Lift Simulation in Empirical Modelling

HANG TIAN

1052482

Abstract

At the beginning of this paper, the definition and basic features of Empirical Modelling will be introduced along with the real world, and then discuss the actions and function of Lift Simulation in Empirical Modelling specifically. After that, talking something about the fair evaluation of this model, which consists of two sub-parts that the overall evaluation of the Lift model and the possible further expansion of the Lift. In section 4, the comparison between Empirical Modelling and conventional programming languages is given. What in the last part of this paper is the conclusion.

1 Introduction

After several years as being a taught module of the department of Computer Science in the University of Warwick, Empirical Modelling has proved to be a useful approach in simulation and programming. Due to the advantages of Empirical Modelling, programmers can simulate models in a more efficient way than using some conventional programming languages.

The purposes of this simulation using Empirical Modelling can be classified into the following two. Firstly, it is to analyse the strong points and the weak points of this new approach as well as compare with other conventional languages. Secondly, it can be used as an experimental example for teaching this course.

1.1 What is Empirical Modelling?

Empirical Modelling (EM) is a relatively new programming language to create interactive environments, which is developed extensively at the University of Warwick. EM mainly focuses on the human-centred construction of models, which consist of the elements of observation, dependency and agency that are encountered in every-day experience.

Generally, it is possible to assign a value to a variable based on the value of other variable in a program in conventional programming languages. While, troubles will occur if these variables change their value without changing the dependent values manually by the program, so this process can be treated as a factor that affecting the speed of the program as well.

In terms of issues like synchronization and instantaneous changing, EM is totally different, the EDEN interpreter makes variable changing through the whole program and updating the dependencies automatically at the same time when other relative variables are changing.

1.2 How dose EM relate to the real world?

Obviously, various models based on the real world can be simulated using Empirical Modelling that is due to the practical essence of this new programming language, as will be demonstrated with the Lift Simulation in Empirical Modelling.

Since the program can propagate the dependencies automatically, the users just have to input the piece of code including changing variables into the tkeden command window rather than executing the whole program files if she or he wants to observe some

changes, on the other hand, for programmers, it is easier to modify the simulation when necessary.

2 The Lift Model

In this part, simulation of the lift model will be talked about, including the introduction of the model and the main function achieved by using EM, rather than in a tame describing way, some details are represented in a discussion.

2.1 Introduction to the Model

This paper is by no means a report on the Lift Simulation in Empirical Modelling itself. Instead it mainly focuses on the utility of simulating the real world using Empirical Modelling.

It is, however undoubtedly necessary for readers to come to understand some major points and features of this model before discussing it in full details.

The aim of the model is to illustrate the dependencies between the lift and the control buttons as well as the changing values. After doing this model, some practical experience should be summarized as a side-objective as well.

2.2 Functions of the Model

The main function in this model can be generally classified into 3 categories: buttons control the location of lift, buttons control the number of person in the lift and location of lift and the person in lift can influence other variables.

Firstly, the function of some certain buttons and the action of lift will be demonstrated as follows in Figure 1.

When press one of the buttons shown in Figure 1, the lift will move to the corresponding floor by calculating the changing position of the lift, and then compare it with the location of the coming floor, if the y-coordinate of the upper-left corner is greater than that of the wanted floor, then it goes up to the coming floor, other it goes down to that one, Figure 2 shows the position relationship between the lift and this 5-floor building.

When one of the buttons was pressed, its colour changes from green, the initial colour, to black once, and will keep being black until the lift reaches to the wanted floor.

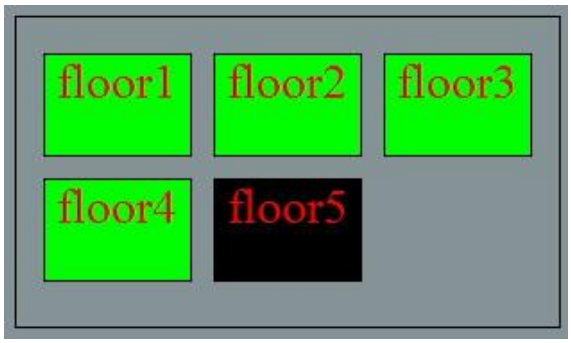

Figure1. Buttons control the movement of lift

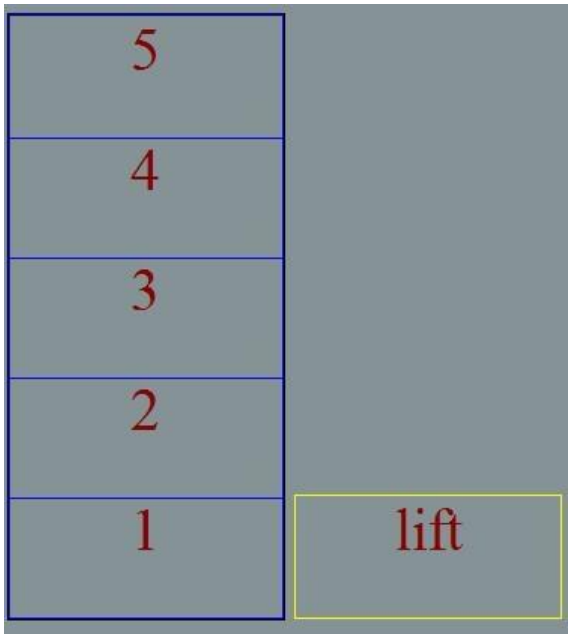

Figure 2. Position between the lift and 5-floor building.

An important function should be mentioned is the pro gotof :counter. The parameter counter is initialized to be 0 and another significant clause is edenclocks= [[&counter ,30]], which is used to control the speed of the lift. The function pro gotof can implement the movement of lift through calling the parameter counter and other relative functions as the function body.

Secondly, Figure 3 shows the buttons control the number of person in the lift, and the number of person can also influence the energy saving to some degree will be illustrated by Figure 4.

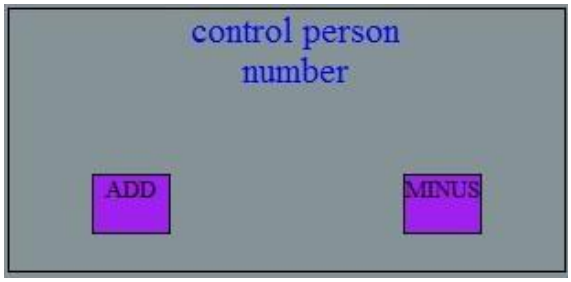

Figure 3.Buttons control the number of person in lift

Figure 4.Current state of the lift

When press the "add" button once, the number of person will plus one in the lift, and it will minus one when the "minus" button is pressed, the colour of the alarm will be red if the number reaches to 10 and then the add button is not available anymore, when the number is zero, the minus buttons goes to be unavailable either, so the number doesn't change any more. Under other circumstances, it's gray. The value of energy-save is related to the number of person in lift according to the equation res_ener_save=40-0.5 * res_capa.

Finally, the lift stops at the certain floor, and the floor number will be shown in the displaying area.

3 Evaluation

This section outlines the evaluation of this EM model in the following areas. Firstly, how well it achieves the goal of modelling a lift is regarded as an overall evaluation of this EM simulation. Secondly, the possible further expansion of this model will be discussed and detailed, which is a helpful way to understand Empirical Modelling more clearly.

3.1 Evaluation of the EM model

The basic functions of the lift have been done using EM in this model, even maybe there are some criticisms of the EM that it not only oversimplifies the scenario, but also overlook some objective factors of lift, while ,to be honest, this model also can be regarded as an easy example for beginners to learn EM as well. It is modelled in simple functions and a clear structure, which is easy to be understood.

3.2 Possible further expansion of this model

While this is currently a small sized model, there is much scope for further expansion. Firstly, it is obvious that there is no door of the lift and another function can be implemented is that several seconds later after passengers pressing the floor-buttons, the door should close automatically, and then move up or down.

On the other hand, if lift reaches the wanted floor, after several seconds, the door should open automatically. All this kind of time delay can be seen as the safety of the passengers.

Another expansion is that if two buttons are pressed at the same time, then the lift should stop at the midway, this will be more similar to the real lift in our daily life.

If all of the problems mentioned above can be solved, then this simulation is closely following with the main point of EM, which is that EM relates much more to the daily life of human beings.

4. EM approach in comparison to conventional languages

As a comparatively new programming language and a modelling approach, it is essential to bring in the merits of previous approaches and then refines itself so as to make more contribution to the field of computer science.

In this section, the advantages and disadvantages is discussed in the first two sub-parts, following that is the future about EM, and the main advantages of EM is emphasized and should be implemented.

4.1 Advantages of EM

Obviously, there are many advantages of Empirical Modelling. While certain aspects of the EM tools are not particularly mature. During simulating this model, the most important advantage should be

highlighted is that, the dependencies can change according to the related variables automatically without doing it manually. Dependency is really a powerful concept, which can be utilized to reduce the extra effort in order to maintain synchronisation between observables.

A prototyping programming language is the further advantage of Empirical Modelling, in which, definitions are refined and dependencies are updated automatically. Additionally, as mentioned above, if some variables change, user can just input the piece of changing code into the tkeden command window rather than execute the whole files to observe the change.

4.2 Disadvantages of EM

Just as every coin has two sides, Empirical Modelling and the modelling tools have their drawbacks. Immaturity must be the biggest problem of this language and the modelling tools. Bugs in the interpreter can cause suddenly crashes and other unexpected events, especially sometimes the error messages are too fuzzy for programmers to settle the problems.

4.3 Future of EM

By analysing the merits and demerits of Empirical Modelling, it is sound to say that EM is an approach that is worth being using broadly. Despite that there is still a long way to go. The interactive nature makes EM used in many fields, such as, according to current research that EM is involves with the education as a tool. Additionally, the dependency in programming is so important and useful, so it is hoped that this merit will be used in other conventional programming languages.

5 Conclusions

After accomplishing the simulation of the lift model, what can be easily reached is that the more someone experienced, the more will be obtained, in particular for learning a new programming language.

In a word, Empirical Modelling is a very useful tool when to develop a model, such as to simulate the simple model Lift Simulation. EM has been regarded as a tool which provides more effective support for programming and modelling than other conventional languages. The automatically updating dependencies and its interactive nature make it easy to be adopted despite there are some problems in either EM notations or developing tools. It's my hope that EM can be applied in a wider range and the demerits of it will be eliminated in the future even there is still a long way to go.

Acknowledgements

I would like to thank all of the teachers who teach us Empirical Modelling as a useful programming language and all of my classmates who help me to solve the problems occur when doing this coursework. Additionally, thank all the authors who write the useful paper which we can refer to as a reference.

References

W. M. Beynon. Radical Empiricism, Empirical Modelling and the nature of knowing. 13(3):615-646, December 2005.

R.Boyatt, A.Harfield ,M.Beynon. Learning about and through Empirical Modelling. Advanced Learning Technologies, 2006. Sixth International Conference.

Online materials from University of Warwick> Computer Science > Empirical Modelling: http://www2.warwick.ac.uk/fac/sci/dcs/teaching/material/cs405/

Meurig Beynon. Visualisation using Empirical Modelling principles and tools. http://www.viznet.ac.uk/cross_domain/VisWS.pdf