001

[3] Dae-Soo Cho, Kwang-Soo Kim, and Jong-Hun Lee, Design of Client-side Caching and Region Prefetching Algorithms for Internet GIS, *Proceeding of International Symposium on Remote Sensing 2001,* pp.100-103, October, 2001

[3] Owen, M.J, Lui, A.K, Lo, E.H.S, Grigg, M.W, The design and implementation of a progressive on-demand image dissemination system for very large images, *Computer Science Conference 2001. ACSC2001. Proceeding 24th Australia,* pp.148-155, 2001