http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 4 (4): 469-475, 2005
ISSN 1812-5638

Map Viewer: A New Add-on Tool for Geographic Information System

K.B. Chari and S.A. Abbasi
Centre for Pollution Control and Energy Technology, Pondicherry University,
Kalpet, Pondicherry, 605014, India

Abstract: A new GIS add-on tool, MapViewer, has been developed to bestow features such as enhanced graphic user interface, 3-D rotation of map windows, resizing of the map windows, re-positioning of the windows and changing their borders, etc, that are currently not available with GIS software. MapViewer also incorporates more powerful window information features and greater flexibility in handling the GIS maps. MapViewer has been coded in MapBasic 5.5 and Visual Basic 6.0. It executes its commands through Visual Basic, using the 'integrated mapping' feature of the anchor software MapInfo Professional. MapViewer has been subjected to extensive multi-user testing for over a year. The problems initially faced by the users were ironed out and since then the software has stood the test of time, establishing its ruggedness and durability.

Key words: Geographic Information System, software, MapViewer, GIS package

INTRODUCTION

Within a short span of time, GIS (Geographic Information Systems), has metamorphed from an expensive and avante garde technique to a relatively inexpensive and commonly used information management tool. If a professional cartographer-who was the original patron of GIS-uses this tool all the time, now so do very many others belonging to as diverse professions as paleontology and pizza selling.

As the number and the diversity of GIS users keeps increasing it has become necessary to keep pace with it by enhancing the power as well as the user-friendliness of GIS.

MapInfo and GeoMedia are two of the most widely used GIS packages. After using their successive versions over several years it became clear that one of their main functions-handling of the sub-maps or map layers can be made much more functionally capable as well as convenient. A few more additions/modifications were also envisaged to further enhance the utility value of these packages. These add-ons have been developed for MapInfo Professional and the resulting module named as MapViewer (Fig. 1). With similar logic, the capability of GeoMedia and other GIS packages can also be enhanced.

Present features of MapInfo and the advancements: The basic advancement made by us is in one of the core facilities of MapInfo-the way the sub-maps or the map layers (which together constitute the composite map of the area under study) are opened and controlled. In this facility, sets of map layers can be viewed either by using the cascade option (Fig. 2a), or the tile option. These features are intended to enable the user to add or delete sub-maps, re-order them, or view one or more map layers arranged side by side under the tile option. The problems one encounters when working in this environment are:

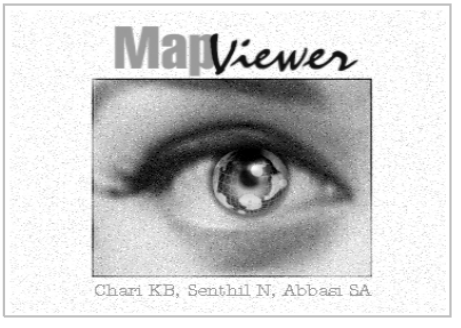

Fig. 1: Splash screen of MapViewer

Difficulty in picking one or the other map layers from the stack of map layers: In MapInfo, the map layers when opened, are stacked one over the other in the active map window. It is not possible to distinguish one map layer from another. As a typical GIS involves working with a large number of individual map layers, this turns out to be a handicap. Moreover, if one wishes to look at each map layer in a separate window, one would need to open the

Corresponding Author: S.A. Abbasi, Professor, Centre for Pollution Control and Energy Technology, Pondicherry University, Kalpet, Pondicherry, 605014, India Tel: +91 413 2655262 or 2655991/362

map layers in a single window and clone this window and remove all the map layers except the one which is needed (Fig. 2a). This is quite laborious and time-consuming.

No provision for rotation of map layers: In MapInfo, the map layers open in a pre-set, fixed manner. It is not possible to rotate the stack of map layers for enhancing the visibility of the constituent layers or to simultaneously reposition, rotate, add or delete more than one map layer.

Other facets which limit user-friendliness: While putting MapInfo to extensive use one feels handicapped by the limited information currently provided by the info tool, and the rather thick window frames limit the visibility of map layers.

In the present GIS add-on tool MapViewer, above mentioned shortcomings have been removed (Fig. 2b). In addition, several features were modified, which would further enhance the user-friendliness of the GIS software (Table 1).

SYSTEM DESIGN

MapViewer begins with a MDI (multiple document interface); after the splash screen (Fig. 1), the user gets right into the thick of MapViewer.

The main menus of the MapViewer are:

- File
- Opened tables
- Options
- Layout
- Windows
- Help

The file menu performs the common operations: opening of the tables, saving, printing and exiting from the software. Using the windows menu the user can then perform other operations on the opened windows: moving, sizing and rotating.

The opened tables menu contains the list of tables that are opened. The table that is active will be checked against the file name. The user can also bring a table to the top (i.e., make it active) by checking that particular table's name in the opened tables menu.

The options menu enables rearranging the order of the tables and deleting them. The layout menu enables the user to compose maps for printing or for exporting them as bitmaps and other raster formats. The help menu displays the detailed help features of the software and the about MapViewer feature.

Toolbar: The toolbar contains several new features in addition to the features present in the menus. Options such as open, save and print are already available in the menus, but still they are also kept in the toolbar for quick access.

The tools-select, zoom-In, zoom-out, grabber, info, label, ruler and text-which are available in MapInfo, have been retained but have been modified in the MapViewer to improve the latter's efficiency.

Table 1: The features that were enhanced newly introduced in MapViewer

	Description	MapInfo (MI)	MapViewer (MV)	Comments
Enhanced interface	The map layers look like typical paper maps; the map windows doesn't have the minimize, maximize and close buttons. This means more map area to view and work	☒	✓	
3-D rotation of map windows	Ability to rotate the map windows along x, y and z axis	☒	✓	
Resizing of the windows	Ability to resize a single/all the windows at once	☒	✓	
Windows positioning	Changing the position of a single window/all the windows at once	☒	✓	
Order of the windows	Change the order of windows; move the windows back and forth in a series	☒	✓	
Window borders	Ability to manipulate the window borders; add a border of desired color and width	☒	✓	
Deleting windows	Delete windows using the menu/button options	☒	✓	
Info tool	The window info tool also indicates the file name and location of the active map layer apart from the attached data	▫	✓	MI: one can only see the attribute data
Opening of the map layers default	Each new map opens in a new map window	▫	▫	MI: there is an option to open map layers in each window while opening MV: all the map layers open by in a separate map layer; to open map layers in the same window it can be done by 'adding the layers' using the layer control box
Layout	Positioning of the windows in the layout	☒	✓	

☒ not available, ✓ available, ▫ limited

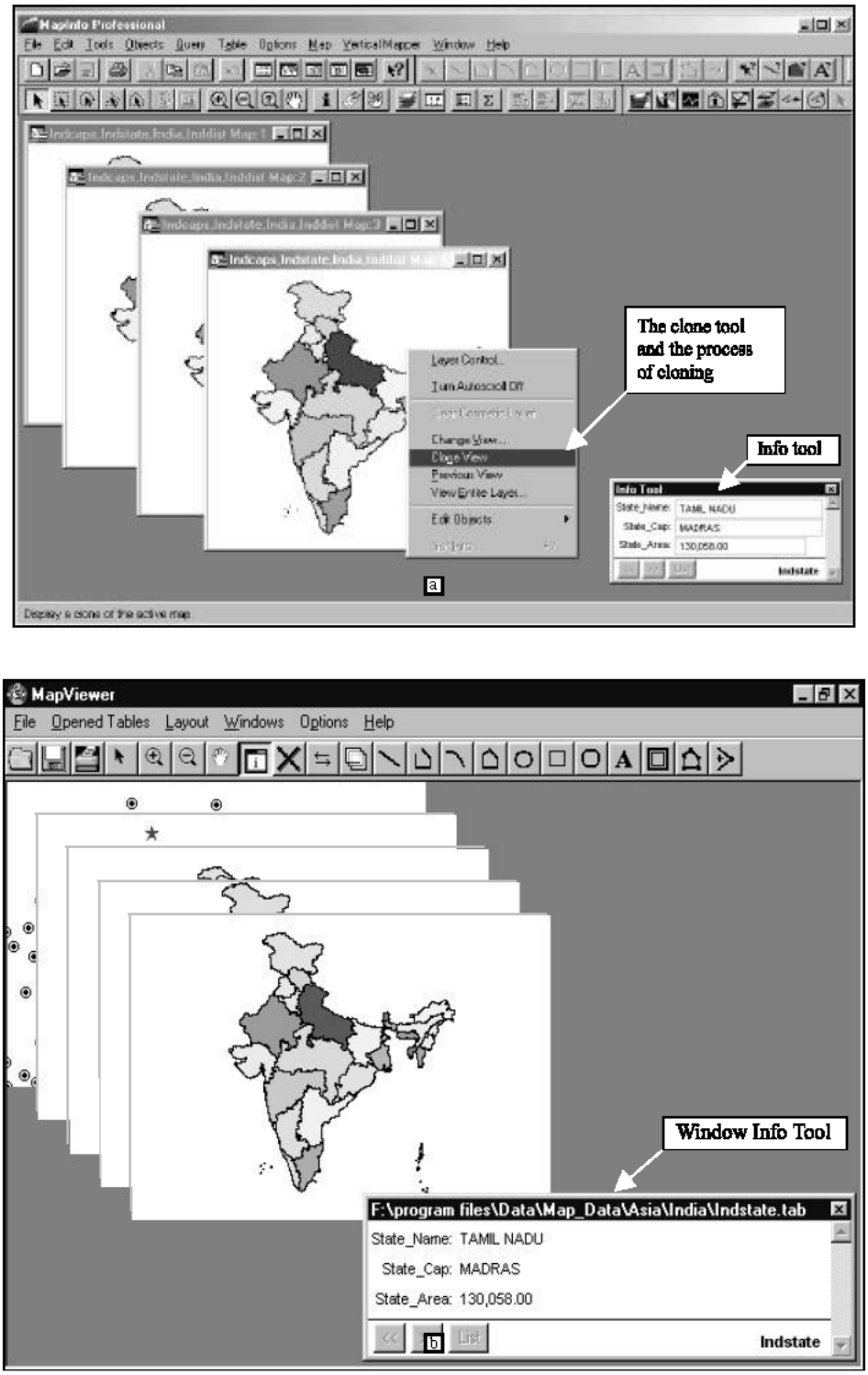

Fig. 2: The typical environment of MapInfo (above) and MapViewer (below). Note that MapViewer has clearer representation of the maps and a more expressive window Info Tool

Some of the new features available in the toolbar of MapViewer are deleting the active window, rearranging the windows with their current size and arranging the windows in a cascade form or as tiles.

Map window: Map window is one of the key features of MapViewer. Here, the specific frameless MapInfo tables are placed over a Visual Basic form. The system is designed in such a way that it gives more flexibility to the user for manipulating the tables.

In MapInfo all the map windows have caption bars and frames like any other normal window. The space taken by the tool bar and frame decrease the ease of visualization of the map layers. In MapViewer, the caption bar was removed and the window frame has been trimmed; these twin modifications facilitate much better visualization of the map layers than the conventional map window of MapInfo.

In order to have a better control on the map windows, in MapViewer the tables are arranged in a cascading fashion as a group of 12 tables, each group called as window group.

Each window group in MapViewer can be moved, sized and rotated at will, that too simultaneously (Fig. 3). The routine cascade and tile operations can also be performed on these window groups. As seen earlier, the names of all the tables, which are opened, are added to the opened tables menu. The user can activate any window by selecting the corresponding table name from this menu.

INPUT DESIGN

The inputs to MapViewer, sent as messages given to the software through the selection of various menus and toolbars, are processed in two ways:

- Through MapBasic statements
- Through Visual Basic statements

In MapViewer the actions that manipulate the MapInfo tables are taken using MapBasic statements, which are sent to MapInfo through Visual Basic. The actions that manipulate the windows are taken directly through the Visual Basic statements.

Starting MapInfo: To start a unique instance of MapInfo, it needs to call Visual Basic's CreateObject function and assign the return value to a Visual Basic Object variable. For example, if the Object variable is named as "mapinfo" then the following statement launches MapInfo:

Set mapinfo=CreateObject("MapInfo.Application")

To attach to a previously running instance of MapInfo which was not launched by a CreateObject call, Visual Basic's GetObject function has to be used.

Set mapinfo=GetObject("MapInfo.Application")

Sending commands to MapInfo: After launching MapInfo, text strings that represent MapBasic statements are constructed. For example, if MapInfo has to execute a MapBasic open table statement, the following string has to be constructed (within Visual Basic):

msg = "Open Table ""STATES.TAB"" Interactive "

This string can be sent to MapInfo by using the Do message as follows:

mapinfo.Do msg

OUTPUT DESIGN

The output of MapViewer is viewed in the Visual Basic MDI form. The MapInfo tables are placed in the child form of the MDI.

Re-parenting MapInfo windows: After launching MapInfo, one has to use the MapBasic statement Set Application Window so that MapInfo dialog boxes and error messages are owned by the client program.

msg = "Set Application Window" and FormName.hWnd

mapinfo.Do msg

Should it be required to integrate a MapInfo window into the Visual Basic application, send MapInfo a Set Next Document statement, followed by the MapBasic statement that creates the window. For example, the following commands create a MapInfo map window as a child window of the Visual Basic program.

msg = "Set Next Document Parent" and MapFrame.hWnd and "Style 1"

mapinfo.Do msg

msg = "Map From States"

mapinfo.Do msg

The Set Next Document statement lets one to 'reparent' document windows. Within the Set Next Document, the hWnd(handle) of a control in Visual Basic has to be specified. The Set Next Document statement includes a Style clause, which controls the type of window one would create.

Manipulation of map windows: The 2-D and 3-D functionalities involve the manipulation of Visual Basic forms. If the user manipulates the windows with the tables in them, then it will be a time consuming process to move or resize the windows. So, a more efficient way is followed in MapViewer for the manipulation of map windows. Whenever the user wants to perform any 2-D or 3-D

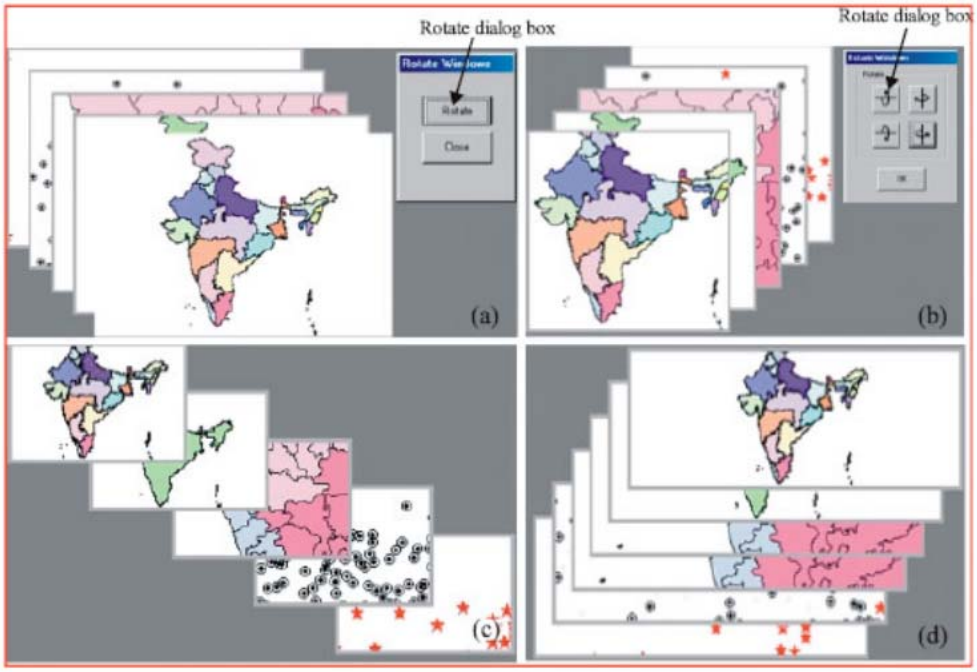

Fig. 3: Different options of rotating the window groups in MapViewer using the tool 'Rotate'; (a) the actual position of the window groups (b) the window groups after rotating leftwards (c) after rotating downwards (d) after rotating rightwards

functionalities on the map windows, all the windows are unloaded after storing the size and position of each window and also the names of the corresponding tables in the respective order.

The user can resize, move, or rotate the windows as desired. These changes are initially reflected in the 'template' window frames; on selecting 'OK', the map tables are then reloaded into the respective window frames.

Optimization of info tool's output: In MapInfo, the info tool displays the attribute data about the object which has been selected using the cursor, if there is some attribute data associated with that object. But it does not give the details about the location of the table.

MapViewer has an info tool, which has been modified to give more info than MapInfo's. This info tool has been mainly modified to give details about the table to which the information is associated wit\h. In this modified info tool, the caption bar contains full path of the table with which the information is associated, including the table name.

CODE DESIGN

In the case of MapViewer the MapBasic statements are executed through Visual Basic. The code has been segregated into different modules so as to achieve easy handling of the coding and better efficiency of the program (Fig. 4).

Apart from these modules, each form has its own coding which is used for the physical designing of the forms at run time. The parts of the coding that are common to more than one form are constructed into Visual Basic modules.

Some of the Visual Basic modules used in MapViewer are:

- Layout
- Move
- Size
- Rotate
- Menus
- Reorder

The layout module takes care of the manipulation of layout of the windows, like re-parenting them to the Visual Basic window, setting the layout units and all the other functions that can be performed on a normal layout window in MapViewer (Fig. 5).

The sizing and moving modules are used to handle the 2-D functionalities of MapViewer. These modules are mainly composed of Visual Basic code and are used to manipulate the map windows' size and position.

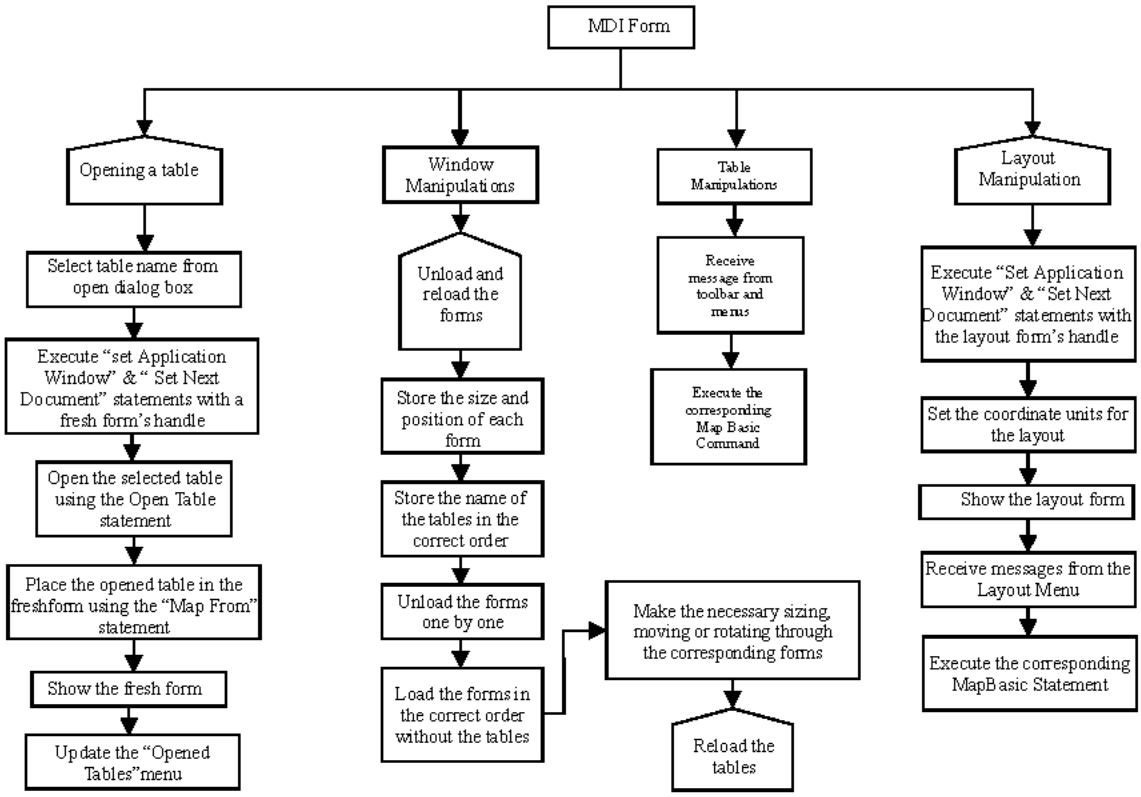

Fig. 4: Overall sequence of control inMapViewer

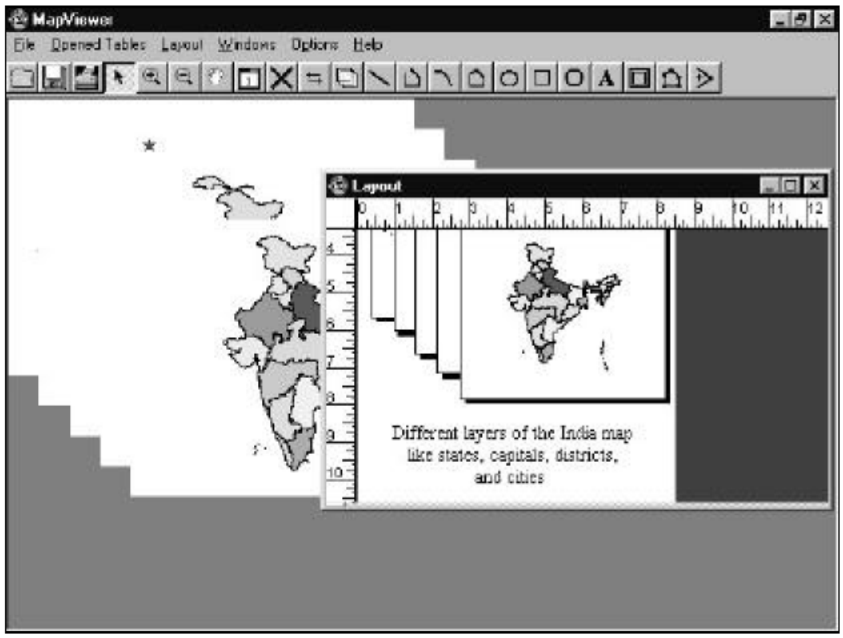

Fig. 5:The layout window of MapViewer after adding text captions. This feature helps in designing the maps and the tables for taking printouts or for exporting them as bitmaps, Giff or other file formats

MapBasic statements are used only in reloading the maps after the manipulation of the windows.

The rotate module implements one of the distinguishing features of MapViewer-the 3-D functionality. Using this module, the Visual Basic form (map window) is made to rotate three dimensionally.

The menus module organizes the activities of menus and toolbars of the main MDI form. Its main function is to track the addition and deletion of map windows and update the runtime menu, which contains the names of the tables that are opened. The reordering module helps in the process of changing the order of the tables that are open and to update the opened tables' runtime menu.

TESTING AND IMPLEMENTATION

Testing: MapViewer was subjected to the following tests and the requisite changes were done till repeated use of the software by all-comers brought forth no hitch or discomfort.

This involved testing of each individual program segment for general robustness of the system.

MapViewer was also subjected to unit testing, function testing, performance testing and integration testing.

Unit testing: All the icons and menus were tested whether they are performing the intended function.

Function testing: Functional test involved testing the system under typical operating conditions, typical input values and for typical expected results. The functional boundary specifies the framework within which the system can function. Three types of functional tests were done to check:

- If all the documented functions are working
- The minimum and maximum input values
- The extent to which it can run properly on giving valid inputs

Performance testing: Performance tests are conducted to identify any bottlenecks in the system and to fine-tune the overall performance of the system. This has been done by observing the behavior of the system by giving all possible combinations of input values and overloading the system in various ways.

Integration testing: The system was finally tested after integrating all the functionalities, since some functions may not work properly when run after integrating.

CONCLUSIONS

A system has been developed which significantly enhances the utility and the user-friendliness of the existing GIS software. The application of the concept to the popular GIS software MapInfo has resulted in the add-on tool MapViewer. This study describes the advancements achieved through MapViewer which include enhanced graphic user interface, 3D rotation of map windows, resizing of the map windows, re-positioning of the windows, changing their borders and a more powerful info tool.

Protracted tests and hassle-free multi-party utilization of MapViewer for over an year indicate the durability and user-friendliness of MapViewer.

ACKNOWLEDGEMENTS

SAA thanks the Ministry of Water Resources, New Delhi, for a major research project, which supported this study. KBC thanks the Council for Scientific and Industrial Research, New Delhi, for the senior research fellowship.