# Modelling of a Sample Help Guide Interface

0958330

**Abstract**

This paper is an attempt of modelling the process of people gaining and understanding the solutions to achieve a certain goal, with a discussion of an original model to illustrate how EM might be used in an application. The whole model takes the a particular case in the real world-- how to successfully transfer the data, which includes both the types of "folder" and "a single file", to the C drive of the user's computer. The model has estimated a particular path of issues which the users who is in needing the help may face during the path towards finally succeed in transferring the data. The main aim and significance of this constructing this model is to try to make it an extension of the application of the real world, along with well implemented approaches and principles of the subject of empirical modelling.

## 1 Introduction

As is concerned today, computers have already become one of the necessities in people's daily lives. And it is definitely an obvious truth that people have already gained a lot of conveniences in dealing with daily issues with being assisted by the effectiveness of computer. However, unfortunately, it seems that not everyone has the capability of understanding and manipulating the computer as a computer technician can do. Sometimes people may find it really tough and complicated to realise one function by following a series of steps in using a series of software. Meanwhile, even after searching on the web to gain some information about how to manipulate the processing of computer, the instructions are really abstractive and very difficult to be comprehended for their containing of too many terminologies.

Such this consequence is caused by the gap between human thoughts and the running mode of computer. To get rid of this gap, one thing to do is to make the human thought be comprehensible to computer, while the other crucial task is to make the response of computer be meaningful to the human comprehension. As a consequence, the more we can do for the above two tasks, the less of the gap between the human mind and computer processing strategy will be.

In order to make an attempt to make some effort to fulfil these two tasks and also to illustrate that the approach and knowledge which is provided and expressed by empirical modelling would be helpful to the deal with the real world application, especially to that of the arena of human computing, the model is set up to deal with the question of "how to transfer data successfully from flash disk to the C drive"----a particular kind of daily applications, while the modelling methodologies are all based on the modelling approaches of empirical modelling.

## 2 Think the Model

### 2.1 Case in the model

The model is going to help the user successfully transfer the data from a flash disk onto the C drive. Generally, the data on the flash disk to be transferred is divided into two main types----single file, which means just a file itself and also with some extensions like doc, exe and etc. ; and also the folder, which contains files or folders inside it. For this reason, the model presents the data on both the locations of flash disk and C drives in terms of the two data types: folder and file.

Hence, on the image of the model there will have a larger box which means folder, along with its name, and a smaller box which means file, along with its name, shown on both sides of flash disk and C drive. The most concerned properties of the data are shown in different ways. The size, name and the free space of the C drive are shown dynamically in a form of list. And there also have the tabs which said "FDisk" and "CDisk" and are located just on the left side of the boxes of folders and files in order to sign the current location of the file and folder which locates on the right side of it. This way can give the user a more vivid and direct idea of the movements of the data.

## 2.2 Logical analysis on action process

On the whole, the case of transfer folder and file from flash disk to C drive can be logically defined in to the following steps of conditions in accordance to the most frequently situations in the real life:

1. The current location of the data to be transferred.
2. The user clicks on the file or folder in the location of flash disk which he or she wants to transfer to the C drive.
3. There is a sign which can show the whether the data which the user wants to transfer is selected.
4. If the user selects the folder, it needs to be examined that whether it shares the same name with the folder already exited on the C drive. Also, if the user selects the file, it needs to be examined that whether it shares the same name with the file already exited on the C drive. Hence, the name of the data on both drives, along with the warning and solution to the current status needs to be observable to the user.
5. If the selected data has taken over the size of the free space on C drive, then it is not permitted to transfer. Thus, information about the size of data and free space on C drive, along with the warning and solution to the current status needs to be observable to the user.
6. If there are no restraints as are mentioned in tip 5 and 6 at all, the user can then succeed in transferring the data he or she wants. That means there must be the sign to tell the user either the folder or the file has been moved to the C drive.

# 3 Construct the Model

## 3.1 Observables & Dependencies

The entire design of the model is made by implementing the knowledge of EDEN, DONALD and SCOUT.

The warning and solution to the condition of NO ENOUGH SPACE as is mentioned on tip 5 above.

<label No_enough_space>

The warning and solution to the condition of SHARE THE SAME NAME as is mentioned on tip 4 above.

< label Share_the_same_name >

The information showing whether the data on the flash disk is transferred in the C drive

< label folder_in_Cdrive >

< label file_in_Cdrive >

The sign of whether the data is selected

<label Folder_to_transfer>

< label File_to_transfer >

The dependencies of the observables are as shown below:

A_Share_the_same_name **is** ((File[3] == 0) && (Folder[3] == 1 && Folder[1] == a)) || (File[3] == 1 && (File[1] == b || (Folder[3] == 1 && Folder[1] == a))) || (Folder[3] == 0 && File[3] == 1 && File[1] == b) || (Folder[3] == 1 && (Folder[1] == a || (File[3] == 1 && File[1] == b))) ? "fill=solid,color=red" : "fill=solid,color=gray";

A_No_enough_space **is** Folder[3] * Folder[2] + File[3] * File[2] > Freespace ? "fill=solid,color=red" : "fill=solid,color=gray";

A_file_in_Cdrive **is** d[5][4] == [490,120,550,160] ? "fill=solid,color=red" : "fill=solid,color=gray";

A_folder_in_Cdrive **is** d[6][4] == [380,120,450,170] ? "fill=solid,color=red" : "fill=solid,color=gray";

A_Folder_to_transfer **is** Folder[3] == 1 ? "fill=solid,color=red" : "fill=solid,color=gray";

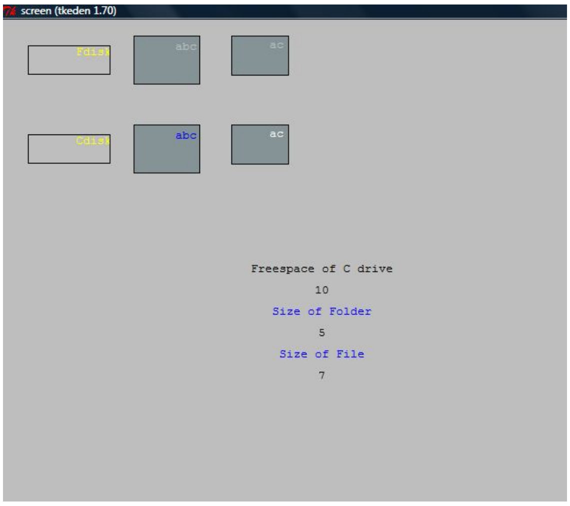
A_File_to_transfer **is** File[3] == 1 ? "fill=solid,color=red" : "fill=solid,color=gray";

The dependencies shown above are written in the form of EM standards. However, there also have the dependencies which is realised and formed in the form of procedural programming language, which is not expected before but can really realise the functionality of the model. **The sign to tell the user either the file or the folder is successfully transferred is determined and defined the action of moving of the window w2 and w3**, which represents the data originally on the flash disk in a vivid way.

## 3.2 Actions for agent and status of the model

The initial status of the model (figure 1):
On the image of the model as is shown on figure 1, it tells that there is a larger box means “Folder” and the smaller box means “file”, with their names shown on the box, at both the location of flash disk and C drive as is shown on the boxes with yellow tabs “Fdisk” and “Cdrive”. Also, the sizes of the

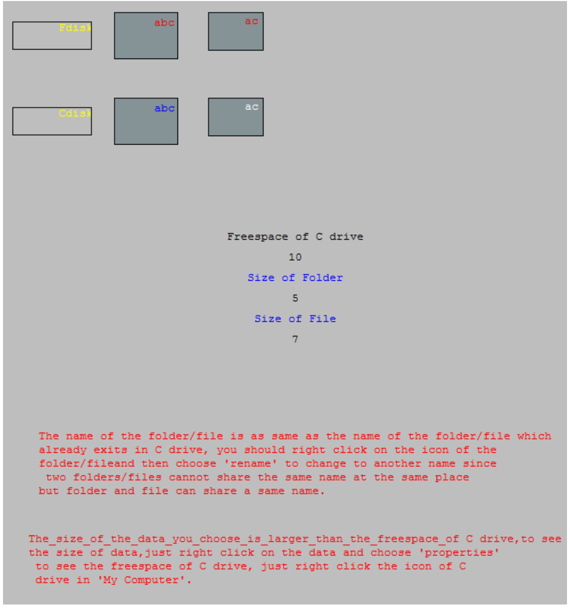
data to be transferred and the free space of C drive are shown too.



```
Freespace of C drive
10
Size of Folder
5
Size of File
7
```

Figure 1: initial state

To start, from the figure 1, we learn that currently no data has been selected, and the file on flash disk and C drive is named "ac" and the folder on flash disk and C drive is named "abc". The current frees space is 10 and the size of the folder and file to be transferred is 5 and 7. Use the mouse to click on the data on the "Fdisk" and you can see figure 2.



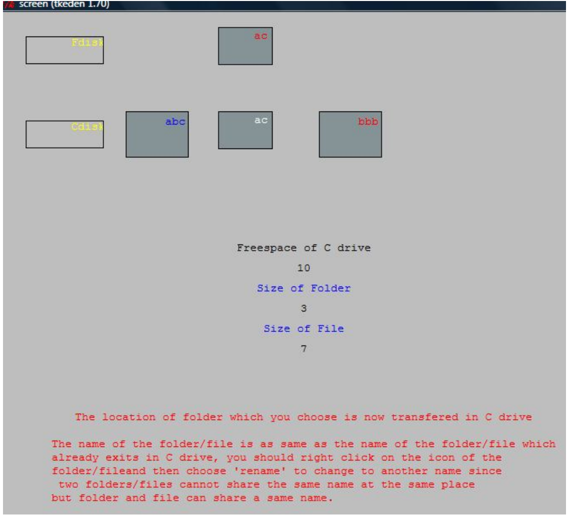
```
Freespace of C drive
10
Size of Folder
5
Size of File
7

The name of the folder/file is as same as the name of the folder/file which
already exits in C drive, you should right click on the icon of the
folder/fileand then choose 'rename' to change to another name since
 two folders/files cannot share the same name at the same place
but folder and file can share a same name.

The_size_of_the_data_you_choose_is_larger_than_the_freespace_of C drive,to see
the size of data,just right click on the data and choose 'properties'
 to see the freespace of C drive, just right click the icon of C
 drive in 'My Computer'.
```

Figure 2: not able to transfer for no enough space and share the same name

As the warning and solution which is shown up in the red words, you cannot transfer the data which you choose (data changed in red colour means you select it). To follow the instructions, try to change the name of the folder or file (defined as a list which is [name, size, state of being selected]). To rename the folder and change its size to a smaller one, input <Folder[1] = "bbb"; Folder[2] = 3;> and click the folder again, you can see figure 3. The way to transfer file from Fdick to Cdrive is just similar (figure 4).



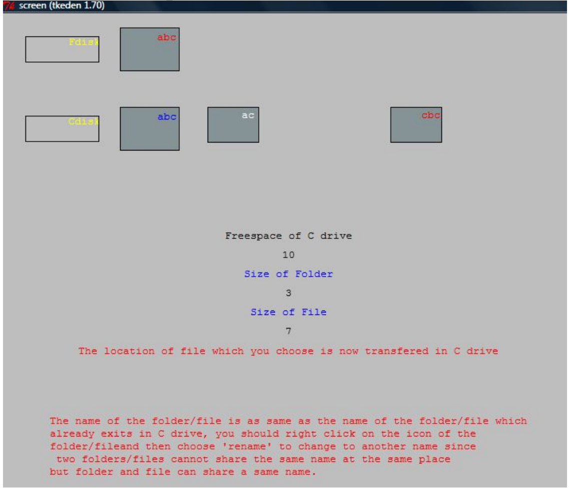
```
Freespace of C drive
10
Size of Folder
3
Size of File
7

The location of folder which you choose is now transfered in C drive

The name of the folder/file is as same as the name of the folder/file which
already exits in C drive, you should right click on the icon of the
folder/fileand then choose 'rename' to change to another name since
 two folders/files cannot share the same name at the same place
but folder and file can share a same name.
```

Figure 3: Folder with a new name bbb has been transferred but not the file ac

The folder bbb is transferred in C as is shown on the position and expressed in the words.

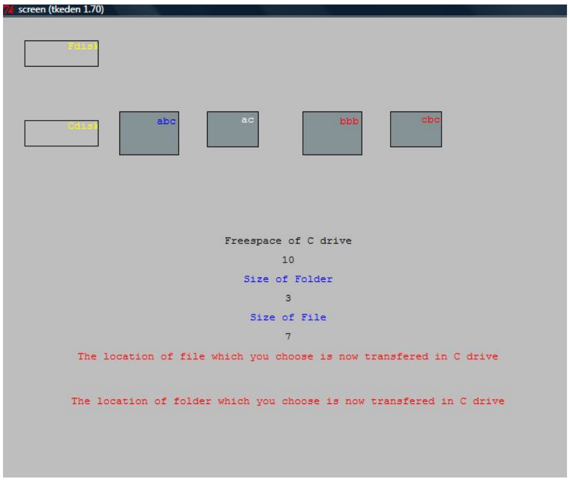


```
Freespace of C drive
10
Size of Folder
3
Size of File
7
The location of file which you choose is now transfered in C drive

The name of the folder/file is as same as the name of the folder/file which
already exits in C drive, you should right click on the icon of the
folder/fileand then choose 'rename' to change to another name since
 two folders/files cannot share the same name at the same place
but folder and file can share a same name.
```

Figure 4: File with a new name cbc has been transferred but not the file ac

The file cbc is transferred in C as is shown on the position and expressed in the words.



```
Freespace of C drive
10
Size of Folder
3
Size of File
7
The location of file which you choose is now transfered in C drive

The location of folder which you choose is now transfered in C drive
```

Figure 5: data transferred successfully

Folder with new name bbb and new size 3, and file with new name cdc are transferred on C as is shown on the position of image and expressed in the two lines in red words.

However, you can try to configure such items **in any patterns you like, which can affects in the dependencies of each observables**, to see what effects would be (note that **you must either use the mouse to click and see the colour turns to red or change the item Folder[3] and File[3] to 1 to ensure that they are selected**)

Folder[1], File[1]
Folder[2], File[2]
Folder[3], File[3]
Freespace
a(represents the name of folder earlier exited on C),
b(represents the name of file earlier exited on C)

## 4 Evaluations and Conclusion

The most significant evaluation is that the structure of the dependency should be improved by adopting another notation like eddi to construct the dependency of the observalble w2 and w3, which represents the location of the appearance of the data to be transferred. Actually I have got the idea of the dependency relationship for w2 and w3:

1. Use ?d; to see the value of display d
2. Define the d and its values in eden
3. After doing the steps above I found d is a list of [w7,w6…w2, w], and each w is also a list contains sub-list and values, so just write in this way: d[5][4] is Folder[3] == 1 && Folder[1] != a && (Folder[2] < Freespace || Folder[2] + File[2] < Freespace) ? [ 490,120,550,160 ] : [ 260,20,320,60 ];

   d[6][4] is File[3] == 1 && File[1] != b && (File[2] < Freespace || File[2] + Folder[2] < Freespace) ? [ 380,120,450,170 ] : [ 150,20,220,70 ];

However, such codes are illegal currently, so I need to try to gain another way in the further research after this coursework. But at least, the modelling of the process of thinking of how to transfer data in the usual case and the meaning of each step in transfer the data successfully are realised by implementing the knowledge of EM. This model has also supposed a particular case for supply enough instructions to the user who do not know how to transfer data at all. The limitation is also laying obviously on the displaying way of the response of the computer in accordance with the human reaction and then richness of the script of ODA. These should be involved in the further research after this coursework.

## Acknowledgements

During the modelling process, I've gain lots of help and effective suggestions from Dr Meurig and Dr Steve, from the thinking stage of the modelling, then to the stage of junior work which contains the procedural programming to realise the construction of dependency, until the final outcome which make some improvements on the ODA and which I can gain a better understanding of "making sence" and "making decisions".

Weighting: Paper 40%, Model 60%

## References

Beynon, W. M., Cartwright, R. I., Rungrattanaubol, J., Sun, P. H. (1998, September). *Interactive Situation Models for Systems Development*, Research Report CS-RR-353, Department of Computer Science, University of Warwick

Beynon, W.M., Boyatt, R., Ward, A. (2003, December) *The dtkeden manual – distributed communication features for tkeden* [Online]. (URL http://www2.warwick.ac.uk/fac/sci/dcs/research/em/software/eden/dtkeden/#usingdtkeden). (Accessed 14 May 2009).

Warwick web source, 'SCOUT - a definitive notation for SCreen layOUT', URL<http://www2.warwick.ac.uk/fac/sci/dcs/research/em/notations/scout/>, available at DCS of Warwick University(Accessed 30 Nov 2006)

Warwick web source, 'DoNaLD - a Definitive Notation for Line Drawings', URL< http://www2.warwick.ac.uk/fac/sci/dcs/research/em/notations/donald/>, available at DCS of Warwick University(Accessed 30 Nov 2006)