The Implementation of DXF Application On Smart Devices

Dong-Hyun Kim, *Member, KIMICS*

***Abstract*— As the mobile devices have powerful CPUs and memories, it is possible to carry the large volumes of design drawings, such as DXF files, on mobile devices. However, since the format of a DXF file is the ASCII, it has the shortcoming where the size of a file is larger than a binary formatted file and it is difficult to display the entities of drawings quickly on mobile devices.**

In this paper, we design the DXF manager system on a smart phone. To speed up the rate of the display and panning, the DXF manager hires the spatial index, the fixed grid file. We also implement the system on the Samsung Omnia-1® to exploit the Windows Mobile® operating system.

***Index Terms*— Smart phone, DXF, Fixed grid file, Mobile application.**

I. INTRODUCTION

ON building construction sites or ship yards, generally, engineers carry along paper drawings in order to refer to the values or the figures of designs. However, the papers are fragile and easy to lose the data of designs. As the mobile devices, such as smart phones or PDAs, become to own powerful CPUs and memories, it is possible to run heavy applications and carry the large volumes of data on the mobile devices. Thus, it is required to study the scheme to refer to the data of design drawings on the mobile devices.

The CAD(Computer-Aided Design) is the design system where designers are able to draws their own designs on computer monitors using pens or mouse and store drawings into storage systems as files. Since the designers are able to modify drawings easily to reflect the changes of fields or the requests of clients, the CAD is efficient and highly productive tool. The CAD is widely used on various fields, such as buildings designs, machine designs, and circuit designs etc.

The CAD exploits the DWG file format in order to store the data and the meta-data of the design drawings. Since the DWG format is the native format of AutoCAD® system, the DXF(Drawing eXchange Format) is hired to export the data of drawing to the different CAD system. The DXF file is composed of seven sections. Among sections, four sections are important because they have the data related directly to the entities of designs: header section, table section, blocks section and entities section.

Since the DXF is the ASCII formatted file, it has the benefit to read the data of entities easily. However, it also has the shortcoming where the size of files is larger than a binary formatted file. Since most designs have thousands or ten-thousands of entities, the sizes of files are tens of megabytes or hundreds of megabytes. When a user loads and view the design of the DXF file which is a large volume, it requires quite a time to display or pan entities on the screen.

In this paper, we design and implement the DXF manager system on a smart phone. To speed up the rate of the display and panning, the DXF manager hires the spatial index based on the fixed grid file. In order to diminish the display time, the index of a DXF file is built on a stationary device before the DXF file is loaded on the smart phone. We also implement the system on the Samsung Omnia-1® to exploit the Windows Mobile® operating system.

The rest of the paper is as follows. In section 2, the related works are presented and the DXF is described in section 3. We present the design of the DXF manager system in section 4 and implemented the prototype system in section 5. The conclusion is described in section 6.

II. RELATED WORKS

The existing works for the CAD are mainly focused on the updates of CAD files. In [1][2][3], they proposed the scheme to keep the integrity of CAD data when updates are arisen by different users. [1] proposed the transaction model for the CAD application. The proposed model is based on the nested transaction model and shows the novel scheme of the transaction split. They also present the new features of the CAD transaction. [2] presents the four basic requirements for the CAD data management and propose the Version Management System(VMS). The

Manuscript received July 27, 2011; revised August 29, 2011; accepted September 5, 2011.

Dong Hyun Kim is with the Division of Computer Information Engineering, DongSeo University, Busan, 617-716, Korea (Email: pusrover@dongseo.ac.kr)

proposed architecture of VMS is independent of CAD tools, so that they can be integrated with VMS without modifying the source code. In [3], they proposed the version management system for the construction CAD databases. The proposed system shows the sharing management of design objects, the dynamic reference of the construction objects and the updates notification of the design objects. However, since the existing works are assumed that the CAD system is running on the stationary workstation, they are not appropriate for the CAD system of mobile devices.

III. DXF

The DXF is the file format for the exchange of the drawing file between different CAD softwares. At first, the DXF is defined by the Auto Desk cooperation in order to export the data file of AutoCAD to the other applications. However, since the market portion of AutoCAD becomes over the fifty percents and the format is opened to the public, the other CAD softwares hire the DXF as the exchange format. Recently, the DXF is de-facto standard in US and similar to the standard in other countries.

Since the data format of DXF is the ASCII, the contents of the DXF can be read easily in the other applications, even the text editor. Each line of the DXF file has the only one value which represents the field name or the data value itself. Since the format is simple, it is easy to manipulate and understand the data. However, it also has the defect where the size of a file is a large volume.

The lines of the DXF are grouped by sections. It has seven sections: HEADER, CLASSES, TABLES, BLOCKS, ENTITIES, OBJECTS, THUMBNAILIMAGE section. Among these sections, following four sections are most important to figure out the drawings. The HEADER section contains the definition of header variables. The TABLE section presents the schema of each table for the entities. The BLOCKS section defines the blocks of the drawing and contains the values of entities which are contained in the blocks. The ENTITIES section has the values of entities which are not contained in blocks of the drawing. The following figure shows the whole format of the DXF

As shown in Fig 1, each section starts with the value 0. The basic unit to store the information of each section is a group. A group is composed of two lines. The first is the group code and the next is the group value.

```
0          //the begin of the HEADERS section
SECTION
2
HEADER
.....      //the header variables
0
ENDSEC     //the end of the HEADERS section
0          //the begin of the TABLES section
SECTION
2
TABLES
0
TABLE
2
VPORT
70
.....      //the variables of a table schema
0
ENDTAB
0
TABLE
2
APPID, DIMSTYLE, LTYPE, LAYER, STYLE, UCS, VIEW, VPORT
70
.....
0
ENDTAB
0
ENDSEC     //the end of the TABLES section
0          //the begin of the BLOCKS section
SECTION
2
BLOCKS
.....      //the components of entities in the block
0
ENDSEC     //the end of the BLOCKS section

0          //the begin of the ENTITIES section
SECTION
2
ENTITIES
.....      //the values of the entities not in the blocks
0
ENDSEC     //the end of the BLOCKS section
0
EOF        //the end of a file
```

Fig. 1. The whole format of the DXF file

IV. DXF MANAGEMENT SYSTEM

A. the Spatial Index

The Fixed Grid File is one of the methods which index objects which reside on a multi-dimensional space. The fixed grid file decomposes the whole data space into a grid of cells which are fixed sizes. It consists of two parts. The first is a d-dimensional array, containing one entry for each grid cell. Each elements of the array contains a pointer which points at a relevant bucket. The second is the bucket which contains the value of a data entry. The Fig. 2 shows the example of the fixed grid file.

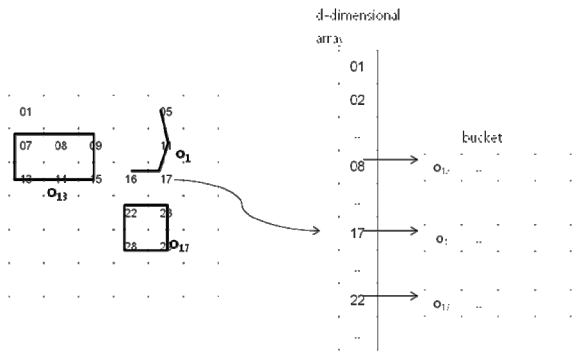

Fig. 2. An example of the fixed grid file

To speed up the display of entities, we exploit the fixed grid file structures. Since the memory and storage of a smart phone are smaller than those of workstation, we build only the d-dimensional array as an index file and the values of elements denote the positions of the data entries on a DXF file.

The index file is composed of three sections: Screen Size, Layer Info List and Grid File List. The following Fig. 3 shows the file format of the index file.

Fig. 3. The format of the index file

Fig. 4 presents the structures of Screen Size, Layer Info List and Grid File List. The Screen Size represents the whole size of a drawing space on a DXF file. It stores the coordinate values of top, left, bottom and right which are coordinates of the drawing. Layer Info List is the space which stores the color data of entities. The layer Name is the layer name of entities and the layer Color is the color value of the layer. The color value follows the definition of the AutoCAD.

The Grid File List stores the coordinate values of each cell and the relative file positions of entities which are contained in the cell. The top, left, bottom and right is the coordinate values of each cell. The Block Point of the File Point List represents the file positions of each block on a DXF and Entity Point is the position of an entity.

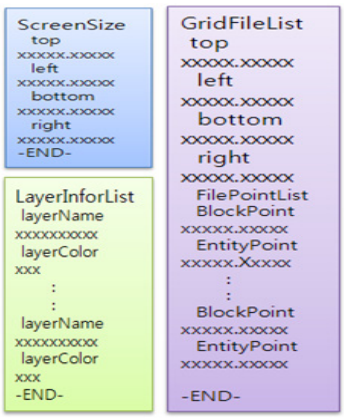

Fig. 4. The structures of Screen Size, Layer Info List and Grid File List

B. The Structure of the DXF Manager

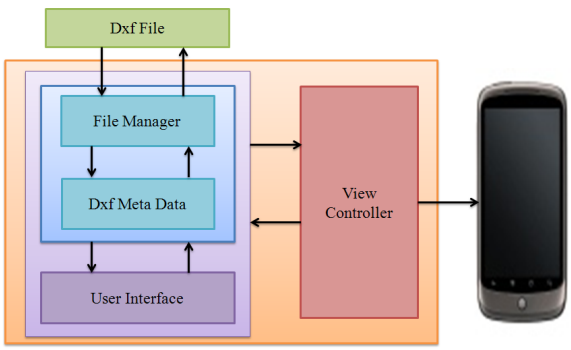

Fig. 5. The System Architecture

Fig. 5 shows the system architecture of DXF manager. The system consists of File Manager, Dxf Meta Data, User Interface and View Controller classes. When a user opens a DXF file, the File Manager class opens the given DXF file and the related index file. Then, the File Manager class builds the structure for the meta-data and the index and then, transfers the structures to the Dxf Meta Data class. When the user wants to handle the data, the User Interface class receives a related command. The View Controller class follows the given command and displays the data of Dxf Meta Data class.

The File Manger class opens a DXF file and an index file using the File Open() and Grid File Read() methods. After analyzing and building the data structures of opened files, it gives the structure to the Dxf Meta Data class. Using the File Point Read() method, it also retrieves the values of the specific entity at the DXF file in order to display the entity on the screen. Fig. 6 shows the diagram of the File Manger class

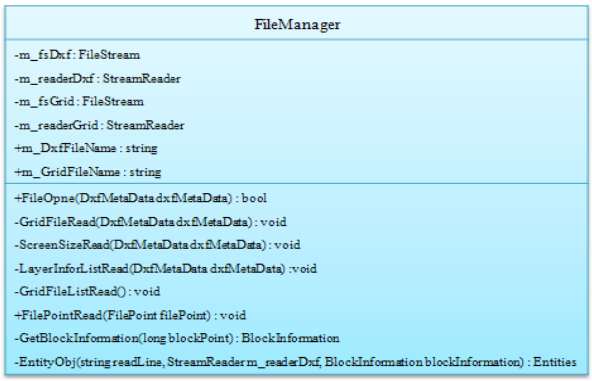

Fig. 6. The File Manager class

The Dxf Meta Data is the class which stores the data structures analyzed at the File Manager class. The Screen Size presents the whole size of a drawing and the Layer Information is each layer data. The block data is stored in the Block Information. The data of a decomposed cell and the file positions of entities are stored in the Grid File.

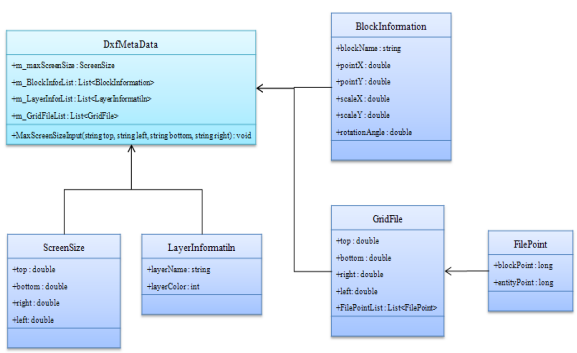

Fig. 7. The Dxf Meta Data class

The User Interface class provides the user interface of the DXF manager to a user and shows various functions, such as zoom-in, zoom-out, move and restoring original size, using Zoom In_Click(), ZoomOut_Click(), UI_Mouse Move() and Original Size_Click() methods. The View Controller class sets the coordinate system of a smart phone and resizes the size of a drawing in order to fit the screen size of a smart phone using Screen Down Size() method. It also draws entities of a drawing using Entity Paint() method.

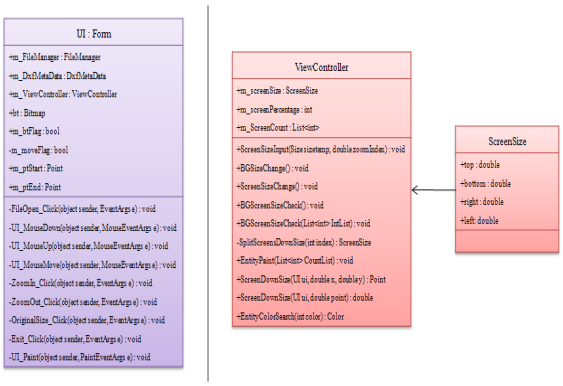

Fig. 8. The User Inter face and the View Controller class

The Entity class draw the entity using Paint() method and stores the configuration of each entity because each configuration of a entity differs from others.

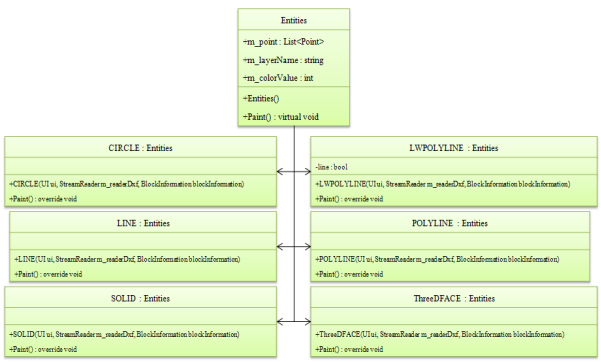

Fig. 9. The Entity class

C. The Execution Sequence

When a user opens a DXF file, The File Manager class opens the DXF file and related the index file. After analyzing the files, the File Manager transfers the meta-data of the opened file to the Dxf Meta Data class and the whole extent of the DXF drawing to View Controller class. The View Controller class requests the file positions of the entities to the Dxf Meta Data class in order to draw the entities on the screen. After taking the file positions of the entities, the DxfMetaData class reads the coordinates of entities through the FileManager class. Fig. 10 shows the execution sequence between classes when a DXF file is opened.

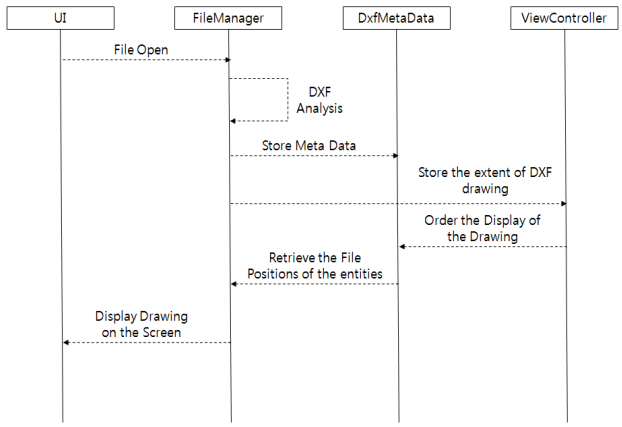

Fig. 10. The Execution Sequence to open a DXF file

Fig. 11 shows the execution sequence to zoom-in, zoom-out and move the drawing. When The UI class receives the zoom-in, zoom-out and move commands, it initiates the ViewController class in order to update the data of the current screen. The ViewController class requests the coordinates of the entities contained in the updated size of the screen to the DxfMetaData class and the DxfMetaData class reads the coordinates of the contained entities according to the file positions of the entities.

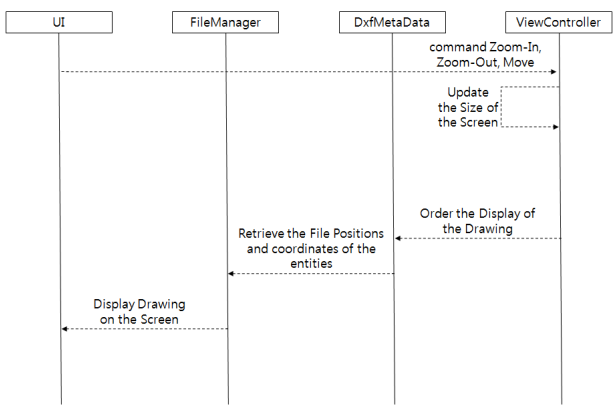

Fig. 11. The Execution Sequence to Zoom-In, Zoom-Out and Move

V. IMPLEMENTATION

Table 1 shows the implementation environment. We implement the DXF manager system on a SamSung Omnia 1 smart phone.

TABLE 1.
THE IMPLEMENTATION ENVIRONMENT

Model	Omnia1(SCH-M490)
Resolution	WVGA(480X800)
Screen Size	3.3 inch
Memory	4GB
OS	Windows Mobile 6.1

We test the prototype system using four different drawings which are construction architecture drawings and AutoCAD 2010 version. The Fig. 12 shows one of the test drawings.

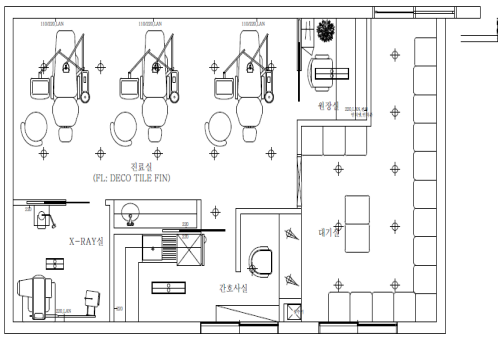

Fig. 12 The Test Drawing

Fig. 13 shows the first page when the prototype system is executed on a smart phone and Fig. 14 shows the dialog windows to open DXF files.

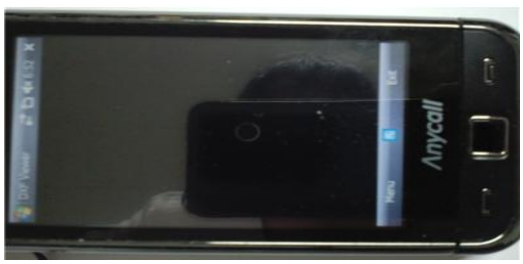

Fig. 13. The First page of the system

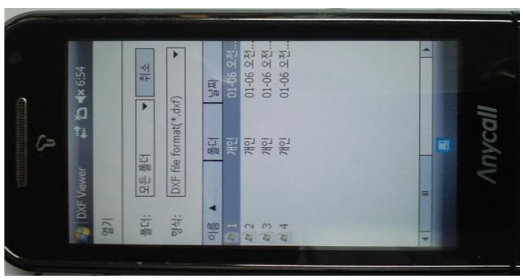

Fig. 14. The Dialog Window to open DXF files

Fig. 15 shows the whole drawing when a user opens the drawing at first.

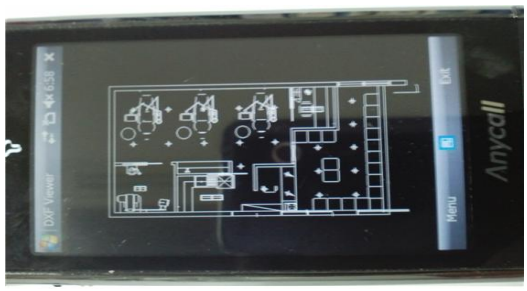

Fig. 15. The Whole Drawing

Fig. 16 and Fig. 17 presents the screen when a user executes the Zoom-In and the Zoom-Out functions.

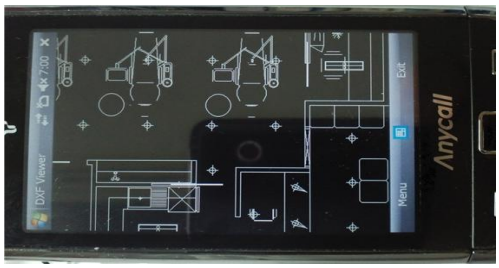

Fig. 16. The Screen of Zoom In

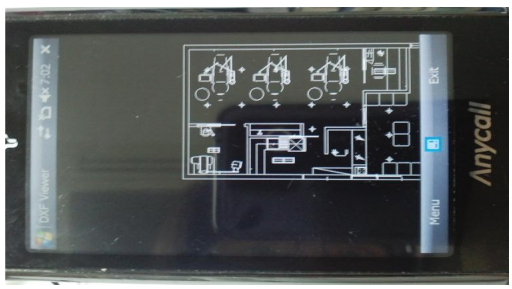

Fig. 17. The Screen of Zoom out

Fig. 18 shows the screen of the drawing movement during the drawing is zoomed out.

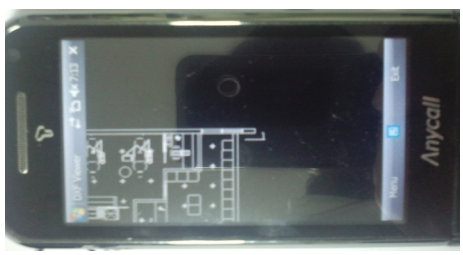

Fig. 18. The Screen of Movement

VI. CONCLUSION

As the mobile devices become powerful, it is possible to run heavy applications and carry the large volumes of data on mobile devices. Thus, it is

required to study the scheme to refer efficiently to the designs of drawings on fields using mobile devices.

Since the DXF is the ASCII formatted file, it has the shortcoming where the size of files is larger than a binary formatted file. In this paper, we designed the manager system for the DXF files on a smart phone. To speed up the rate of the display and panning, the DXF manager hires the spatial index based on the fixedgrid file. We also implemented the prototype system on the Samsung Omnia-1®. Future work is to build the DXF manager system on top of different platforms, such as Android® and iOS® and to exploit different spatial index structures.

ACKNOWLEDGMENT

This research was supported by special research grant funded by Dongseo University.

REFERENCES

[1] Hyun Seok Kang, Seok Woo Han, Sook Young Choi, Myung Shin Gwak and Chol Dong Lee,"A Study on Transaction Management of CAD Database," *KIISE Database Society of Korea*, vol. 6, no. 1, 1990.
[2] Hae-Jin Lee, Ju-Won Song, Wan-Sup Cho, Kyu-Young Whang, Young-Uk Yu, Gil-Soon Kang, "Design and Implementation of a Version Managed system for CAD Databases," *Journal of KIISE,* vol. 21, no. 3, 1994.
[3] Whal Soo Kim, Mi Kyung Min, Hyun Joo Park, "Version Management System for Architectural CAD Database," *Proc. Of Annual Workshop of KIISE*, vol. 25, no. 1, 1998.,
[4] Nievergelt J, Hinterberger H, Sevcik KC (1984) The Grid File: An Adaptable Symmetric Multikey File Structure. ACM Transactions on Database Systems (9)1:38-71
[5] Autodesk, "DXF Reference", http:// usa.autodesk.com
[6] H. W. Garter, "Computer-aided design of integrated circuits," IEEE computer, Vol 19, No 4, 1986
[7] Pradeep Padala, "A Survey of Grid File Systems," Grid File Working Group, Global Grid
[8] Hanan Samet, " Foundations of Multidimensional and Metric Data Structures," Morgan Kaufmann Publishers, 2006

Dong Hyun Kim received the B.S., M.S., and Ph.D degrees in Computer Engineering from Pusan National University, Busan, Korea, in 1995, 1997 and 2003, respectively. His research interests include Databases, GIS, LBS, Sensor Databases and Indexes.