DESIGN OF THE EXCEL VBA PASSWORD CHANGING MODULE

Jang, Seung-Ju[1]

[1]Department of Computer Engineering,
Dongeui University, Korea
Sjjang@deu.ac.kr

ABSTRACT

In this paper, I design a module that can change password when Excel VBA password is set. Especially, I designed it to be applicable to VBA environment using macros in Excel. It often happens that you cannot access important file data due to password loss, after setting the VBA password. For these users, it is possible to access valuable data by more convenient VBA password using the proposed program. I actually implemented the proposed design in this paper. Experiments were performed on the actual contents of the excel file. Experimental results show that the contents of Excel VBA password can change.

Keywords: Excel VBA, Password Change, Password Recovery, VBA(Visual Basic for Application), Excel File Security.

1. INTRODUCTION

This paper relates to an Excel spreadsheet that we use daily in our daily life. VBA is often used in Excel spreadsheets for complex data processing. We design a module that enables changes to the password used in this VBA. Recent Excel spreadsheets have an OLE structure. Object Linking and Embedding (OLE) files (or COM Structured Storage) format has been used as document formats for Microsoft Office, Word, Excel, and PowerPoint. The OLE file format has the same file system structure as the internal hard disk. Therefore, there is the concept of folders and files inside the OLE file.

Since OLE files are one small file system, MS Office has backward compatibility and upward compatibility. For example, even if you create and save a document in MS Word 2007, you can read it in MS Word 2003. If you create a storage / stream that is used by the version of MS Word 2007 and a storage / stream that is used by the version of MS Word 2003 inside the OLE file, then the version of MS Word 2003 only refers to that version of storage and stream. Even if the MS Word 2007 related storage and stream exist, the version will not access the area, and it will have the effect as if the lower version software reads the document in the upper version without any problem [1, 2].

In this paper, I design a module that can change password when Excel VBA password is set. Especially in Excel, I designed it to be applicable to VBA environment using macros. After setting the VBA password, it often happens that you cannot access important file data due to password loss. By using programs for these users, it is possible to access valuable data through more convenient VBA password access. I actually implemented the proposed design in this paper. Experiments were performed on the actual contents of the excel file.

In this paper, Chapter 2 refers to related research, Chapter 3 refers to VBA password module design, Chapter 4 experiment, and Chapter 5 conclusion.

2. RELATED RESEARCH

The following Fig. 1 shows the internal structure of a doc file written in MS Excel simply by SSViewer (Structured Storage Viewer) program.

Figure 1. Internal Structure of OLE Document using SSViewer
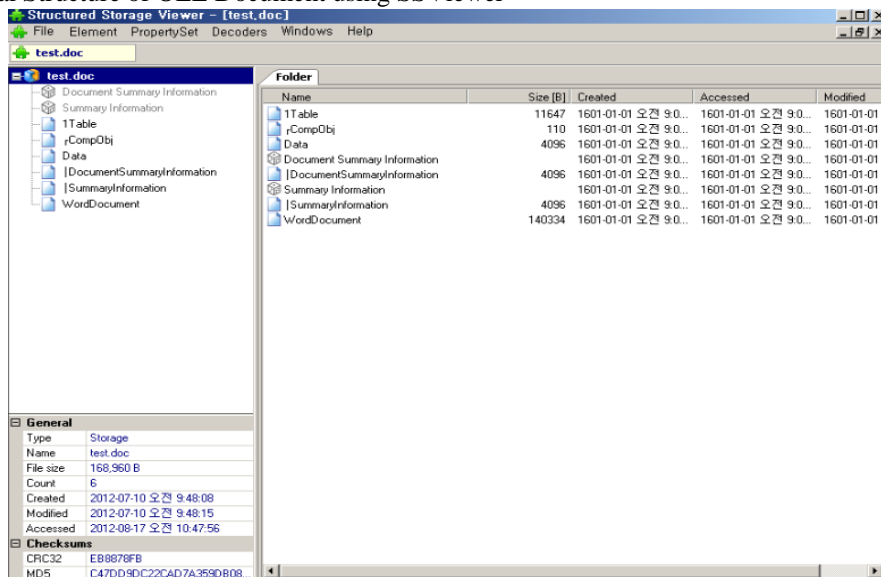
OLE file is divided into two big blocks. They are a header block and a data block. The header block has a size of 512 bytes, and the data block has 512 bytes or more. The header block contains the main information of the entire OLE file, and the data block has the following information.
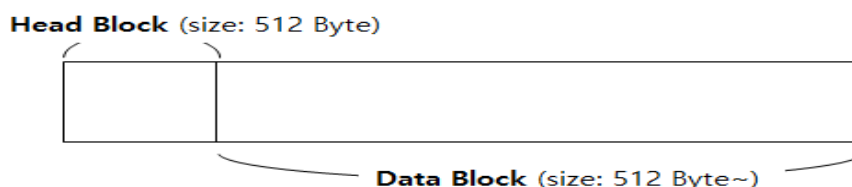

Figure 2. Structure of OLE data block

- Properties
- Stream data
- Big Block Allocation Table (BBAT)
- Small Block Allocation Table (SBAT)

Most of the data blocks are occupied by the stream data.

2.1 Structure of OLE Block

An OLE header block is a block that contains the main information of an OLE file. The OLE file is divided into 512 bytes and the block number is assigned.
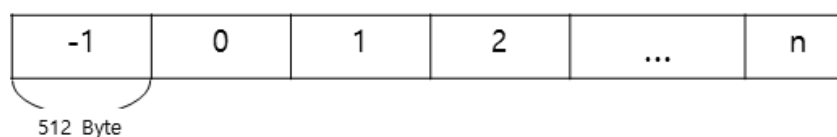

Figure 3. Structure of OLE Block

In this case, -1 block is a header block, and 0 .. n block is a data block. When opening an OLE file and reading it by 512 Bytes, it is like reading the header block. An OLE file is a file format used in various word documents. It consists of storage and streams similar to the FAT file system. Due to the limitations of the traditional file system, CFB(Compound File Binary) is structured according to the needs to put file and folder structure in one file [3, 4].

2.2 Storage and Stream

You can think of an OLE file as a normal file system. It consists of storage of folder role and stream of file role. It takes a very similar approach to the FAT file system to store storage and streams within OLE files. If you have a basic knowledge of the FAT file system, you can access it more easily. All information is stored in 512-byte sector units.
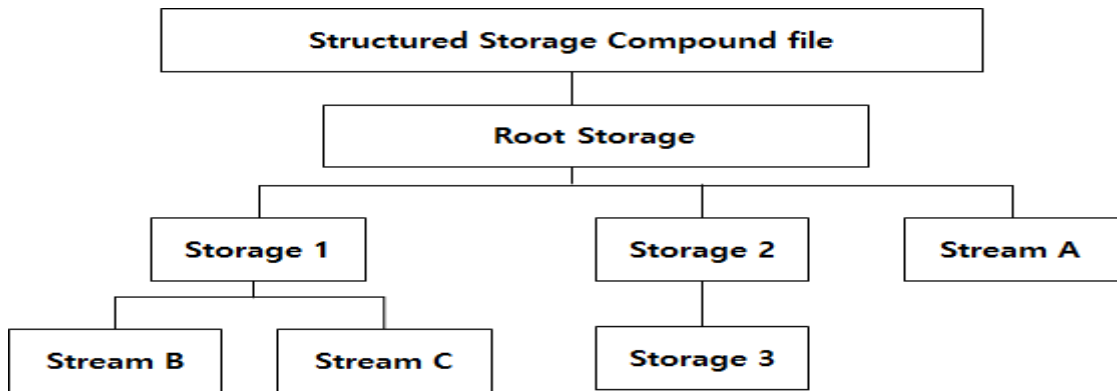
Figure 4. Structure of OLE storage and stream

2.3 CFB(Compound File Binary) Components

To understand the compound file, you need to know the various components that make up the file. The Fig. 5 shows the structure of the compound file.
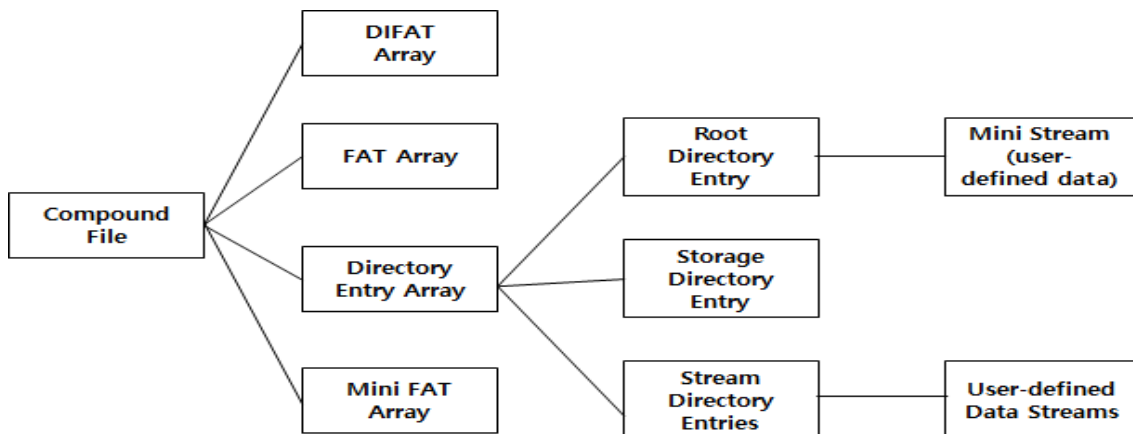
Figure 5. Structure of OLE Compound File

2.4 FAT(File Allocation Table) Array

A FAT Array is an array of 32-bit integers. The index of the array refers to the sector number, and the element corresponding to the index refers to the next index number. OLE file does not store the stream as a single line, just as a large file is not written to the disk from the beginning to the end. The storage is divided into sectors of 512 bytes size.

2.5 Directory Entry Array

A directory entry is where the meta information related to the storage stream is stored. A Directory Entry Array is an array of this Directory Entry. Directory Entry Arrays have meta information, so if you have Directory Entry information, you can easily access the information [7, 8, 9].

3. DESIGN OF VBA PASSWORD CHANGING MODULE

As we have seen in the related study, it is necessary to analyze the file structure of Excel in order to design Excel VBA password release module. The internal structure of Excel can be checked using SSViewer (Structure Storage Viewer).
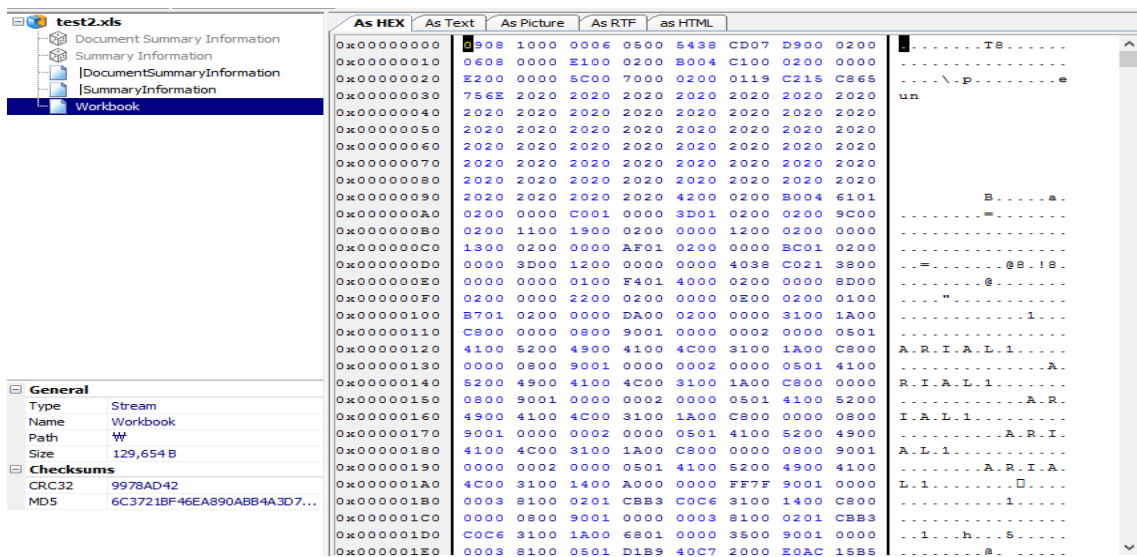

Figure 6. Structure of Test2.xlsm File

The Excel file has an OLE structure as shown Fig. 6 above. The OLE structure consists of a header and a block. For Excel files that use macros, the extension is in .xlsm format. If you open Excel's xlsm file format with Notepad, it is shown like Fig. 7 below. As shown in Fig. 7, the Excel file format has an unknown structure. This is because the Excel file is compressed. In the case of an Excel file that uses a macro, the format of the file is a compressed file.

Figure 7. Binary Structure of Excel File

In this way, macro-format files in Excel are basically compressed. Therefore, the macro file of Excel can be opened by changing the extension to compressed file format. The following Fig. 8 is the result of opening a macro format Excel file using a compression program.

By using these results, it is possible to recover the password error environment in VBA more conveniently by allowing the users to change the extension name etc. by using the program without manually performing this process.

When opening the above compressed file, there is a .bin file. The VBA program is stored in this file. In this paper, we change the .xlsm extension to .zip and open the modified compressed file using a compression program. This process is designed to be done through the program. If the compression program opens successfully, it looks for the .bin file. When you find the .bin file, it is extracted from the compressed file.

It then programs the .bin file extracted from the compressed file to use binary edits. It runs binary edits so that you can edit the .bin file extracted from the compressed file. It is written in a program so that these processes can be operated.

When you run binary editing of the .bin file, it modifies the DPB string in the .bin file. It changes DPB string to DPx string. It will then save the .bin file. This process is shown in Fig. 8.

Figure 8. Decryption Process
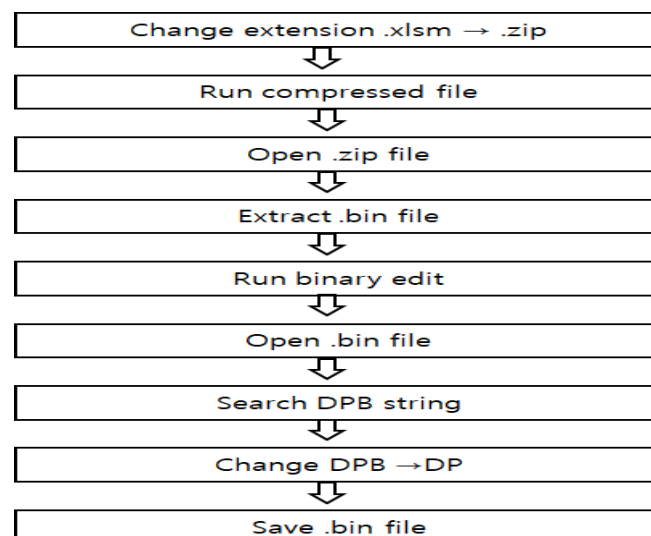
4. EXPERIMENT

I design the contents in this paper using C program. The operation of the implemented program is shown in Fig. 9.
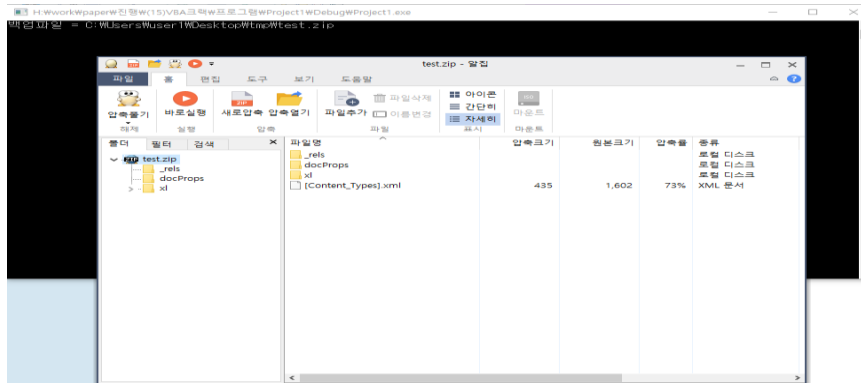
Figure 9. Screen of Program Operation

Fig. 9 above shows the program screen that executes the proposed design in this paper. The program performs the function of changing the .xlsm file to a .zip file.
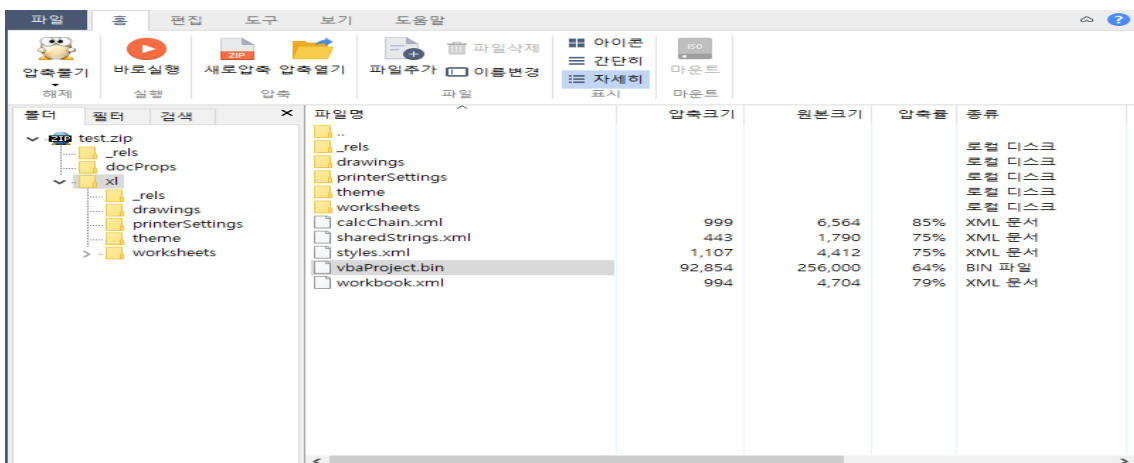
Figure 10. Screen of Compression Program Execution

Fig. 10 shows the screen of compression program execution. It extracts the vbaProject.bin file from the compressed program. The program is that binary edits can be performed on the extracted vbaProject.bin file.
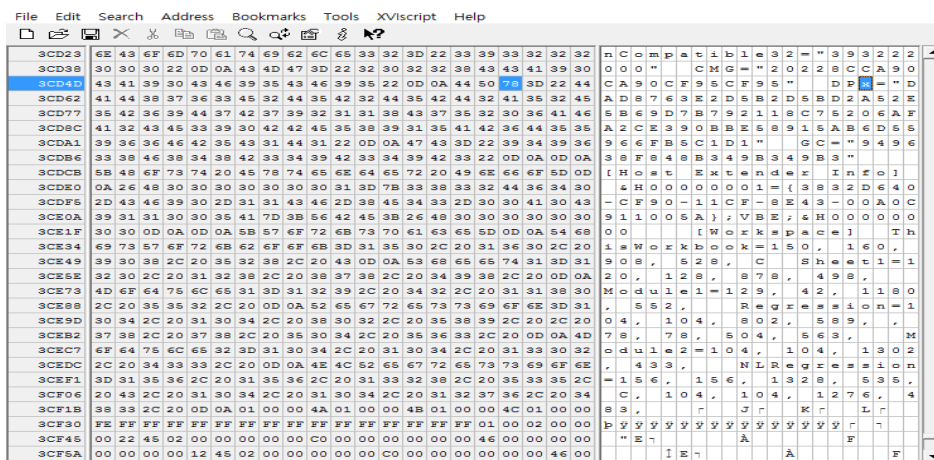

Figure 11. Screen of Starting Binary Edit on vbaProject.bin

After performing the binary edit of the vbaProject.bin file, it looks for the DPB string. After finding, it modifies the DPB string with DPx. The process of modifying with DPx. After

modifying the DPB string, it saves this file. When the vbaProject.bin file has been modified, the vbaProject.bin file is added to the compressed file.
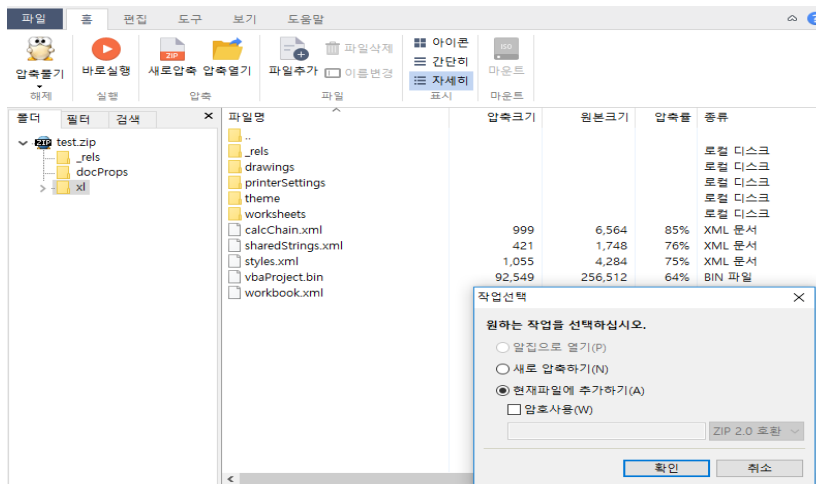

Figure 12. Add the Modified vbaProject.bin File to the Compressed File

After adding the modified vbaProject.bin file to the compressed file as shown in Fig. 12, it returns the .zip extension name back to .xlsm. This process is shown in Fig. 13 below.
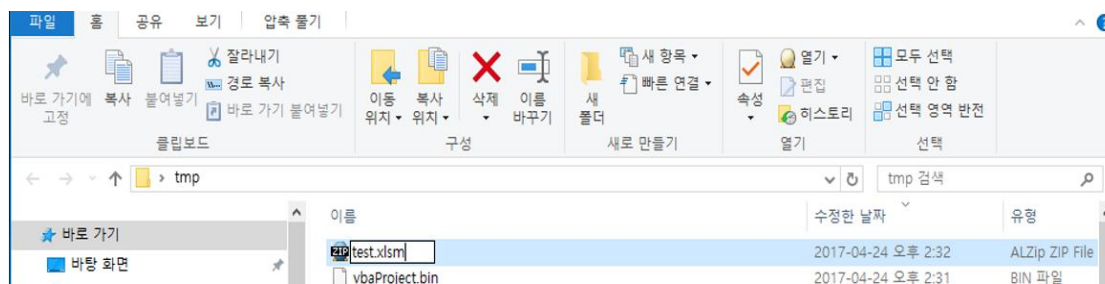

Figure 13. Return Extension Name to .xlsm

After returning the extension name to .xlsm as shown in Fig. 13, you open the Excel file. If you open this Excel file, the file will open. Then you can set a new project property password in the security function area. I set "123" as the new password. Then, we store the Excel file. When we exit Excel and open it again, it works. When you run the VBA module, you must enter the password. At this time, if you input the newly changed password "123", you can check that the VBA module opened normally.

5. CONCLUSION

In this paper, I designed a system that could change the password when using Excel VBA. Especially, I designed it to be applicable to VBA environment using macros in Excel. It often happens that you cannot access important file data after setting the VBA password, due to password loss. The proposed program can make it possible for these users to access valuable data using more convenient VBA password change module.

I designed a program that makes it easier to release the password for the file that is set VBA password. The structure of this program is roughly divided into two parts. One is the process of extracting the .bin file from the Excel xlsm file. The other is the configuration process to modify the VBA password through changing the DPB keyword in the extracted .bin file.

I designed the program so that the above two processes can be used easily and conveniently. It changes the extension of the xlsm file to zip. After changing to zip, it starts a program that can decompress. When the decompression program is started, it extracts the vbaProject.bin file from the compressed file.

Next, it performs to edit binary of the vbaProject.bin file extracted from the compressed file. Then it runs the binary edit program to edit the vbaProject.bin file. When the binary edit program starts, it loads the vbaProject.bin file. If the vbaProject.bin file is opened normally, DPB string modification work will be done in this file. When the string modification is completed normally, the changed vbaProject.bin file is saved.

Once the modification of vbaProject.bin file has been completed, it will be saved in the original compressed file. When the vbaProject.bin file is newly saved, the extension zip of the compressed file is modified to the original extension xlsm.

REFERENCES

[1] Robin Abraham, Margaret Burnett, Martin Erwig, (2009) Spreadsheet Programming, Wiley Encyclopedia of Computer Science and Engineering.

[2] Thomas A. Grossman, (2008) "Source Code Protection for Applications Written in Microsoft Excel and Google Spreadsheet", arXiv.org > cs > arXiv:0801.4774 Software Engineering.

[3] Jocelyn Paine, (2009) "Documenting Spreadsheets with Pseudo-Code: an Exercise with Cash-Flow and Loans", arXiv.org > cs > arXiv:0908.1187 Software Engineering.

[4] Larry J. LeBlanc, (2007) "Implementing Large-Scale Optimization Models in Excel Using VBA", *Interface*, Volume 37, Issue 4, pp. 370 – 382.

[5] J Pichitlamken Email author, S Kajkamhaeng, P Uthayopas, R Kaewpuang, (2011) "High performance spreadsheet simulation on a desktop grid", *Journal of Simulation*, Volume 5, Issue 4, pp 266–278.

[6] Michael R. Bartolacci, Larry J. LeBlanc, Yasanur Kayikci, Thomas A. Grossman, (2012) Optimization Modeling for Logistics: Options and Implementations, *Journal of Business Logistics*, Volume 33, Issue 2, pp. 118–127.

[7] Mangesh S Gharote and Dr. Sachin Lodha, (2010) "ExBOS – Excel Based Optimization Software Framework", SOM 2010 XIV Annual International Conference of the Society of Operations Management, pp.1-8.

[8] http://www.reversenote.info/ole-file-format/

[9] Pichitlamken1, S Kajkamhaeng, P Uthayopas and R Kaewpuang, (2011) "High performance spreadsheet simulation on a desktop grid", *Journal of Simulation*, no. 5, pp. 266–278.