A 3D simulation case study of airport air traffic handling

Henk de Swaan Arons
Erasmus University Rotterdam
PO Box 1738, H4-21
3000 DR Rotterdam, The Netherlands
email: hdsa@cs.few.eur.nl

Abstract
Modern Windows-based simulation packages bring simulation within reach of decision-makers. The use of graphics enables the manager to observe an animation of the simulated reality, to focus on the essentials of the model without the need to bother about implementation details. In most applications a 2D representation will satisfy the simulation objectives. One field of application in which 3D simulation is highly recommended is air traffic handling on and above an airport. In order to get a real understanding of this kind of process 3D simulation is indispensable. In this article a simulation is described in which 3D animation is essential to base management decisions on.

Keywords: discrete simulation, 3D simulation, interactive programs, animation software, air traffic handling

1. Introduction

In many fields of application simulation is an accepted tool to support management decisions. In recent decades simulation was restricted to experts in computer science because in these days simulation required many programming efforts. In the last few years a new generation of simulation software programming environments have emerged. These packages allow domain experts to design, develop and process their simulation model by using graphics elements such as icons, drawings etc. One interesting characteristic of these simulation languages is their ability to present to the end user an animation of the simulated system. This allows the end user to observe a simplified on screen version of the actual system. For decision-makers this is a great help to understand the processes displayed, to track shortcomings and to suggest improvements.

In most applications a 2D representation will satisfy the simulation objectives, but there are examples that only a 3D simulation will provide the necessary information. The air traffic handling is such an example which is basically three dimensional. On the ground aircraft are moving from hangars or platforms to the runway and vice versa. In the air aircraft are leaving from the airport or approaching it through a circuit and possibly stay there temporarily in a hold position (in fact in a queue). In order to follow these two processes a (pseudo) 3D simulation provides the necessary information.

In this article a case study is carried out to an airport with only one runway. One objective is to find out whether such a 3D simulation model with its associated animation can help to support decision-making. A related objective is to determine whether a 3D simulation helps to better understand the traffic processes. Furthermore, it can assist to find answers to several questions concerning the capacity of hangars and runway when air traffic intensity increases.

2. Rotterdam Airport

The airport of Rotterdam is the second largest airport in The Netherlands lying North-West of Rotterdam with Schiphol as largest. In 1993 there were about 38126 cross-country flights and 77068 local flights compared to 277661 and 17896 for Schiphol. Cross-country flights are mostly made up of commercial flights with big aircraft. Commercial flights consist of scheduled flights (ca. 80%), charter flights (ca. 20%) and freight (ca. 3000 tons). Local flights are pure training, private or touring flights in small propeller-driven aircraft. In 1993 the kind of traffic at Rotterdam was more concentrated at small aircraft. The trend for the coming years is a shift from small aircraft to the bigger aircraft doing commercial flights.

The airport has one runway that can carry aircraft up to 200000 kg AUW (All-Up-Weight, i.e. full tanks + passengers/cargo). The runway is too short to support aircraft that are heavier. The maximum capacity of the runway lies around the 46 flights per hour.

2.1 Air traffic procedures, general theory and practice

There are two types of flights, VFR (visual flights rules) and IFR (instrumental flights rules). This kind of flight require different traffic procedures. With IFR, a pilot is flying purely on instruments, neglecting the outside world. The pilot flies on radio beacons and by guidance of radar. In this case the pilot does not have to know visual marking points such as a tower or a bridge to navigate on but navigates purely on its heading. The opposite is true for visual flights. When flying visual, a pilot checks his heading but also determines his position based on visual marking points. Roughly speaking, small aircraft fly mostly VFR (under appropriate weather conditions) and the bigger aircraft such as Fokker 27, Airbus or Cessna Citation fly IFR.

An aircraft that wants to land, first has to enter a circuit above the airport. The aircraft that flies VFR first enters downwind to do the checks that are associated with it (checking the windbag to see the wind direction) then turns to do the baseleg approach and finally turns to final approach to come in front of the runway. Commercial aircraft that fly instrumental make a straight in approach on final to land. The way the circuit is flown depends on the wind direction. Suppose the wind goes from west to east. Aircraft take off and land with the nose in the wind, so in this case it is a left hand circuit. In the opposite case, when the wind comes from the east, the circuit would be the other way around. When the wind makes an angle of 90° with the runway, both circuit rounds can be used.

When traffic around the airport is very heavy, air traffic control can place aircraft in the air in a holding. Holdings can consist of stacks or holdings with a single aircraft. Stacks are formed according to the FIFO method. An aircraft that enters the holding first is at the minimal altitude and each following aircraft that enters it, is placed in the holding with a certain vertical separation on top of it. So, each time an aircraft leaves the holding the stack falls down until the highest aircraft is at minimal altitude. Holdings are also formed on the ground. Aircraft can wait on the platform or in holding before the runway.

Air traffic control of Rotterdam has a control zone (CTR: Controlled by Tower and Radar) with a radius of 10 nautical miles until 3000 feet Minus Sea Level and is 24 hours operational. Within the control zone, pilots have to report when they arrive at certain points or when they file special requests. The IFR flights are conducted at 2000 feet and VFR at 1000 feet in the control zone. This for air traffic reasons.

Concerning the runway, 06 and 24 are in use. When the wind direction varies between 150° and 330° then runway 06 is in use and the circuit is left-hand. When the wind direction varies between 330° and 150° the 24 is in use and the circuit is the other way around. Rotterdam Airport has two circuits. A training circuit at 500 feet for pilots who want to practice touch and go landings and the normal circuit at 1000 feet for full-stop landings. The training circuit has two holdings (Alpha and Bravo). The reason for this is that aircraft flying commercially have priority to land. The normal circuit has no holdings but there are some other points within the control zone marked for eventual holdings before entering the circuit. This will be discussed later on. The maximum number of aircraft in both circuits is set at two per circuit. For IFR flights, there are two holdings beginning at 2000 feet. These holdings can be stacked. Final approach length is about 10 nautical miles and is longer than for VFR flights.

There are some relevant air traffic procedures. The minimal time between an aircraft that wants to take off (take off lasts about 30 seconds) and an aircraft that wants to land is about 3 minutes. If the time difference is smaller, aircraft at the ground have to hold until the runway is vacated. Runway 24 has an ILS (Instrument Landing System) area. Air traffic control can instruct VFR flights to stay clear of that area when used by IFR flights. Wake turbulence causing a problem when a big aircraft is in front of a small aircraft, results in a certain time between the aircraft. If more than two aircraft want to enter the normal circuit, they have to wait at certain points until one aircraft has landed. Aircraft for the training circuit have to wait on the ground until the circuit comes available to them. These topics are discussed in more detail later on. Maximum speed within the control zone is 250 knots for jets and turbo-prop aircraft and about 180 knots for smaller propeller-driven aircraft.

3. Simulation with Service Model

ServiceModel [ServiceModel, 1995] is a Windows-based simulation software package developed specifically for service systems. ServiceModel has a broad range of applications. Some of the more common uses include: customer service, office automation and distribution & warehousing.

3.1 Modeling elements

A system consists of several elements. A typical business system consists of various types of *activities*, which transform *inputs* into distinct *outputs*, in order to produce value to the end user of the product or service. These activities are subject to the system *environment*, including the constraint of scarce *resources* and *interrelationships* between events. In ServiceModel, the activities discussed before are called processes, inputs and outputs are called entities, and the scarce resources are simply resources. All processes must occur at a location. Locations are either fixed, physical places in the system where activities occur, or they can be imaginary places where entities enter simply to perform some logic and then leave.

3.2 Modeling in ServiceModel

The process of modeling in ServiceModel is build up around the graphical representation of the service system that one wants to simulate. Modeling in ServiceModel starts with placing the background graphics of the service system one wants to model in the layout-window. Background graphics may contain anything one wants to appear in the background of the animation. After that the following main elements can be added to the model:

Locations

The locations editor is used to place all locations on the layout. Special symbols and icons may be used to represent locations. When a location is created a record belonging to the location is also automatically created. A record contains information about capacity, number of units, downtimes, and queuing rules for each location.

Entities

The entities are defined in the entities editor. Entities may have multiple graphic icons to represent different states or forms of the entities or no graphic representation at all.

Arrivals

When the entity types are specified in the model one can specify when and where these entities arrive in the arrivals editor. Each arrival record defines information about the type of entity and arrival location, the number of entities, first arrival, frequency between arrivals, and logic associated with the arrival of each entity.

Resources

Resources can be people or anything else that has to be present for an operation or activity to take place. Resources are created in the resource editor. In the resource module, information can be defined among other things about the name, the downtime, and the number of units of the resource.

Processing

The processing logic defines everything that happens to that entity at each location where it is processed. In addition, the processing defines where an entity is routed to once all operations at a location are complete. The processing logic is defined in the processing editor.

Subsequently the following details can be added to the model:

- Variables
- Attributes
- Distribution functions
- Downtimes
- Shift schedules
- Path networks

Remarks about working with ServiceModel

The way in which ServiceModel is organized gives great support by building a simulation model. One can start simple and use the options from the build menu to add locations, entities, path networks, resources, processing and arrivals to the model. In this way one is more or less forced to use a structured approach to modeling: a very positive quality.

The statement builder helps one to create model logic. It is a menu-driven tool that guides one when one is creating a statement. When the statement builder menu appears one may select the desired logic element (statement, distribution, location, etc). One will then be prompted for the necessary parameters. The statement builder is accessible from many fields in ServiceModel. However, the elements listed in the menu will depend upon the nature of the field. This tool is an enormous help when one first starts using ServiceModel. When one is a more experienced user he can create statements without the use of the statement builder.

The runtime error-messages that are generated by ServiceModel provide a great help by debugging the model logic. When the runtime error-messages are not sufficient then one can use the easy to use ServiceModel debugger.

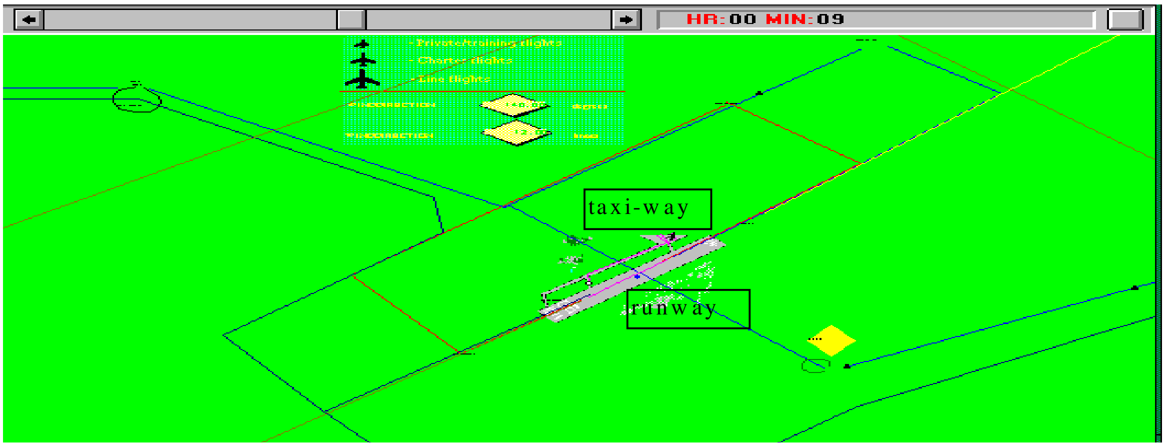
ServiceModel provides global or detailed output reports with statistics on among other things locations, entities, resources, and operations. Bar charts, pie charts and histograms can be created based on the output reports.

4. The simulation model

The figure shows the animation of the airport.

The system is bordered at the beginning of the control-zone (so a radius of 10 nautical miles). Aircraft can enter the system at the beginning of the control-zone or via the platform. They leave the system the other way around. There are five types of locations in the model, the first one is that of the arrival from aircraft at the control-zone, the second is the location at the platform where aircraft start their engines and move to the taxi-way. The third location is where aircraft can be instructed to wait on the ground (before lining-up at the runway) and in the air. The fourth location is the runway. There are some more locations, but they only serve as to support operational logic.

The routes that aircraft take are when moving in the control-zone, are divided in IFR/VFR and DEPARTURE/ARRIVAL and special path-networks for the training-circuit and the route aircraft take from the platform to the runway.



Airports gain the most from scheduled and charter-flights.

This because large volumes of passengers make use of the facilities of the airport and the heavier aircraft have to pay more landing fees. The chosen entities are based on these type of flights. There are three types of entities modeled:

1. Private/Training flights (VFR, 100 knots),
2. Scheduled flights (IFR, 250 knots),
3. Charter flights (IFR, 250 knots).

Since the first category constitutes a more random arrival pattern, for their arrivals a probability distribution is used. As a first approximation a negative exponential distribution is selected [Kelton and Law, 1991]. The second category is modeled via a fixed time-schedule. The last category is not modeled at the moment.

Concerning the runway, only 24 is modeled. This because for the problem to measure waiting time, it was not necessary to model two different circuit rounds.

Since wind is an important factor in the aviation, it is included in the model to calculate the speed change that is caused by this factor and the resulting time to overcome the distance. In this way, the effect of the wind on the results can be measured.

Naturally, air-traffic procedures such as *maximum of two in the circuits*, *priority of scheduled* *flights on final* and *aircraft in air have priority over aircraft on the ground* are modeled. Also at some supportive locations in the model, decisions to delay a certain aircraft are made to avoid that aircraft collide on final approach.

5. Experiments and results

There is a diversity of experiments that can be done with this kind of models. Examples are:

1. Increase the frequencies of all categories and measure the effect on waiting time, runway and hangar capacity.
2. Change wind direction/speed to see how it effects waiting time, also in relation to capacity problems.
3. Increase the frequencies and measure their effects on landing fees.
4. Increase the number of passengers and observe the effect on aircraft utilization, air-traffic intensity and landing .

6. Acknowledgements

The undergraduate student Ronald Ritsma has had an important contribution to this article. He prepared the research by providing the necessary literature and helped to speed up progress by his programming efforts. In the final stage he got assistance from Jeroen Filbri. The author acknowledges their assistance.

7. Literature and references

[ServiceModel, 1995] User manual ServiceModel, ProModel Corporation, Utah, US, 1995.

[Kelton and Law] Law A.M., kelton W.D., Simulation Modeling & Analysis, McGraw-Hill International Editions, 1991.