

Interactive Empirical Model Design Simulation of Aircraft

0756981

Abstract

This paper explores the interactive simulation environment design with the definitive notation and the interactive principles in the empirical modelling. Constructing virtual environments which simulate the objects and behaviours in the real world is one of the important researches in the computer science. Empirical modelling attempts to build truly open and experiential interactive environments. Different from the traditional approach, the empirical modelling exploits the computer for pre-theory experiment which uses experience to model unknown interactions and un-well understood situations. Definitive notation is the formal language in which to describe the model. In the following paper, the design of the interactive simulation environment with the empirical modelling and the definitive principles will be significantly investigated with a demo for the simulation of aircraft.

1 Introduction

The purpose of this paper is to investigate the design of the interactive empirical model with the definitive notations.

1.1 Background

Empirical Modelling (also known as EM) is an approach of modelling based on the concepts of observation, dependency and agency. The EM attempts to construct accurate computer models that reflect observations, dependencies and agencies for continuous experimental interaction. The interactive environments built by the EM are truly open and experiential. The EM exploits the computer for pre-theory experiment which uses experience to model unknown interactions and un-well known understood situations. Definitive notation is the formal language that the EM used to model the real world phenomena. The interpreter of the definitive notations is tkeden which supports a variety of definitive notations and runs in several platforms. The empirical model investigated in this paper is built with three different definitive notations, EDEN, DoNaLD and SCOUT. EDEN is a general-purpose language that supports the concept of definitions. DoNaLD is a definitive notation for 2D line drawings. SCOUT is a definitive notation for describing screen layout. The tkeden is EDEN notation interpreter, which converts the definitions written in DoNaLD and

SCOUT notations to EDEN definitions before being interpreted.

1.2 Definitive notations for interaction

Many interactive human-computer interfaces can be formulated clearly and conveniently using the definitive notations which can be considered as notations based upon sequences of definitions. Such notations are regarded as implicit ingredient in the spreadsheet packages. The human-computer interfaces such as dialogues have relatively simple kind. Typically, the dialogues use some generalisation or a similar spread-sheet. The definitive notations are considered as a suitable medium for the dialogues when the dialogues are described as sequences of human-computer or computer-human transactions which have common points of reference. In this paper, the term interaction means as its narrow sense which is dialogues in which many transactions occur.

1.3 Definitive principles

The emphasis of the definitive programming notation is upon interaction. Generally, the principle is to use design process to describe drawing of the diagram when the intended semantics of the diagram has an important role, and an appropriate framework is provided within the design of the conveniently

represented dialogue. Essentially, definitive notations are represented as underlying algebra notations which consist of data types and operators together with a set of typed variables. The values of the typed variables can be defined implicitly by formulae constructed with other variables, or simply given explicitly. Each stage of the design process is then represented in the context of a graphics notation which is a suitable set of variable definitions used to describe the abstract structure. Scalar values for the definitions of fields in a spreadsheet define the abstract relationships.

2 Simulation of Aircraft

2.1 Motivation

Simulation of Aircraft is an example for the design of the interactive graphic environment with the EM. This empirical model simulates the simple behaviours of an aircraft. These behaviours represent the simple principles of flying for the aircraft. The EM model provides an interface that allows the simulated aircraft to be interacted for the investigation of its behaviours. The EM model simulates the following situations for the aircraft:

- When the aircraft is ready to take off, the indicator show full fuel in green
- When click take off button, the aircraft takes off and air flow is shown after the wings in normal length
- When the aircraft is in the air and click turn left button, the right flap will open to block the air and the left flap keeps close, the tail turns right, the air flow shown at left side is longer than right side indicating the resistance of air flow at right side is larger than left side so that the aircraft is turning left. The situation for the aircraft turning right is opposite.
- When the aircraft is landing, both the left and the right flaps are open to slow down the speed of the aircraft and the tail should be straight position.

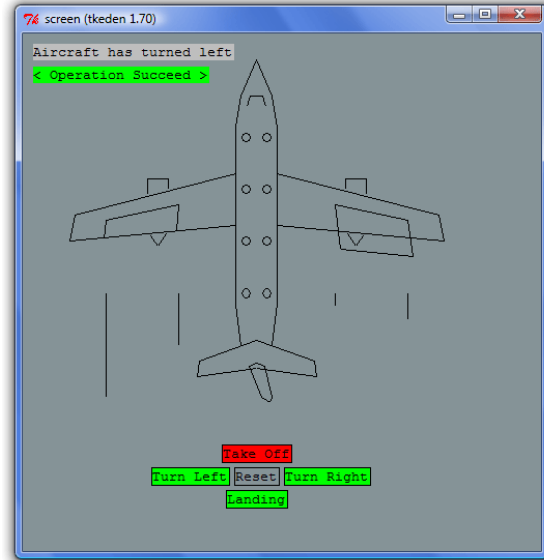
The EM model simulates the behaviours of the aircraft for the real world environment so that for some situation the operations on the aircraft are not permitted.

- When the aircraft is flying which means the take off button has been clicked, the take off operation can not be activated again, and warning is on.
- When the aircraft is landing, all of operations are not permitted.

In the EM model, the invalid operations are displayed in red colour, otherwise green. If the operations in red colour are performed, the warning is on in red, otherwise green.

After the aircraft landed, to restart the model the reset button must be clicked to clear the current status.

Figure 1: EM for Simulation of Aircraft



2.2 Behaviours of aircraft

The empirical model for the simulation of aircraft implements the behaviours of aircrafts described in the following logic tables.

Table 1: Logic for simulated aircraft

	Left flap	Right flap	Tail	Air flow
Take off	Close	Close	Straight	$L = R$
Turn left	Close	Open	Right	$L > r$
Turn right	Open	Close	Left	$l < R$
Landing	Open	Open	Straight	$l = r$
Reset	Close	Close	Straight	Invisible

Table 2: Logic for A/D

	Take off	Turn left	Turn right	Landing	Reset
Take off	Deny	Deny	Deny	Deny	Allow
Turn left	Allow	Allow	Allow	Deny	Deny
Turn right	Allow	Allow	Allow	Deny	Deny
Landing	Allow	Allow	Allow	Deny	Deny
Reset	Allow	Allow	Allow	Allow	Allow

3 Empirical Model Design

3.1 Observables in model

Observables are the entities of the situation being modelling. The identity of an observable can be established through experience and its current state

can be captured by experiment. The observables are the essential items in the EM. The EM forms dependencies with two observables, and the agents in the EM are also based on the observables. The changing dependencies driven by the agents are also observables. In the simulation of aircraft, the observables are such as the flaps of the aircraft, the tail of the aircraft, etc.

3.2 Dependencies in model

A dependency represents relationship between two or more observables. Dependencies are used to express the affection for the values of observables with the change of one or more observables. The dependencies between observables acting in the situation are experienced by the modeller. The dependencies connect different observables of model so that the dependencies are complicated interactive systems different from a group of individual parts. In the simulation of aircraft, the dependencies are such as the relationship between flaps and tail, the relationship between flaps and air flow, etc.

3.3 Agencies in model

An agent can be considered as a special observable which is able to change the current status of observables. If observables and dependencies are the objects for the empirical model, the agencies make the objects interactive for each other. The empirical model is active because of the use of the agencies. In the simulation of aircraft, the agencies are such as the operations to turn the aircraft left or right, to make the aircraft taking off or landing, the flaps which change the intensity of air flow, etc.

3.4 Model Design with Notations

The design of the empirical model for the simulation of aircraft uses the three definitive notations, DoNaLD, SCOUT and EDEN which are mentioned in the previous paper, therefore the definitive script consists of three main sections which are identified with %donald, %scout and %eden. In the DoNaLD section, the shape of the aircraft is drawn with a set of lines and a set of circles. Additionally, the air flow is drawn with a set of lines, but the air flow is invisible when the interpreter initialized the definitive script. The left flap, the right flap and the tail of the aircraft are the three collections of shapes drawn with the definitive notation openshape. These three collections of shapes are responsible to simulate the behaviours of the aircraft together with the effects of the air flow. In order to control the actions of the flaps and the tail, there are five variables are declared in the EDEN section, flopen, fropen, tt, airl, air. The flopen and fropen are represented with two

values 0 and 1. Any of them with value 1 will cause the shape of the corresponding flap changed, which means the flap is open, otherwise value 0 causes the flap close. The variable flopen is for left flap and the fropen is for right flap. The variable tt is for the tail and it has three values -1, 0 and 1, which makes the tail turn left, be straight and turn right. The airl and airr are float numbers, which indicates the length of the air flow. In the SCOUT section, the buttons and texts used in the empirical model are formed. In the EDEN section, the trigger functions for the buttons formed in the SCOUT section are defined. The logic for describing the behaviours of the aircraft is implemented in this section. The behaviours of the aircraft is controlled by changing the values of the variables discussed in the DoNaLD section together with the rules described in the simulation of aircraft section.

4 Conclusions

This paper investigates the design of the interactive graphic environment with the empirical modelling. The paper has introduced the background of the EM, which attempts to construct open and experiential interactive environment based on the concept of observation, dependency and agency with the definitive notations which is interpreted by tkeden. The definitive notation for interaction and the definitive principles in the EM are introduced. An example of the interactive empirical model is investigated and explained. The terms of observable, dependency and agency are introduced and illustrated in the example. The investigation in this paper proves that the EM is capable to build truly open and experiential interactive environment for pre-theory experiment.

Acknowledgements

Thanks to Meurig Beynon and Steve Russ for their feedback and articles.

References

Meurig Beynon. Definitive Notations for Interaction.

Meurig Beynon. Definitive Principles for Interactive Graphics.

Weighting: 5 : 5

R.A. Earnshaw. B. Wyvill. New Advances in Computer Graphics.

M. Beynon, S. Russ, W. McCarty. Human Computing – Modelling with Meaning

M. Beynon. Empirical Modelling as a platform for non-theory based musical analysis

Beynon W.M. Software Construction Using Definitions: an illustrative example.

Russ S. B. (1997) Empirical Modelling: the computer as a modelling medium.

W.M. Beynon. Radical Empiricism, Empirical Modelling and the nature of knowing.