UNIVERSITY OF NOTTINGHAM

USING SIMULATION TO ASSIST RECRUITMENT IN SEASONALLY DEPENDANT CONTACT CENTRES

Leeanne May
4192573
MSc Management of Information Technology

USING SIMULATION TO ASSIST RECRUITMENT IN SEASONALLY DEPENDENT CONTACT CENTRES

ABSTRACT

The weather is unpredictable and can have a large impact on the profitability of seasonal businesses, particularly if staffing requirements are highly temperature-dependent. This dissertation has developed a what-if analysis tool using simulation methodology to assist affected SMEs in determining the best case scenario for timing hiring new staff and deciding the optimum length of temporary employment contracts. A boiler maintenance company was used as a case study and the objective to create a prototype of a tool that can be used by users with minimal statistical and modelling knowledge. Publicly available data on contact centre staffing was be used, along with any internal data which was made available by the company. The findings are that contract length could be used to improve meeting targets and a solution to show impact of weather simulated call volumes.

ACKNOWLEDGEMENTS

I would like to thank Peer-Olaf Siebers, my supervisor, for his assistance throughout this project and Simon Woolhead for helping me with MATLAB.

CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

The aim of choosing a subject for this dissertation was to attempt to solve a real world problem through the techniques and methodologies gathered from completing the MSc Management of Information Technology. I have access to a company which provides boiler maintenance and repair services, and was therefore able to work with the company to provide a technological solution to one of the challenges they face – staffing levels over the winter period, when it cannot be known in advance how severe the winter will be.

Payroll costs to companies can take up anywhere from 18-50% of operating expenses (SHRM, 2008). For businesses which must substantially increase their full time equivalent (FTE) members of staff during peak periods, it is important to get the timing of recruitment correct to avoid unnecessary costs. When the peak period is decided by something as variable as the weather, this decision is particularly difficult.

There has been a lot of work carried out in the area of contact centres and hospitals to determine the optimum number of staff needed based on the service requirements of the business. For example, Erlang formula calculators are available, which can estimate staffing requirements when provided with the amount of calls and average service time (Westbay Engineers, 2014). However, there is less research into what period of time is needed to achieve the staffing levels required. For example, it may take over a month to recruit and train a new starter and over 7 weeks before they are fully trained (CFA, 2012, p.39). If recruitment starts too early then staff will have increased idle time; recruit too late and the work increases faster than staff can cope, leading to complaints and lost customers. Since some businesses hire up to 3 times as many FTE at peak times, this can cause significant damage to business if misjudged (Myron, 2002).

Energy companies are heavily affected by the weather. Temperature is the main driver of gas use in the UK (CelsuisPro no date p.5) this means that the changes in temperature which come with the first cold week of the year are a huge strain on boilers and thus boiler repair companies. Academic work to date, which includes ambient temperature measurement of “degree-days” as a way to forecast consumption (Quayle and Diaz, 1980, p.246), has focussed on energy companies. I will apply the same principles to a boiler repair and service company on the basis that the gas-based central heating systems are the primary way energy customers heat their homes (Meier and Rehdanz, 2009, p.952).

This project will show that by using a discrete event simulation it is possible to build a tool which models recruitment and in-coming call processes sufficiently well for it to be used to investigate recruitment levels, interaction effects with winter severity, and also conduct experiments to suggest effective lengths of contracts for temporary staff.

2. LITERATURE REVIEW

2.1 INTRODUCTION

This review of previous work will begin with a discussion of call centre definitions and staffing tools, Human Resource processes, and then preparing for the impact of changes in weather, and an explanation of different simulation types. Current approaches combining the three areas will conclude this section.

2.2 CONTACT CENTRES AND THE ERLANG C FORMULA

INTRODUCTION TO CONTACT CENTRES

Around 1 million people are employed in the contact centre industry in the UK (CCA Global, 2014, p.5) . A call centre is the department (or group of departments) that handle customer enquiries. It may be the customer-facing operations department of a company, or a company on its own to which others outsource work (Cfa, 2012). Interactions with customers can take the form of phone calls, emails, electronic chat and post, all of which have their own assumptions and data requirements.

This project is focussed on a single department's interaction with customers through calls because it simplifies the requirements of the model. This means that other departments, such as Sales, can be excised from the scope of the model, streamlining the inputs and outputs needed.

A contact centre is judged by its ability to meet a set of Key Performance Indicators (KPIs). These alter slightly from company to company; however, they generally fall into the following categories:

Best of Best Practices

✓**G. Consider implementing a balanced scorecard…**

Key Performance Indicator	Weighting	Performance Range		Actual Performance	Metric Score	Balanced Score
		Best Case	Worst Case			
Service Level (% answered in 20 sec.)	30%	90%	65%	81%	64.0%	19.2%
First Call Resolution (%)	25%	80%	60%	63%	15.0%	3.8%
Abandon Rate (%)	15%	3%	10%	5%	71.4%	10.7%
Average Talk Time (seconds)	10%	180	360	255	58.3%	5.8%
After Call Work (seconds)	10%	180	360	345	63.9%	6.3%
Average Handle Time (seconds)	10%	360	720	600	33.3%	3.3%
Total	100%					**49.1%**

Figure 1 Summary of Best Practice (Scafaio, Slide 37)

The table above is from an American Government department emphasising good practice in call centre management and shows a balanced scorecard. All the measures are useful in slightly different ways. For example:

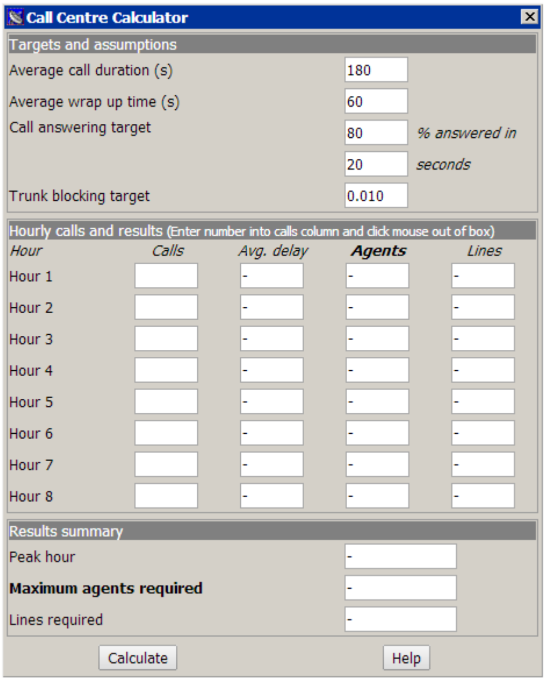
- a high First Call Resolution (FCR) should increase customer satisfaction, as they do not have to call in again, however it can have an effect on Average Handling Time
- Average Handling Time (AHT) is how long the advisor spends on the customer's query, and is the sum of the total time on call and total after call work. Longer talk times are a sign of more complicated calls, a high FCR or of inexperienced advisors.
- After Call Work Time is an indicator of other processes, which could involve the advisor writing up notes from the call or having to take action away from the desk. Longer times will impact the number of advisors available to handle calls.
- Service Level (% Answered in 20 Seconds) ensures that call queues do not get too long. A low number can be a sign that there are too many staff available, but too high and the level of abandoned calls will increase.
- Abandonment rate is the proportion of customers that terminate the call before speaking to an advisor. A high rate leads to customer dissatisfaction and further attempted calls.

A call centre manager will attempt to balance these indicators by matching the number of staff required to peak call times. Getting this prediction wrong can lead to customer dissatisfaction and increased stress for employees, but overstaff and the labour costs can be very high.

TYPICAL APPROACHES TO CALL CENTRE STAFFING

There are two common ways for call centres to predict staffing levels in house: spreadsheet models or software, often based on Erlang-C formula(A. T Ernst et al., 2004 p.11).

Spreadsheet models use historic data to predict staffing levels. Usually based in Microsoft Excel, the modeller will use assumptions such as the number of calls an advisor can take an hour and historic numbers of incoming calls to estimate the necessary number of staff. Excel can generate random numbers and some statistical distributions to improve the accuracy of the prediction. Spreadsheets are cheap, need a low level of skill to use effectively and are familiar to office workers; however, they are restricted in the ability to show the interaction of different elements, and can become very unwieldy in size and complexity (Robinson, 2004 p.40). Spreadsheet models can be used to forecast short term staff needs, such as how many people are needed at different parts of the day or week, and also to suggest how many people will be needed at different parts of the year.



Both gratis and fee-based commercial software packages are available. The example from the screenshot below can be found at http://www.erlang.com/calculator/call/ and is a simple example of the types of inputs and outputs used.

The calculator uses the Erlang-C traffic model, established by A.K. Erlang, and used in telephone traffic theory and generally in queuing theory. This model makes the following assumptions (Westbay Engineers) (Weisstein, 2014):

- calls are presented randomly to the advisors with a Poisson distribution
- when no advisor is available, the caller will wait in a queue which has unlimited capacity
- length of calls are exponentially distributed

- calls are answered in the order they arrive
- the call is directed at the first available advisor

The makers of the calculator acknowledge that queues and customer patience is not infinite in real life but argue that if the call centre has reasonable targets, i.e. most of the calls are answered in 20 seconds, then this lack of reality is not a problem (Westbay Engineers). The company then offers a paid-for local version that provides further statistics, such as advisor utilisation[1].

Academics in the area of queuing theory argue that there are many limitations of this method. These include the assumptions of which distributions are used, and that all customers and advisors are "statistically identical"; the method doesn't take into account customer impatience or the advisor's level of skill (Brown et al., 2005 p36-37).

Therefore, the benefit and limitation of the Erlang formula is its simplicity (Harris, 2014, p.6), and for day to day use it is a useful tool for determining what the staffing levels should be. As seen in the screenshot, the manager just needs a small amount of historic data and is presented with recommended staffing levels in a quick and convenient way but something more reflective of reality would be an improvement.

Some companies have adapted products that better reflect the complexity of the situation, For example, Portage Communications offer a piece of software similar to the Westbay Engineer version, but which also takes into variables such as abandonment (PortageCommunications, 2014). However, even these methods are very static in terms of timescales. A week of planning can be achieved but long term effects on calls are more difficult to plan for.

More complex mathematical approaches have been taken to solving contact centre staffing problems. The paper *Staff scheduling and rostering: A review of applications, methods and models* outlines how constraint programming has been shown to be useful when the workspace is rigid but that it can be slow and uncompromising (A. T Ernst et al., 2004 p.17), that set-covering approaches can be too heavily simplified to be effective (A. T Ernst et al., 2004 p.19), and that a multi-method approach, such as constraint processing combined with linear programming's branch and bound has been shown to get around some of these issues (A. T Ernst et al., 2004 p.17). In effect operational research mathematical approaches are effective for call centre staffing but they unsuitable for use by call centre managers, who are not suitably trained.

Furthermore, one shared limitation of all of these methods is the reliance on historical data; as will be shown, this is very risky.

2.3 COMPANIES AFFECTED BY THE WEATHER

Almost every home in the UK has a central heating system that is powered by either gas or electricity, and all systems need maintenance. For example, an article by David Myron (Myron, 2002) showed an increase from 3,000 to 8,000 calls per day to the New York City Department of Housing Preservation during the colder months. The colder weather forced customers to contact the company taking care of their property when something went wrong with their heating systems. Most customers with boiler cover will also need to book a service each year. Therefore, historic call data is affected by a mix of service due times and the severity and timing of the winter months. In these scenarios the unknowable impact of the severity of the weather can make planning the staff resources required very difficult, especially if the company does not have years of historical data to use as a baseline from the different types of winter.

[1] utilisation is a measure of productivity calculated by the amount of time spent productively over the amount of time the advisor is available. For example, if a call centre agent works from 9am till 5pm with 50 minutes of breaks and is on the phones for a total of 5 hours, utilisation is 5h/(7h10m) = 70%

Degree-days are a standard unit for use measuring energy consumption, and can be used as an alternative to mean temperature when measuring weather patterns. They have been shown to have a correlation with heating costs to a 99% confidence level (Meiera and Rehdanzb, 2010, p.954). To calculate them, take a relevant temperature for the scenario, e.g. 15°C, and work out how long and how many degrees the temperature is below that amount. For example:

Example	Degrees below 15°C	Time below 15°C (days)	Total degree-days
All day at 14°C	1	1	1
All day at 13°C	2	1	2
Half a day at 16°C, half a day at 14°C	1	0.5	0.5

One benefit of degree-days is that the comparator number can be changed as needed; for example, it may be better to compare it to 17°C depending on what the measurement is used for. It also increases with the cold, which is easier to follow when comparing it to incoming call rates and finally, as will be shown later, it has a stronger correlation with incoming calls than mean temperature.

Figure 2 shows the number of degree-days by month over a ten year period[2]. The blue line shows the mean for the season and it is clear that no two years are the same. Figures 3 and 4 emphasise the variability of the weather temperature, as the peak month, height of the peak, width of the peak and the month at which the temperatures drop are all variable. A contact centre using historic data when determining staffing levels for the winter of 2011 would end up drastically overstaffed with significant cost implications.

This section has been emphasised to show the risk of relying on historical data during call forecasting without using the context of the weather when incoming call rates are affected by the weather. By providing an example from a company that manages home maintenance and a study that shows correlation between degree-days and heating use, the aim is to show that, although this topic is not well-covered by the academic literature, there is a strong reason to consider weather when determining staffing levels.

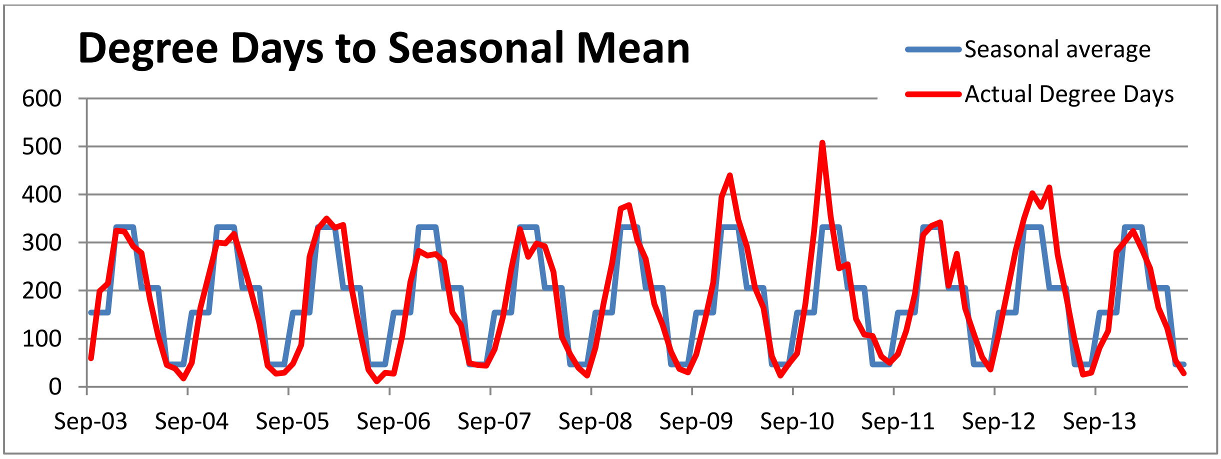


Figure 2 Seasonal Degree Day Data

[2] Data from www.vesma.co.uk (VESMA)

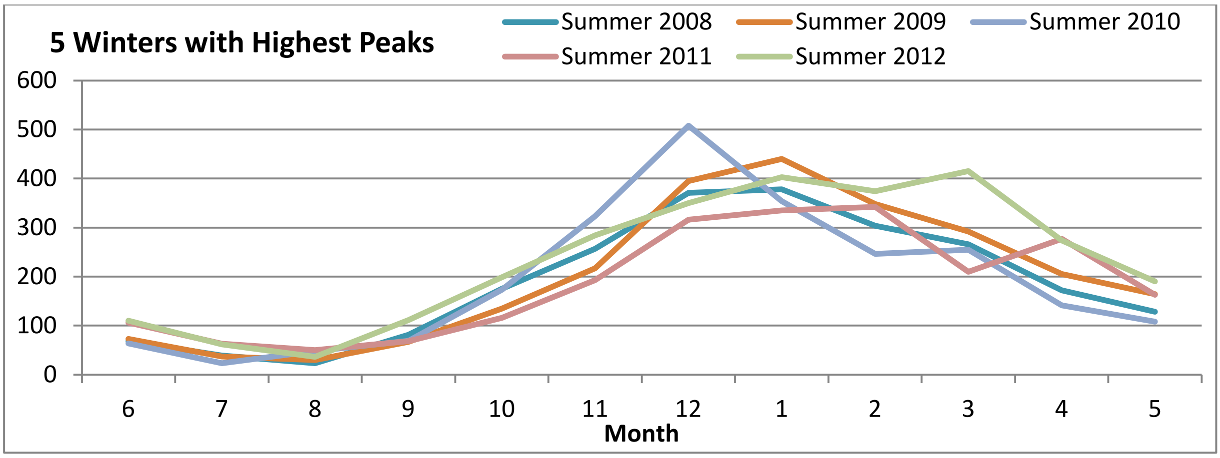


Figure 3 5 Coldest Winters

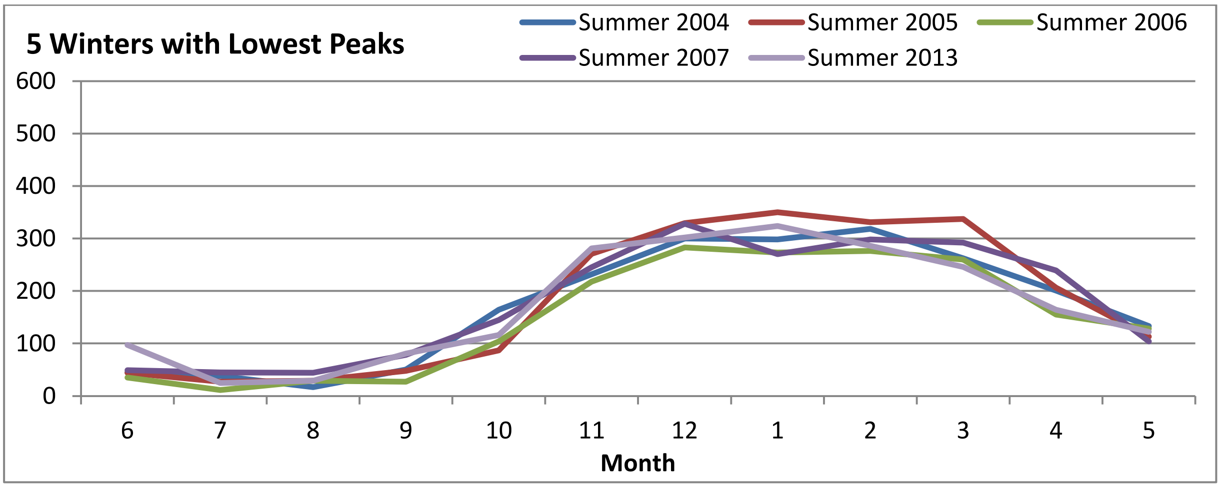


Figure 4 5 Warmest Winters

2.4 HUMAN RESOURCE PROCESSES

Recruitment can be carried out by a Human Resources (HR) department, individual line managers or, most commonly in contact centre businesses, a mix of the two. The time taken for recruitment is the sum of several different activities: placement of an advert, assessment of applications, accepting candidates, and candidate training. The overall time will be a minimum of three weeks (Appendix 1, HR Questionnaire). At any point a preferred applicant can drop out of the process, so wasting a lot of time and money. The risk of a potential staff member leaving is worse for contact centres than other industries, with a 25% rate of average turnover verses 18-20% for the rest of the economy (Research, 2004, p.7). As a result, predicting the number of contact centres which will be required is a difficult task even if the number and timing of vacancies is known well in advance.

Although there is software for managing individual applications and personnel files, there does not seem to be publically available software aimed at optimising recruitment processes.

2.5 SIMULATION

Given the situation has two stochastic variables, weather and recruitment success, it was determined that simulation would be the best approach.

Robert E. Shannon concisely summarises the benefits of simulation:

> *"Simulation is one of the most powerful tools available to decision-makers responsible for the design and operation of complex processes and systems. It makes possible the study, analysis and evaluation of situations that would not be otherwise possible. In an increasingly competitive world, simulation has become an indispensable problem solving methodology for engineers, designers and managers."(Shannon, 1998, p.7)*

This project takes the definition of simulation after Stewart Robinson:

> *"Experimentation with a simplified imitation (on a computer) of an operating system as it progresses through time, for the purpose of better understanding and/or improving that system" (Robinson, 2004, p.4)*

Simulation has three key advantages as a modelling method:

1. The variability inherent in the real world system can be modelled. This is particularly necessary for this scenario, due to the large number of uncontrollable variables.
2. The number of assumptions required is dependent on the time and data available, not the methodology, so final model can be as complicated as needed from the scenario
3. The results are transparent to the end user, so it is much easier for call centre managers to understand and trust the model (Robinson, 2004,p. 7).

Simulation is, however not without its disadvantages. Notably, it is an expensive option for a company to purchase software for a one-off project, and to hire the non-trivial expertise needed to create a high-quality simulation; it is also very time-consuming and can require a lot of data, as a lack of constraints can lead to a solution which is too detailed. It can also lead to overconfidence due to its apparent transparency; it is always important to consider the results of a simulation as part of a wider cost-benefit analysis (Robinson, 2004, p.10).

There are three core types of simulation available, and two are particularly applicable to this scenario.

SYSTEM DYNAMICS

System dynamics was first developed to investigate economic and social systems like the adoption of a new product. The technique considers the physical system as a network of feedback loops with a high degree of mutual interconnection, making it useful for high level process modelling(Borschev, 2013, p.38). If this was a model of a larger system, such as how departments affected each other, than a system dynamic simulation would be more applicable. For example, increased sales would have a positive feedback loop to increased calls; this would show that the number of call centre advisors would need to increase to manage these calls or it could cause the number of customers to decrease.

DISCRETE EVENT MODELLING

Discrete event modelling is where the system is operated as separate sequences of events in time. The technique saves on computational requirements because the only points in time marked are those when the system experiences a change in state (Robinson, 2004, p.15).

Each process begins with a source, which generates entities; they will then pass through queues and delays that represent the system(Borschev, 2013, p.45). Typically the technique has been used to model manufacturing processes that lend themselves well to queues, such as conveyer belts and service blocks where goods and workers interact. This emphasis on queuing also lends itself to call centres, as different types of call can be created and split according to type, then entering a queue until an advisor is available to deal with the call.

AGENT-BASED MODELLING

Agent-based modelling was developed as a result of improvements to computer modelling techniques from computer science and improvements to computing hardware. Instead of modelling the process, the objects are modelled individually so that individual entities are generated that have their own unique behaviours. The agent is made up of a set of states that it transitions through in response to external or internal triggers. Agents can then interact with an environment or other agents according to their current state (Borschev, 2013, p.50). It has been shown that agent-based modelling can effectively show proactive behaviour and that this improves simulation accuracy (Peer-Olaf Siebers, 2011 p.22). To be used in a call centre scenario, we may consider customers to be modelled as agents with different needs. For example, the customer may be impatient but with a minor complaint that will call in multiple times and abandon if the call is not answered very quickly or be a type that will wait for 30 minutes to speak to an agent.

SIMULATION AND CALL CENTRE STAFFING

Some call centre software includes simulation. As summarised by Mehrota (2003, p.136) there are three key ways simulation is currently being used in the contact centre industry:

1) The creation of specific models to analyse a particular area
2) An application provided with the contact centre routing software that allows routing engineers to see the impact of potential decisions
3) An application provided with the contact centre scheduling software that uses simulations powerful use of stochastic variables and process interaction as part of a mathematically based optimisation solution

Figure 5 is an image from the conference paper that shows the key variables to be considered when creating the engine.

In line with Erlang-C calculators, the key factors are the queue, and volume; however, these are also able to consider abandonment, advisor skill levels and schedules (Mehrota and Fama, 2003, p.137) to provide much more realistic outcomes.

The paper is concerned by the distributions used for service time and customer patience due to the lack of real data. As a result, these systems rely heavily on averages and so may not be providing a true reflection of the

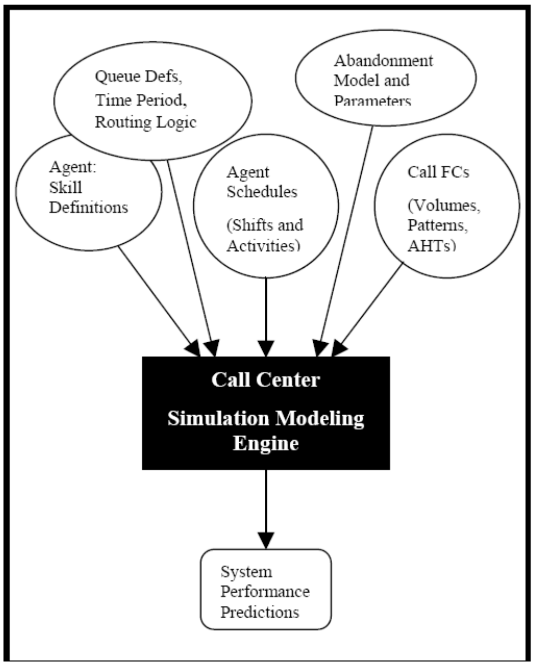


Figure 5 Call Centre Modelling Diagram (Mehrota and Fama, 2003 p.137)

situation (Mehrota and Fama, 2003 p.138). This concern is supported elsewhere as one of the few case studies of real data available publically found that service times and time to abandon did not match the traditionally used exponential and Erlang-C distributions (Brown et al., 2005, p.49)

This emphasises the importance of having access to reliable call data to confirm the assumptions of the simulation are close enough to reality before performing experiments.

2.6 PUBLICALLY AVAILABLE SIMULATIONS

Any Logic is a programme that can be used create all of the types of simulation discussed, including systems which are made up of hybrids of two of more of the methods. It is built in Java, allowing objects to be customised, or new objects created, by using the Java programming language.

There are two publically available Contact Centre Models available for AnyLogic[3], shown in screenshots on the following page.

Call Center is a default simulation available from the AnyLogic help files. It is a discrete event simulation that allows the user to manually change call arrival rates and call service times. The main use of this model is that it allows the user to see the effect of practising cross skilling, in which advisors are trained to handle different types of calls so that they can answer calls from other departments when queues get too long. For example, we can imagine that sales advisors usually handle sales calls but that during peak periods during the day they could be trained to take the overflow calls from general customer enquiries. As the advisors do not usually take the calls, they will take longer to answer them. This simulation would be useful to someone who wanted to determine the benefits of cross training and how well trained advisors would need to be in order to be effective.

The simulation is easy to understand, with good use of graphical representations such as histograms and pie charts of the model processes to give instant feedback from the inputs chosen. A particularly nice touch is the bar which fills from blue to red as the queue gets longer, providing instant visual feedback. One problem is that it is difficult to set precise numbers using the sliders; it is also not clear whether the unit is based on a time unit of minutes or seconds.

The model uses the inputs and outputs discussed earlier:

- inputs: arrival rate, service time, number of advisors
- outputs: calls answered in 20 seconds, number abandoned, average waiting time

Customer Support Centre is a much more complex simulation. It is a hybrid discrete event & agent-based model, where requests of different types and complexities are generated through a more realistic method that increases during peak periods of the day, with user-configurable peak levels. The staff who deal with the requests are modelled as agents with work and idle states that they change between based on whether they are interacting with a request. New agents are generated each week as set by the user; these agents are removed over time from the available resource so that they can take part in training and improve in skill set and expertise. This model can be used to determine the effects of separating different types of tasks such as answering emails or calls between different types of skilled advisors, and includes the effect of advisors leaving the system. Discrete event simulation objects are then used to allow the workers to interact with the tasks that reflect the stages of the task handling process.

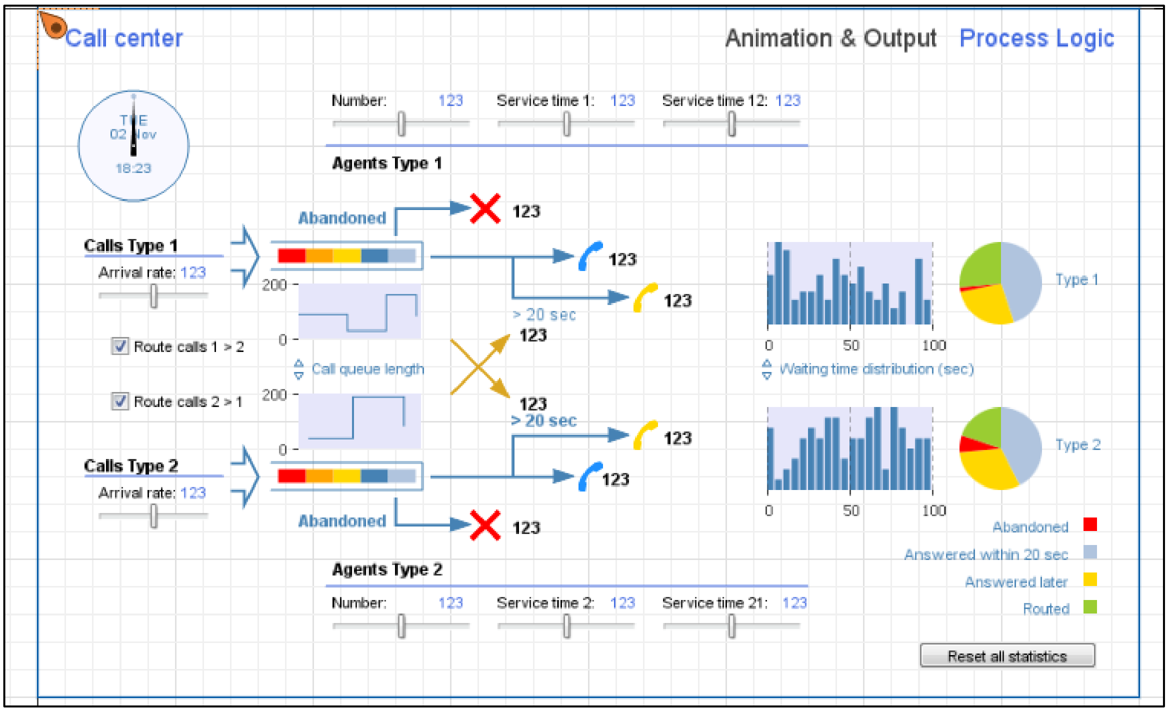
This model is designed for an experienced user. There are no explanations on the starting page but the user is presented with 27 different inputs that can be updated. These include the ability to set the starting state of the

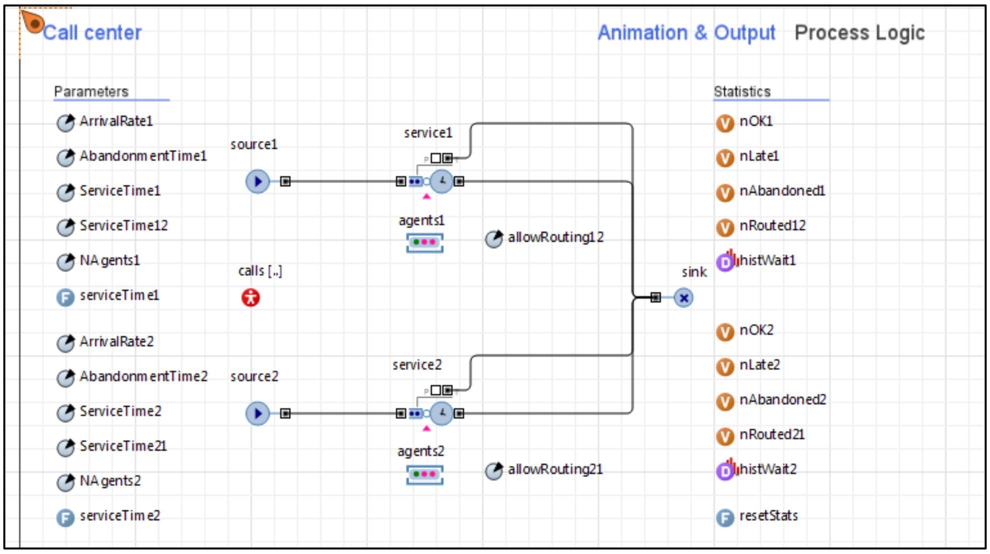
[3] To try these models go to Appendix 3: How To

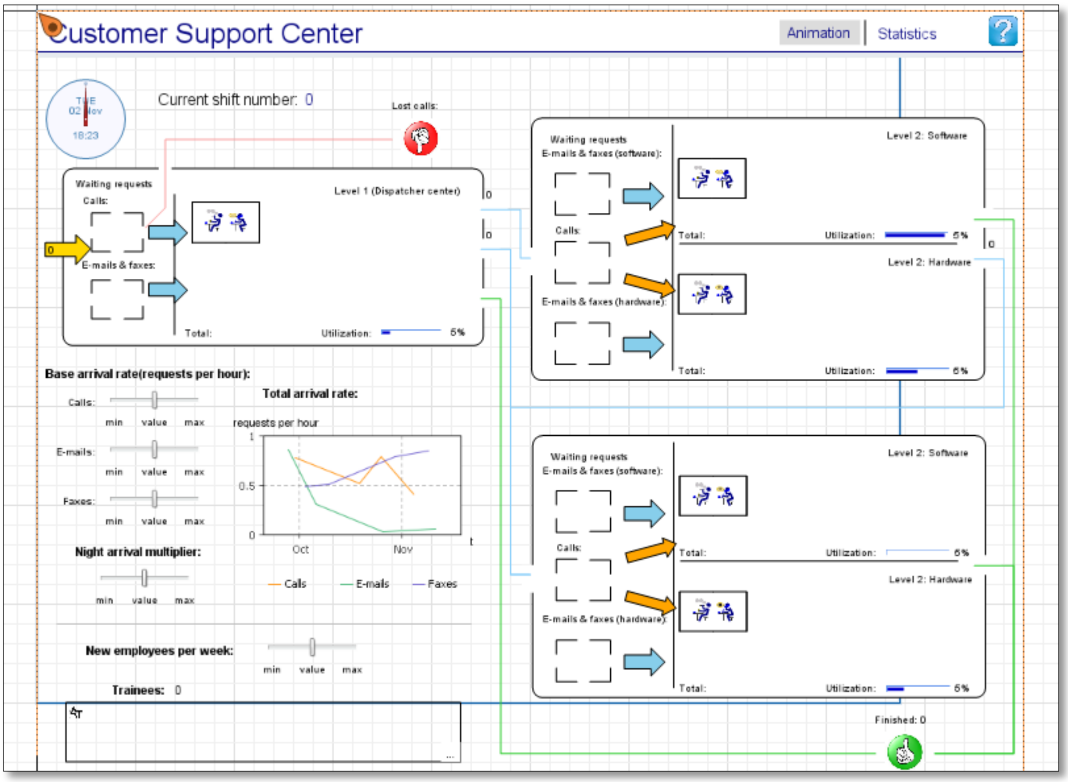
number of workers trained to each level, the complexity level of different types of tasks, service times and base arrival rates. Although default values are available, this level of detail can be overwhelming and makes it difficult to know what effect changing the inputs has.

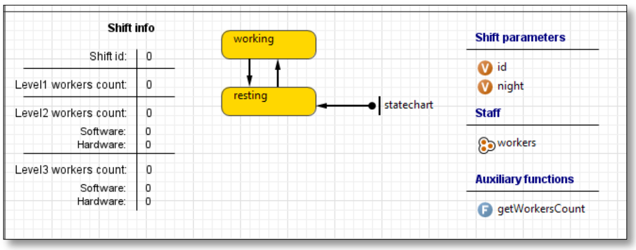
This model is unique in the use of tasks with varying complexity and workers with different skill levels.

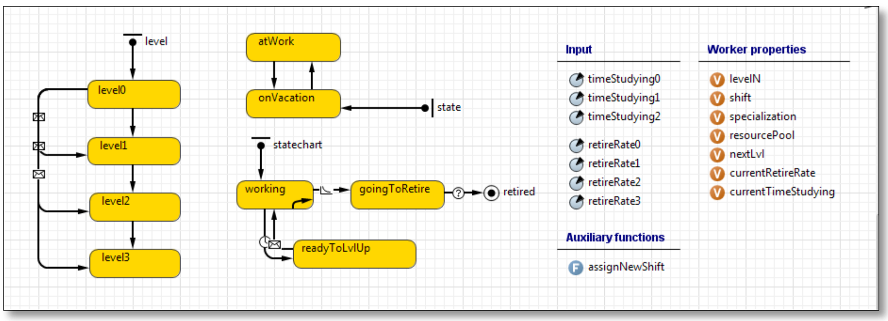
CALL CENTER






Customer Support Center
Animation
Statistics
Current shift number: 0
Lost calls:
Waiting requests
Calls:
E-mails & faxes:
Level 1 (Dispatcher center)
Total:
Utilization:
Level 2: Software
Level 2: Hardware
E-mails & faxes (software):
E-mails & faxes (hardware):
Base arrival rate(requests per hour):
Total arrival rate:
requests per hour
Night arrival multiplier:
New employees per week:
Trainees: 0
Finished: 0
Calls
E-mails
Faxes


Shift info
Shift id:
Level1 workers count:
Level2 workers count:
Software:
Hardware:
Level3 workers count:
working
resting
statechart
Shift parameters
id
night
Staff
workers
Auxiliary functions
getWorkersCount


level
level0
level1
level2
level3
atWork
onVacation
state
statechart
working
goingToRetire
retired
readyToLvlUp
Input
timeStudying0
timeStudying1
timeStudying2
retireRate0
retireRate1
retireRate2
retireRate3
Auxiliary functions
assignNewShift
Worker properties
levelN
shift
specialization
resourcePool
nextLvl
currentRetireRate
currentTimeStudying

2.7 CONCLUSION

In conclusion, there are currently a number of software solutions for day to day optimisation of call centre staffing levels. However, they are best used when the system is reasonably static with predictable peaks and troughs. Call centres which are affected by the weather are complicated by the interaction of two extremely variable stochastic inputs: call levels over the winter period and recruitment success.

Simulation offers an answer to these difficulties. Previous work has already its suitability for short term analysis, and it may be the best approach for long-term what-if analysis.

The literature recommends that the key inputs should be service time, customer patience and incoming call rate. The key outputs should be a measure of utilisation, calls answered in a specific timeframe and handling time. It has also been shown that special attention should be taken when deciding on distributions used when creating the simulation.

3 METHODOLOGY

The aim of this project is to create a tool to support contact centre decision making for staffing levels in response to winter pressures. This will be met through simulation.

3.1 SIMULATION LIFE CYCLE

Simulation is more than the final model; it is the process at which that model is arrived. Main stages of a simulation life cycle, as outlined by Robinson (Robinson, 2004, p.52) are:

1) Have a real world problem: there is no limit to what can be modelled, only a restriction on time and resources. Therefore a clear understanding of the problem is needed to create an effective simulation.
2) Conceptual model: a paper based model outline that can be kept separate from the computerised model. It is the process by which the problem is understood and realised. Creating the conceptual model involves communication with stakeholders in order to ensure that requirements are met and a good model will clearly outline the outline the data requirements and limitations needed for the computational model. It may be that the conceptual model is all that is required to offer a solution to the problem as the questions the modeller needs to ask can be enough to highlight paths not previously considered.
3) Computer model: this is where the conceptual model is coded into a computer model; this can be through the creation of a new programme or by using simulation software.
4) Solutions/understanding: experiments are performed using the simulation in order to find a solution to the real world problem.
5) Implementation: in a perfect world the proposed solutions are implemented.

As shown in Figure 6, these steps are not a linear progression. A conceptual model may be signed off by a stakeholder but the chosen method of implementation may add an unknown constraint that means the conceptual model needs to be updated. The real world situation may change during the creation of the simulation leading to assumptions being incorrect. Furthermore, continuous validation throughout the modelling process can mean that the implementation significantly different from the original simulation imagined.

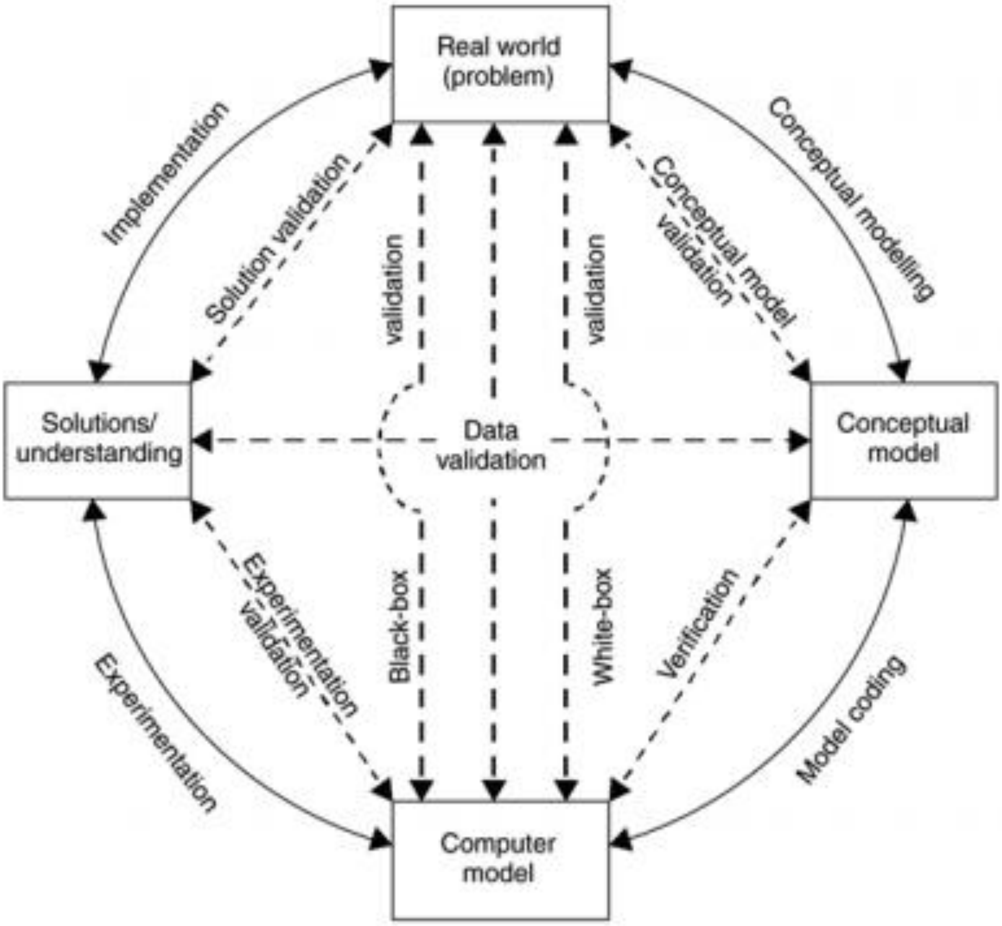


Figure 6 Simulation Life Cycle (Robinson, 2004, p.211)

Using this outline, the following methodology was used:

1) The real world problem is the difficulty of preparing for staffing levels in a call centre which handles boiler maintenance and repair, because the number of calls is affected by unpredictable weather. This is compounded by the lack of foreknowledge as to how successful the recruitment process will be.
2) A conceptual model was generated from the findings from the literature review. This produced a framework that was used in informal interviews with call centre managers where assumptions and requirements were validated.
3) The simulation is a discrete event simulation as the problem is predominately a supply problem with a high impact random event. The original aim was to improve this with an agent-based hybrid, but owing to time-constraints this simulation is predominantly a discrete event simulation with passive agents that have individual characteristics. It was created in AnyLogic due to familiarity with the software and the ability to create hybrid systems quickly and effectively. It was also designed with a user friendly front end designed around Nielson's usability heuristics; this is explained fully in the model design section.
4) The simulation was created in two steps. First, as a static model with no changes in call rates or number of advisors; this was validated and verified, and the warm-up period and number of replications determined. An experiment was also conducted at this stage to compare distribution types of abandonment as this issue had been highlighted in the literature review section. Secondly,

experiments were performed using a seasonal version of the model where different versions of advisor contract lengths and weather severity were compared and contrasted. The user friendly version of the simulation was shown to stakeholders and feedback received.

5) Implementation: the simulation did not reach the stage of implementation, however key findings from the data collection stage resulted in a change of processes once the effect was shown with the simulation and validated against real-world changes.

Data used in this simulation was taken from publically available sources when possible. This was then refined with real-world data that has been slightly altered and redacted in places in order to protect potentially sensitive information related to the company.

3.2 CONCEPTUALISATION

PROBLEM

A boiler maintenance and repair company receive calls from customers throughout the year to book appointments. Service calls stay reasonably constant throughout the year, however breakdown calls increase during colder months. Staffing levels need to be increased in order to meet service level targets, however level is determined by financial constraints. The company needs a simulation to assist managing the risk of hiring levels decided.

OBJECTIVE

Consistently meet the following KPIs from June to May:

- Average time in system < 6minutes
- Abandonment rate < 5%
- Calls answered within 20 seconds >90%
- Calls per advisor per hour between 6 to 8

MODEL JOURNEY: OBJECTIVES

As shown in the literature review, there are many potential targets for a call centre. It was decided to choose one indicator of utilisation, one indicator for customer satisfaction and one indicator of queue bunching and advisor experience.

Both abandonment rate and calls answered within 20 seconds have been used as indicators of customer satisfaction, this is because during the requirements gathering stage it was determined that only one of the indicators was used whereas in all literate of best practise, both could be found. Due to the difficulties around the correct distribution to use for abandonment it was determined to use both indicators and see the interaction between the two.

Calls per advisor per hour were used instead of utilisation due to a limitation with the AnyLogic software. The simulation was constructed in AnyLogic v6 and later ported to AnyLogic v7. The resource pool object and how utilisation is calculated is not well documented, and it was determined that it would be better to choose an output that could be calculated independently of the software. This issue was fixed in v7 of AnyLogic, unfortunately too late for the model to switch over.

Average time in system became a measure of advisor experience, as once criteria are set to minimise waiting times, the biggest impact on the time in the system was found to be how long it took the advisor to handle the customer's query.

The targets set are in line with best practice and call centre benchmarking (Government, 2003).

CONSTRAINTS

- Must be at least 3 member of call centre staff on shift
- Contact Centre receives calls 24/7

MODEL JOURNEY: CONSTRAINTS

This section underwent the least amount of changes. Rostering.co.uk was used for daily call data and staff distribution; the primary difference between this data and the problem scenario was that the real-life call centre has 24 hour working. This needed to be included in the model so that the call centre never completely emptied from one day to the next, but the night time data should not be included in the data collection phase. This is because the number of advisors for night time working is determined by labour laws ensuring that sufficient staff members are available to cover for breaks during the shift. The lack of incoming calls is made up for by the advisors conducting other forms of work that are not reflected in the model.

INPUTS

- FTE available at model initialisation
- Target FTE
- Winter severity level
- Customer base number
- Experimental only:
    - Abandonment rate distribution
    - contract length of new staff

MODEL JOURNEY: INPUTS

The model inputs went through many stages. At first, all objects affected by a distribution were counted as inputs, for example service time, time till customer abandons the call, and time till advisor leaves the company. Once these distributions were determined through statistical analysis of real world data, it was judged to be overcomplicating to model them as inputs rather than levels of detail associated with the object. Further work could be carried out to add these as user accessible inputs, similar to the Customer Support Centre simulation, as a separate and more detailed screen for experienced users. This was beyond the scope of this project, and so the inputs for the non-technical user version are limited to the number of staff and level of winter severity.

The customer base is used instead of number of calls per day. This was for two reasons:

- Confidentiality – the company is able to use a number similar to actual customer base and get a reasonably realistic version of results, and can set this number easily. This is not the default figure as shown in the simulation.
- Breakdown and service calls can come from one figure. Instead of setting the number of calls on service lines and breakdowns from two separate inputs, these figures can be calculated as a

proportion of customer base, so providing a simpler method of setting incoming call rates for non-technical users.

OUTPUTS

- Calls answered:
    - number of calls that reach the sink when the call was not abandoned
- Calls abandoned:
    - number of calls that reach the sink when the call was abandoned
- Answered less than 20 seconds:
    - number of calls that where answered after queuing for less than 20 seconds
- Number of advisors available:
    - total number of advisors available to work that day
- Time in system:
    - length of time the call is in the call process system

This data was collected from calls that took place between 8am and 8pm. This is the standard daytime operating hours of companies that offer boiler repair and servicing (British Gas, 2014, nPower, 2014).

Once the objectives were defined the outputs fell into place. These outputs offered the raw data needed to perform the required calculations for the graphs that with provide information to users when running the simulation and as exported to excel data files for analysis when conducting experiments.

REQUIRED DATA

Robinson outlines three types of data(Robinson, 2004, p.97):

- Category A are already available because they are clearly visible or have been previously collected
- Category B are data that needs to be collected but is currently not available.
- Category C are data that is unavailable and will always be unavailable.

In the case of this project, category A data are available for most requirements.

Data Requirements	Data Type	Data Source	Other comments
Distributions of calls and staff throughout the day	Type A	Public Source	Publically available data about typical peak times was used
Call data	Type A	Company data	Included: number of calls by queue, time waited in queue and outcome (abandoned or answered)
Fixed Term Contract Lengths	Type B	Questionnaire	Available from Appendix 1 HR questionnaire. Some information has been redacted.
Average time till advisor leaves	Type B	Questionnaire	

Although real-life call data was available, distributions for day-to-day modelling were taken from a public source so that model validation could be shown in this project without redacting information. Testing with call centre managers confirmed that these distributions were similar to that experienced by the business, although this did cause issues with trusting the model during the user testing phase.

Although actual data could have been gained for advisor length of service, the lack of public data to hide this information and the time required for the company to provide the information proved to be too great a disadvantage and so basic questionnaires were used to get high level information. Where this matches benchmarking studies, the information has been left in the project and sensitive information has been hidden. For example, three lengths of contract were used for experimentation but it will not be stated which is used by the company.

This limits the validity of the model but should still enable the resulting simulation to be useful.

ASSUMPTIONS

- Weather and recruitment are not affected by location in the UK
- One member of staff interacts with one customer at a time
- Customers who have abandoned calls do not call again

As the data had to be protected it was better to take overall amount of calls rather than calls per individual customer, which would have allowed for a more accurate amount of incoming calls. It was determined that this approach would suffice for a model of this detail; however, this is a prime area for further study.

SIMPLIFICATIONS

- Customers calling in multiple times is included as part of arrival rate
- Two types of calls i.e. services and breakdowns included, but not sales calls
- Staff members leave according to distribution, there is no difference between staff members who resign or are dismissed
- Customer base is static
- Recruitment process focus on training time
- Unlimited queues – no dropped calls

To keep the model to a reasonable size, the types of calls and staff were kept simple. It was hoped to include agency staff and an advisor type but this proved to be unachievable in the required time frame. However, some of the simplifications have no effect; for example, it does not matter in a discrete event simulation why an advisor leaves, only that they do. This would be more important for an agent-based model as other variables could affect the advisor’s chance of leaving.

The restriction on customer base means that the simulation should only be considered for one winter at a time.

The time taken to recruit was an unnecessary variable as it could be reflected with an input of new starters each week for a period of time. It was more productive to focus on the time taken to train the advisors as this had an impact on the number of advisors available and the experience level of the call centre staff.

CONTENT

SCOPE

Component	Include/Exclude	Justification
Calls	Include	Key input
Advisors	Include	Key input
Managers	Exclude	Unnecessary level of complexity
Queues and service points	Include	Key process

Location	Exclude	Unnecessary level of complexity
Weather	Include	Key Input
Time	Include	Key variable

LEVEL OF DETAIL

Component	**Detail**	**Include/Exclude**	**Justification**
Calls	Type (Service/Breakdown)	Include	Necessary level of complexity
	Complexity	Exclude	Difficulty of call inferred in separation of call type
	Outcome	Include	Whether call answered or abandoned are key outputs
	Time in queue	Include	Needed for output
	Time in system	Include	Needed for output
Advisors	Type (Permanent/Fixed Term)	Include	Effects setting for time till leave
	Time till leave	Include	How long the advisor will stay with the company, needed to maintain realistic staffing levels
	Manager		
	Experience Level	Include	Necessary as effects service time
Queues and service points	Allow time out	Include	Allows calls to abandon
	Service time	Include	Set here rather than with the customer so that can be more easily affected by advisor experience
Weather	Degree-days	Include	Experimental factor
	Precipitation	Exclude	Unknown effect
Recruitment	Time to place advert	Exclude	Unnecessary level of complexity
	Trainer time		
	Time to conduct interviews	Exclude	Unnecessary level of complexity
	Time for training	Include	Impacts number of available advisors
	Time till experienced	Include	Impacts effectiveness of advisors
Time	Day	Include	Use for daily distributions
	Week	Exclude	Level of detail between weekday and weekend unnecessary level of complexity
	Month	Mixed	Changes in monthly weather included but any variation within month

DIAGRAMS

UML and process flow diagrams have been used in keeping with techniques seen during the literature review.

AGENT CLASSES

These diagrams are the classes used for the passive customised agent objects

Call Centre Advisor Class

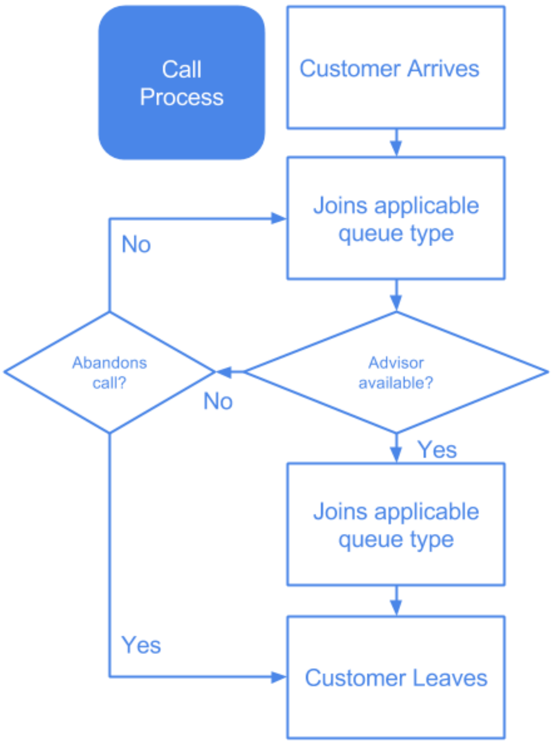
- Type (permanant or fixed term)
- Level of training
- Time till leave

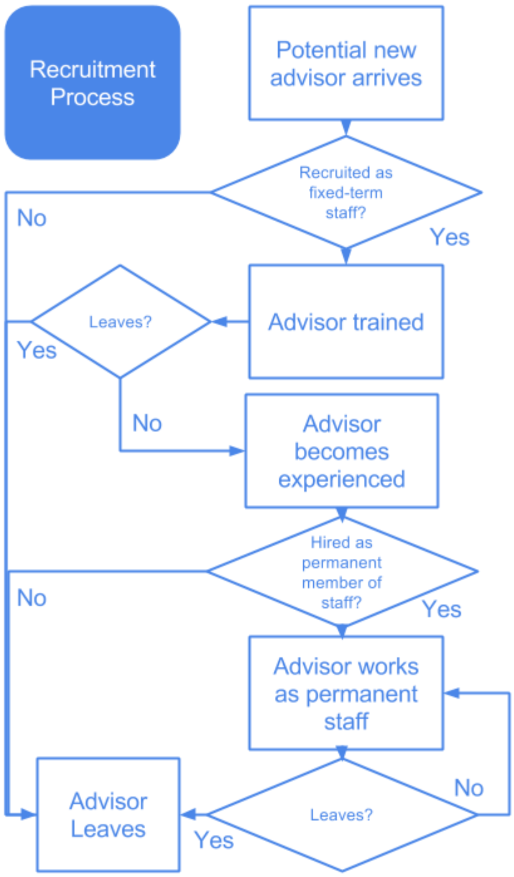
Call Class

- Type (Service or breakdown)
- Time till abandon
- Time enter queue
- Time leave queue
- Time enter system
- Time leave system

PROCESSES

The processes are modelled with logic flow diagrams typical for discrete event simulation (Robinson, 2004, p.72)





4 MODEL

4.1 IMPLEMENTATION

INTRODUCTION

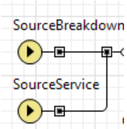
The simulation was built using the software *Anylogic: V6 University Edition* and then refined in *Anylogic: V7 University Edition Trial* and can be viewed using the How To in Appendix 3.

The presentation facilities and ability to run the model in a Java-enabled internet browser mean that the simulation produced can be shown to stakeholders that do not have access to the software.

The first step was to create a limited simulation that could take inputs of data from industry and provide realistic outputs; this is referred to as the static simulation. This simulation was then tested for validity and adapted to accept changing variables required for experimentation, referred to as the seasonal simulation. The Java applet available from the seasonal model designed around non-technical uses is referred to as the recruitment tool.

4.1.1 KEY MODEL ELEMENTS

SOURCE OBJECT



Two source objects were used per process to generate the different types of calls and advisors. The object has a rate of arrival and number of entities per arrival and assigns unique values to the newly created entities. A full breakdown of the code used in the source can be found in Appendix 3

The distribution of staff and calls throughout the day was determined by using a company that offered roistering support (Resource Analysts, 2014). Their website shows a traditional day in a contact centre with a number of calls and the staff required to handle them. These graphs are shown in Figure 7. The black line on the top left graph show the number of staff required to meet the number of calls presented (top right graph) with 90% of calls answered within 20 seconds. This provides us with the total number of staff – 18 for one day, and the total number of calls, 726 for one day and how they are distributed by half hour.

The simulation calculation for the number of calls per hour can therefore be calculated as:

```
(callsPerDay*rateCallSchedule.getValue())/1*minute()
```

where the `rateCallSchedule` is the proportion of calls expected that hour.

The number of staff available was set by a function that was triggered at the beginning of the simulation and every following hour. The code for the number of advisors[4] per hour is:

```
functionNoOfStaffInPool()
```

[4] In the code agent and advisor are interchangeable. This is because the model was designed in the language of contact centres that have 'agents' but when preparing the project it was important to differentiate the call centre agent from agent-based modelling.

```
/*get the total number of agents available to work that day and
multiply it by 0.71 to reflect 5 day a week working */
pNoCCAgentsInPool = (int)(pNoCCAgentsAvailable * 0.71);

/*find out the proportion of agents that should be working at this
time of the day*/
double a = agentsAtWorkSchedule.getValue();

/*set the resource pool to the number of staff that should be
working, unless the schedule is set to 0 (to reflect nightime
working), in which case, set the number of staff to 3. The number of
staff must be an integer*/
if (a == 0) {
        pNoCCAgentsInPool=3;
        } else {
                a = a * pNoCCAgentsInPool;
                if (a<=3){
                a = 3;}
        pNoCCAgentsInPool = (int)a;}

CCAgentPool.set_capacity(pNoCCAgentsInPool);
```

The changes made from the website data is the extension of the hours from daytime working to 24 hour and a move from half hour to hourly intervals to simplify the scenario.

The bottom graphs show data from the simulation, showing that the distributions have been realistically reflected. This shows that white box validation has been met for this object[5].

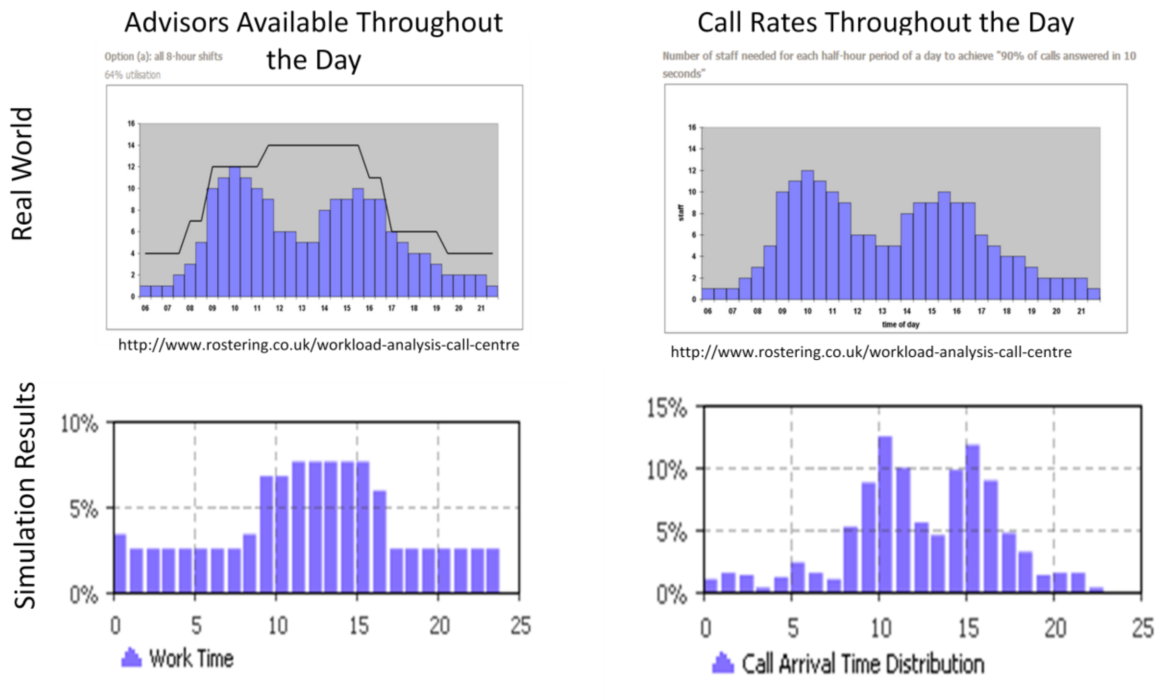
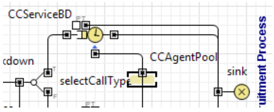


Figure 7 Comparison of Daily Simulation Output to Real world Distributions

[5] The different types of validation will be discussed in the validation chapter but it made more narrative sense to show key stages of validation with the associated object.

SERVICE AND RESOURCE POOL



A Service object is made up of a queue and a delay. For this simulation the service objects fulfilled the following roles:

1) Limited the number of calls answered per hour by linking the call to worker, this is done through the resource pool whose capacity was discussed in the previous section
2) allowed for the logging of queue times and advisors that waited less than 20 seconds

```
if (time left queue – time entered queue <=20*seconds())  then
answeredIn20Sec++;
```

3) Allowed calls to time out and leave the queue if their Time To Abandon was reached before interacting with an advisor
4) Set the service time

```
(lognormal(5.79643, 0.614067, 0)/60) * vServiceTimeModifier
```

It was found from the real world data that service times for different types of queues were very different.

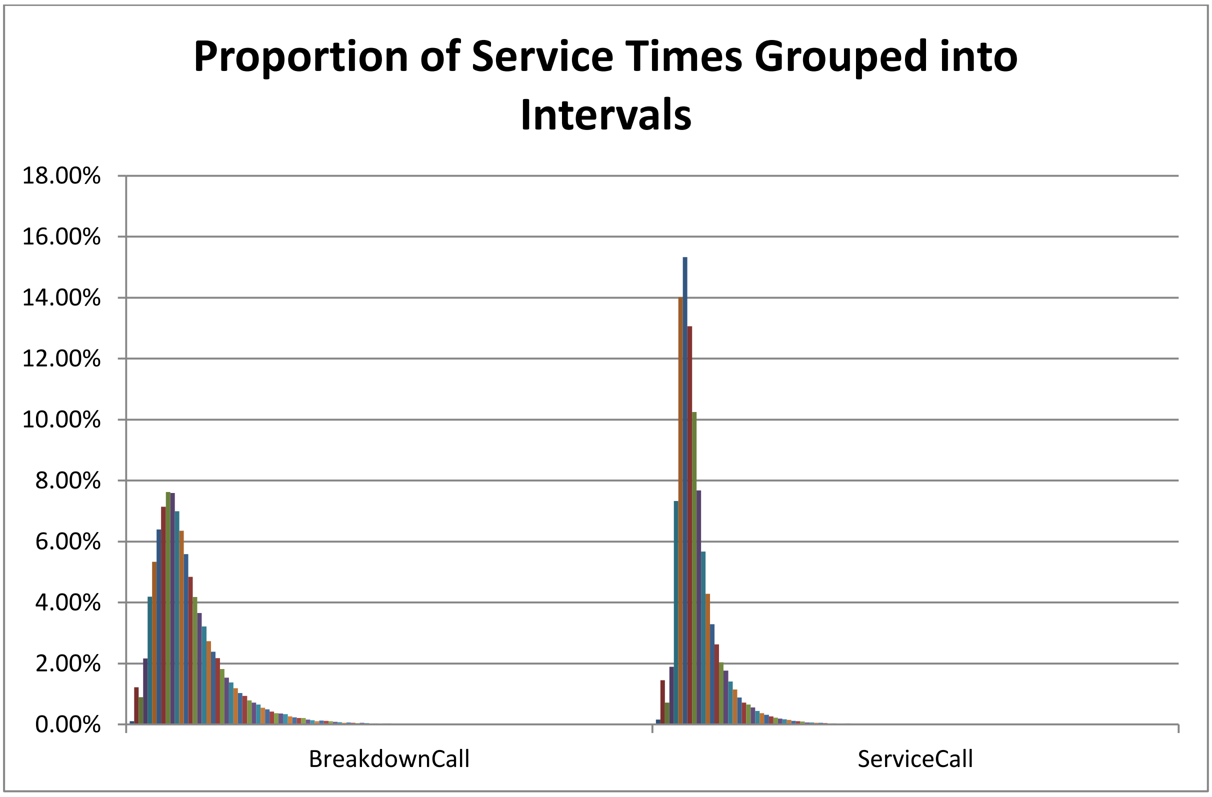


Figure 8 Real World Service Times

At this point MATLAB distribution fitting was used to calculate the distributions with best fit, as shown in Figures 8 and 9. It was shown that this data matched findings from the Brown study that lognormal rather than exponential or gamma were the best fit (Brown et al., 2005, p.39-40). Therefore the lognormal distribution was used for service times.

The base service time was supplemented by the experience level of the current batch of staff. As stated before, it was not possible to implement true interaction between calls and agents. One way around this problem was to keep track of the total level of experience and apply this as a multiplier to the service time. This is not as effective as truly modelling a newly trained advisor with a break down call, but it is suitable to reflect the general effect that new starters have on a group of trained staff – they affect more than their own calls, as they have to ask questions of other workers and will have a higher chance of needing to escalate calls to a manager.

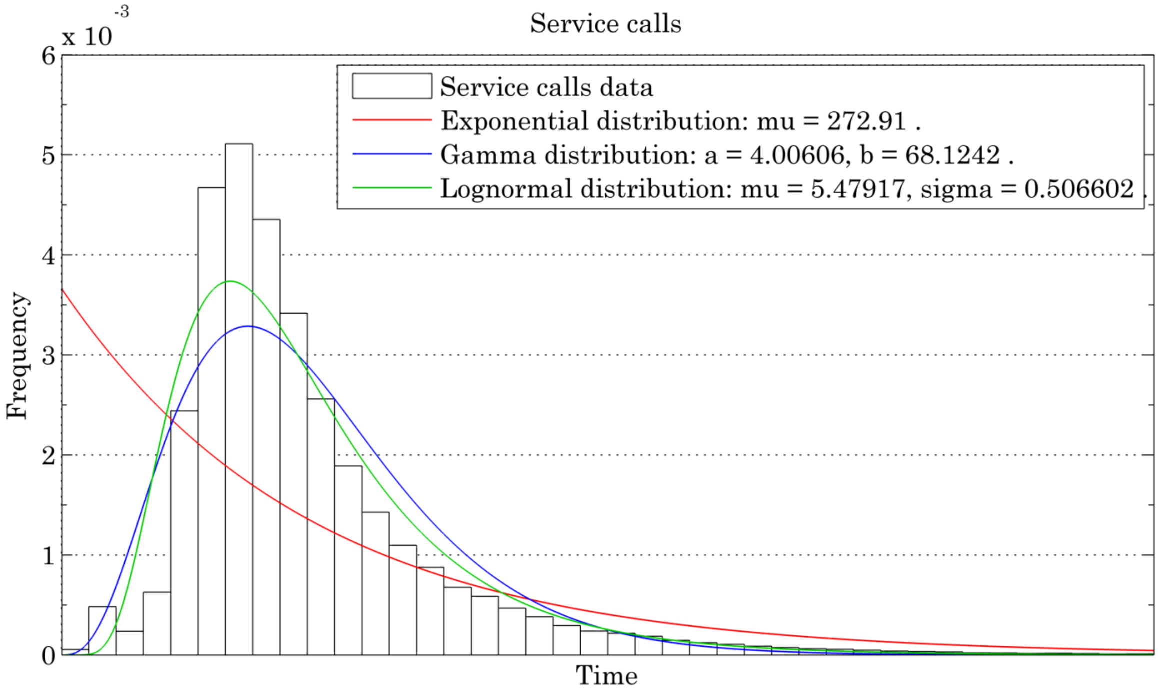


Figure 9 Potential Distributions for Service Time for Service Calls

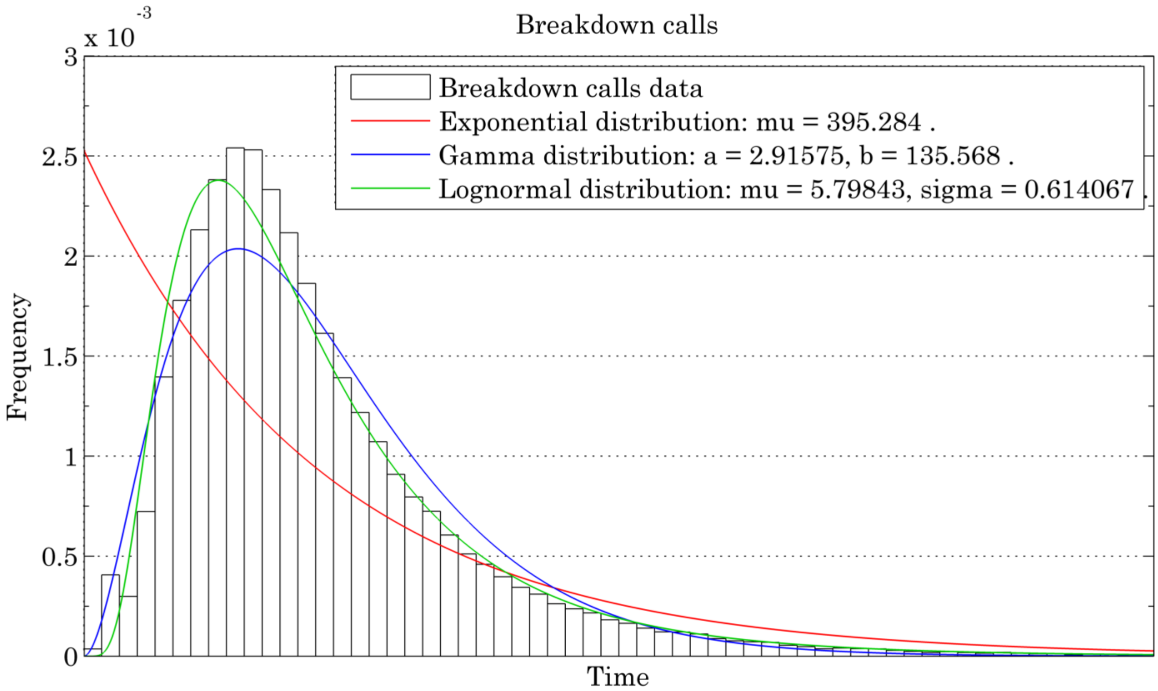


Figure 10 Potential Distributions for Service Time for Breakdown Calls

EVENTS

Events are objects that can be used to trigger actions based on a time, rate or condition being met. As the simulation was so reliant on the passage of time and the month of the year, events were used to trigger the updating of hourly, daily and monthly output variables and triggering the Recruitment Process.

For example, this is part of the code triggered every hour to updated the day versions of the outputs

```
/*hourly event*/

/*Update the Daily data output variables if the it is between 8am
and 8pm */
if (vHours >=8 && vHours<= 20){
/*the day variables are updated with the results from the previous
hour*/
	vDayTimeInSystem = vDayTimeInSystem + vHourTimeInSystem;
	vDayAbandoned = vDayAbandoned + vHourAbandoned;
	vDayAnswered = vDayAnswered + vHourAnswered;
	vDayAnsweredLess20 = vDayAnsweredLess20 + vHourAnsweredLess20;
/*The highest number of advisors that work that day is recorded so
that the calls per agent per hour can be calculated*/
	if (vHourCCAgentsInPool > vDayCCAgentsInPool) {
		vDayCCAgentsInPool = vHourCCAgentsInPool;
	}
}
```

The most complex part of this section was considering the timing on when and what number to take for the number of advisors. The number of advisors was set at the start of the hour but the number of calls answered is not known until the end of the hour. In this case it was important that the number of agents working was gathered before the `noOfStaffInPool` function. Next, when getting the number of agents that worked that day, the highest number of agents working at any hour that day needed to be recorded. If the advisors were instead assigned to shifts then this problem would not occur.

CALL CLASS

A Call class was created so that individual calls could be created with their own attributes. The most important attribute is `dblTimeTillAbandon`, which is assigned when the object is created by the source.

In call centre simulations, time to abandon is traditionally either ignored or assumed to follow an exponential distribution. Although it may appear visually from the bars that exponential is the best fit, a lognormal distribution actually follows the gradient more closely, as seen in Figures 11 and 12 from real-world data.

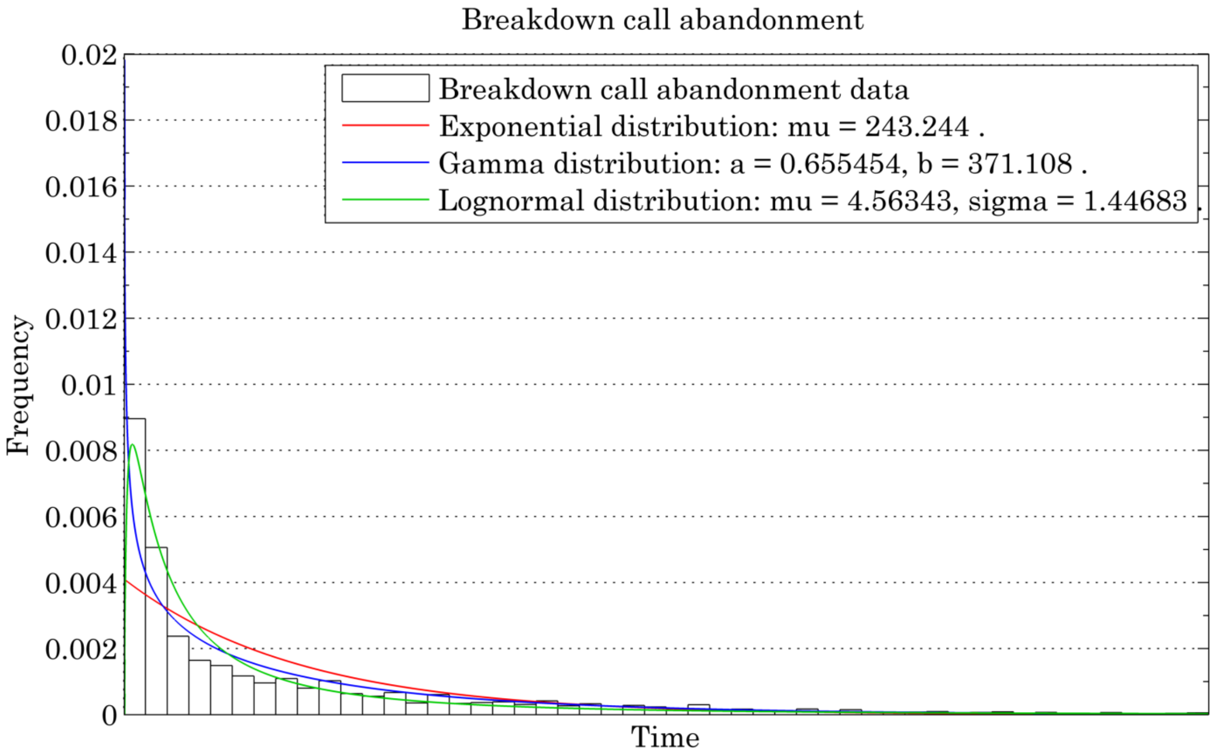


Figure 11 Breakdown Calls Abandonment Distribution

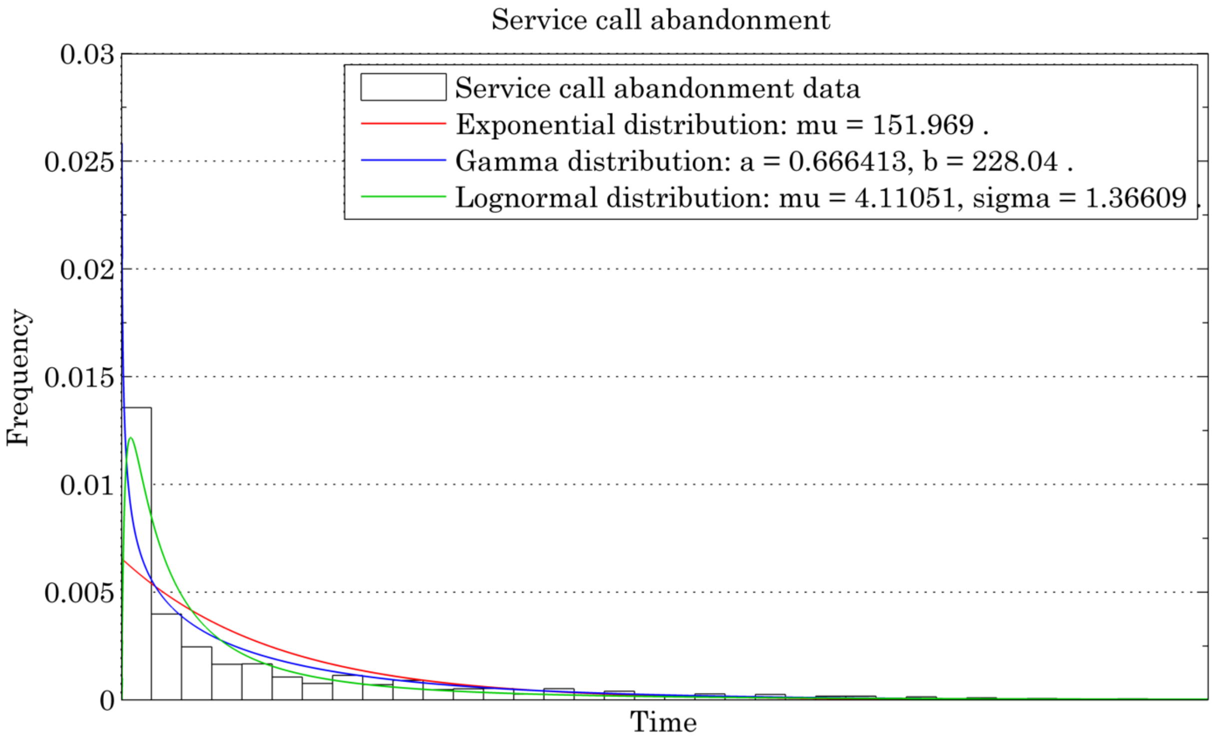


Figure 12 Service Calls Abandonment Distribution

Call Type	Time Till Abandon
Service	`entity.dblTimeTillAbandon = lognormal(4.11051, 1.36609, 0)/60;`
Breakdown	`entity.dblTimeTillAbandon =  lognormal(4.56343, 1.44683, 0)/60;`

A Time to Abandon is assigned is assigned as a random number from within the distribution when the entity is constructed. The proportion of calls abandoned fitted within the expected range from the scenario and so this was determined to be reasonably valid. The Rostering website did not include abandonment figures, so an assumption had to be made. A full investigation into the effects of different distribution rates can be found in the Experimentation section of the report.

It was first thought that customers calling in for breakdowns would be more patient than typical call centre work, so a high target for calls answered in 20 seconds target would not be overly necessary. However, although customer service Energy companies were accused of having long call waiting times of between 21 seconds to 17 minutes (Which, 2013), (suggesting that customers are willing to queue that long), most people will wait for less than a minute before abandoning the call, and an even shorter time when calling for a service.

These findings, and the results when the model was run with these figures, explained why calls waiting more than 20 seconds is an industry standard. It appears that this indicator is the cause and abandonment rate is the effect.

This was confirmed against historic reporting that showed when the average call waiting time was above [redacted] seconds, the abandonment rate would increase dramatically. This validated this choice of distribution and implementation.

This finding highlights the need for call centres to know key information about their call queues, as it would otherwise be assumed that customers who are more at risk should be prioritised when triaging, but in fact attending to less patient customer queues may be of greater benefit. This does not necessarily mean that a breakdown caller should have to wait 2 minutes and a service caller for 1 minute, but that when two calls of different types have been waiting for the same amount of time, the less patient queue type should be answered first. However, while this may improve abandonment rates, it has a significant potential for customer dissatisfaction.

CCAGENT CLASS

The CCAgent class reflects the call centre advisor. These entities are affected by the process at numerous times.

First, when generated, they are assigned an expected time of leaving. For permanent agents this is a period of a few years to reflect benchmarking studies (Research, 2004, p.39) with some advisors leaving sooner to reflect people at different stages of their career. For fixed-term agents a period was chosen to reflect short-term contract length. Three contracts lengths were considered:

- 3 months
- 6 months
- 9 months

One was chosen as it reflected the approach taken by the company and the other two after considering the patterns of historic winter weather. The 6 month version is included in the table below as the middle option as to provide an example. The max length of service was always set to one month greater than the contract length to reflect a company being flexible with its workforce.

Triangular distribution was used to assign the random length of Time to Leave because the data was not available. The min, max and mode were chosen based on the reply to the questionnaire, and triangular distributions are recommend when there is limited data (Borschev, 2013, p.567).

Once assigned, an agent would be assigned a new Time till Leave if their type changed from Fixed-Term to **Permanent.**

CCAgent Object	Time till Leave			Effectiveness
	min	max	mode	
Permanent	4 weeks	3 year	1 years	1
Fixed-Term Training	2 weeks	7 months	6 months	0.6
Fixed-Term Experienced	2 weeks	7 months	6 months	0.8

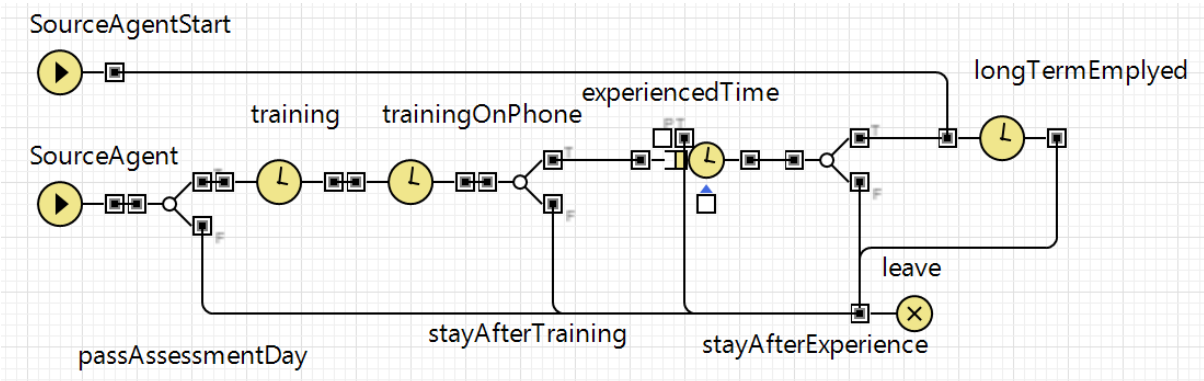
The other attribute that would change for CCAgents is the level of experience. As the advisor passed through the stages of training and time passes, they become more effective. The effectiveness level was then applied to the service times to reflect the level of experience within the current set of staff.

This is the area with least statistical foundation to the assumptions made, but the lack of publically available information meant that a broader approach was necessary to ensure that no confidential information was used.

These assumptions were based on the high level information provided in Appendix 1 – Interviews, Questionnaires and User Tests. Furthermore, the model is designed to only run for one winter at a time so it was not necessary to simulate long term advisor behaviour.

4.1.2 MODEL PROCESS

RECRUITMENT PROCESS



The simulation opens with a set of experienced CCAgents. The user sets this number and also the total number of staff they wish to have employed by mid October that the simulation will attempt to match. At the start of September the simulation starts injecting a randomly determined amount of new starters each week to represent assessment centres, the number was determined through a questionnaire to HR staff.

The advisor has a chance of passing the assessment centre and begins one week of basic training. There is no time here because this effectively represents behaviour that begins in August with the placement of adverts. Due to the low quantity of data available for this section of the process, it was important to simplify and emphasise the areas where more information was known.

After the first week of training the advisors receive on phone training and at this point are counted as a resource – though not very productive. After training some agents would leave and the rest would spend three months becoming experienced. During this time their effectiveness improves until finally some are kept on permanently and others complete their fixed-term contract. The advisors were able to timeout whilst becoming experienced or long term employed to reflect people leaving the company.

As stated before, call centres turnover is 5% higher than the national average and it was important to reflect that instability.

The following changes were made when performing the experimentation stage:

Length of Contract	Recruitment Starts	Recruitment Ends
3 months	September	February
6 months	September	November
9 months	September	November

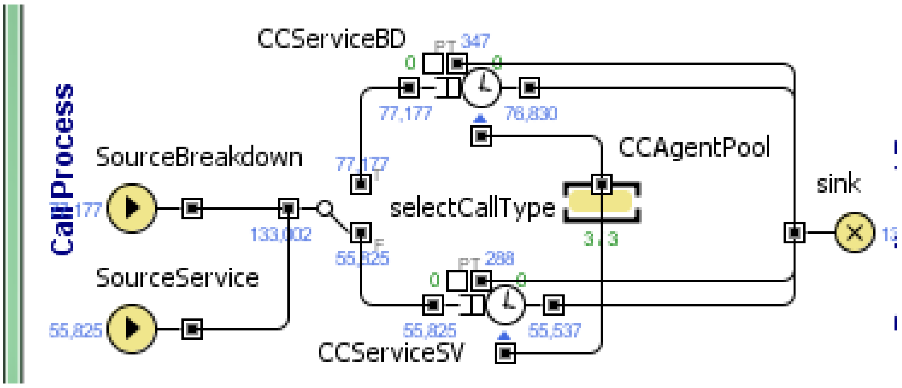
The recruitment event ran as follows once a week:

```
if (pNoCCAgentsAvailable < targetForMidOct && monthOverrun
>=[Recruitment Starts]  && monthOverrun <=
Recruitment Ends]  ) then
        int agentsToEmploy = uniform_discr(2, 10);
        SourceAgent.inject(agentsToEmploy);
        vNoOfAssessments++;}
```

The 6 and 9 month recruitment scenarios have a shorter recruitment phase to reflect the positive aspect of longer contract lengths that less time needs to be spent on recruitment. The 3 month contracts allow managers to be more flexible with the recruitment process, but needed a longer recruitment period to take into account the higher chance of staff turnover.

The number of assessments required was also logged, as it was a useful to feedback to the user how well the recruitment had gone during the simulation run. The simulation is limited in its reflection of reality; for example, it is possible for there to be recruitment needed every single week if the randomly generated number is always low. Although this run of bad luck could happen in reality, it is far more likely that a company would adapt by hiring agency staff instead or just running through the winter with the number of staff available.

CALL PROCESS



The Recruitment Process feeds into the Call Process through the setting of the number of units available to the resource pool that supports the service objects.

Following the conceptual model, calls are sorted according to their type, breakdown or service, and are directed to the correct Service Object. They wait there until either an Advisor is free or they reach their preset time-to-abandon, at which point they go to the sink and the object is destroyed.

This section went through the least number of adaptions after the first draft of the conceptual model. For experimentation only the breakdown source was used so that the Rostering website scenario could be emulated more closely as it suggested one queue.

4.1.3 EXPERIMENTAL DESIGN

CALL RATE AFFECTED BY WEATHER

It was determined early in the process that there was a relationship between the weather and the number of incoming breakdown calls. Real-world data was used to compare the proportion of the customer base calling in with breakdowns to the mean temperature. Mean temperature data was used from the Met Office (Hadley Centre, 2014) and is plotted against the call data in figure 13.

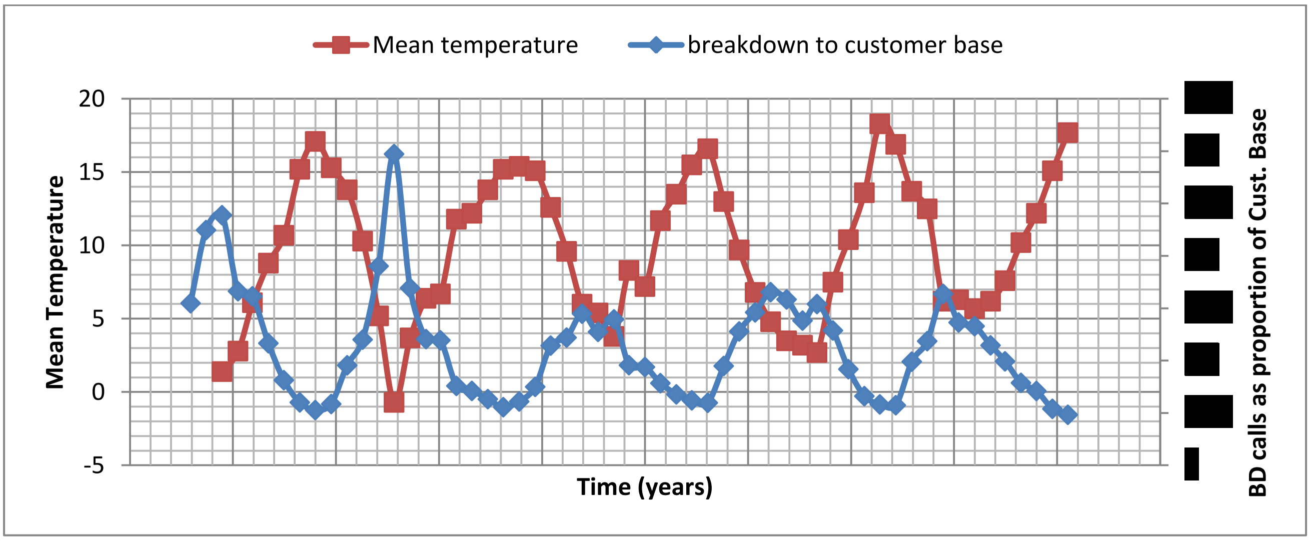
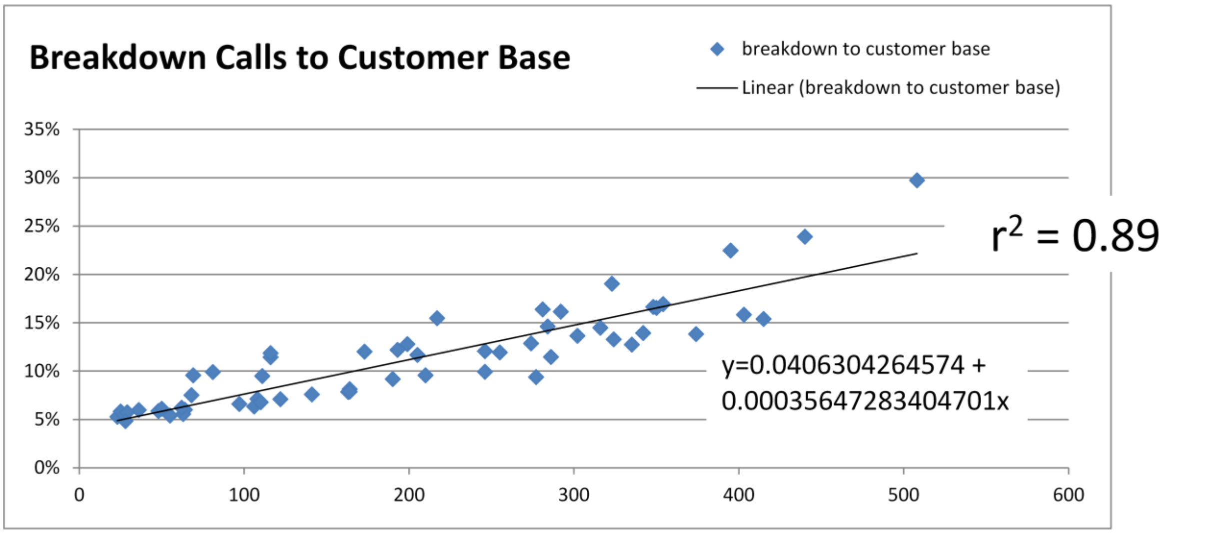


Figure 13 Breakdown Calls as Proportion of Customers Compared to Mean Temp.

The correlation coefficient between breakdown calls as a proportion of customer base with degree-days was calculated using Excel, giving an R2 value of 0.88971. The correlation with mean temperature was found to be slightly lower, at -0.88120.

Once a relationship was found between calls and degree-days then the relationship could be shown with an equation:



A function was written to handle the user choosing different severities of winter and apply the results are follows:

This section of code comes from the first page so that users got feedback as to the effect of their decision so the default month is set to October (9 rather than 10 and January is 0)

```
int month = 9;
int daysInMonth = 30;
int weatherProfile = initWeatherProfile;
```

The degree-day profile is modelled using the Gaussian function, which is the basis for the Normal distribution. The Gaussian distribution is given by:

$$f(x) = a\ \exp\left(-\frac{(x-b)^2}{2c^2}\right) + d$$

where:

Gaussian parameter	**Meaning**	**Variable**
a	Height of the peak	degreeDayDistr_peak
b	Month in which peak occurs	degreeDayDistr_month
c	Width of curve parameter	degreeDayDistr_width
d	Minimum level	degreeDayDistr_min

The user chooses the weather severity from a combo box. 0 is the lowest severity and is based on the mean degree-day data for the mildest five of the last ten years, 2 is the highest severity and is based on the worst five of the last ten years and 1 is the average of all ten years.

```
switch(weatherProfile)
      case 0 :
            degreeDayDistr_min = 10 ;
            degreeDayDistr_month = 13;
            degreeDayDistr_width = 2.2;
            degreeDayDistr_peak = 290;
            break;
      case 1 :
            degreeDayDistr_min = 20 ;
            degreeDayDistr_month = 12;
            degreeDayDistr_width = 2.2;
            degreeDayDistr_peak = 383;
            break;
      case 2 :
            degreeDayDistr_min = 25 ;
            degreeDayDistr_month = 11;
            degreeDayDistr_width = 2.2;
            degreeDayDistr_peak = 421;
            break;
      }
```

The Gaussian function is now implemented:

```
/*get degree-day*/
double degreeDayVal = (degreeDayDistr_peak-degreeDayDistr_min)
      *exp(
      -pow(month-degreeDayDistr_month,2)
            /(2*pow(degreeDayDistr_width,2)))
      +degreeDayDistr_min;
```

The degree-day relationship equation is applied:

```
/*get breakdown call total for month*/
double bdFrac = 0.0406304264574+(0.00035647283404701*degreeDayVal);
```

The service calls are added as 8% of the customer base to ensure every customer has a service:

```
feedbackMonth = (int)((initCustBase*bdFrac)+(initCustBase*0.08));
```

The monthly figure is divided by the number of days in the month to give average call per day:

```
double e = feedbackMonth/30;
feedbackDay = (int)e;
```

Figure 14 shows the change in degree-days from one month to the next. Even during the calmest of winters there is a high level of variation from January onwards. For example, Summer 2007 would be a curve if not for January when the temperature suddenly improves and then worsens.

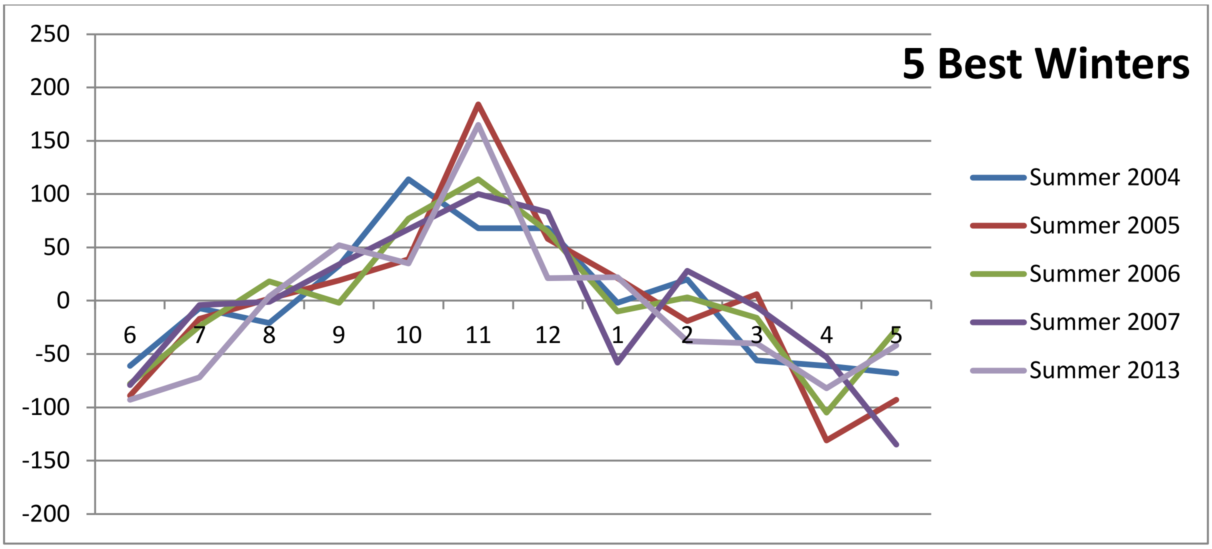


Figure 14 Degree Day Change Between Months

To reflect this variation, noise was added. This was worked out based on what provided realistic looking data. Figure 15 shows how results when calculated with Excel.

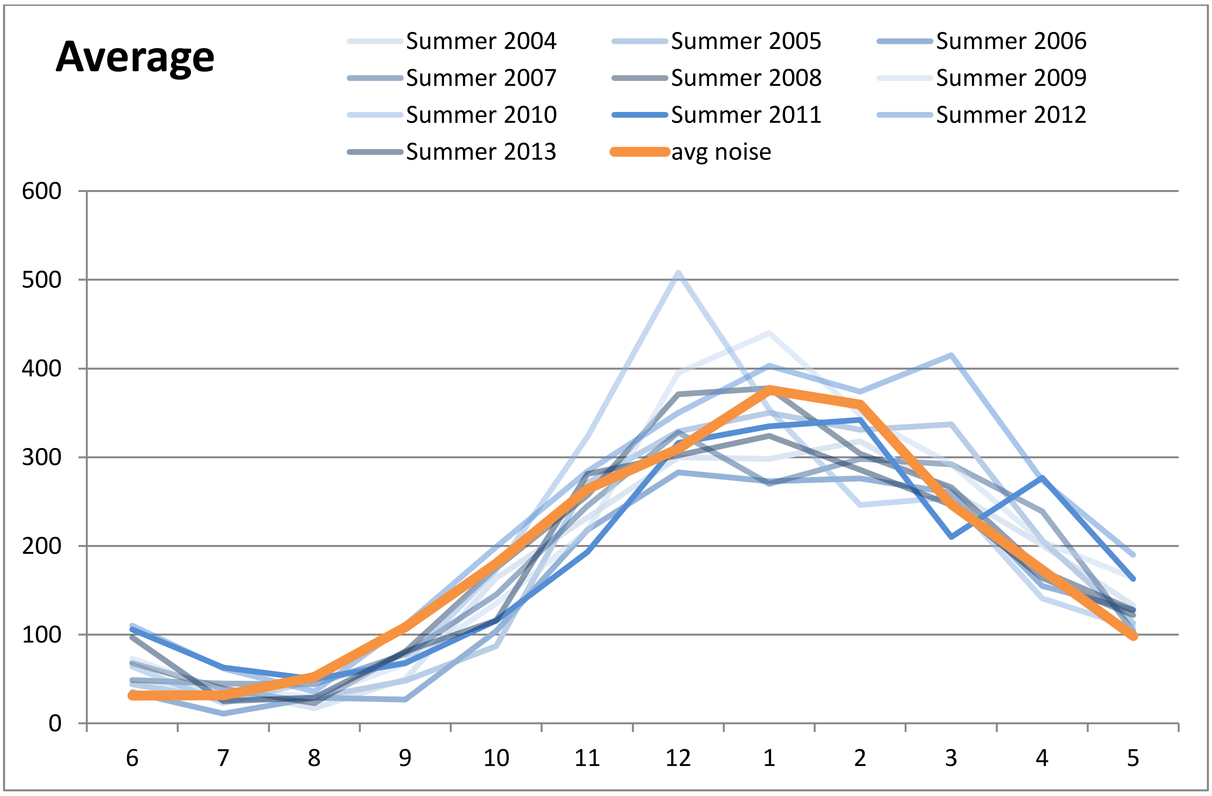


Figure 15 Average Weather Function - Excel

Figures 16 to 17 show the results of this work. There is room for improvement in increasing the volatility but generally the function is working as intended.

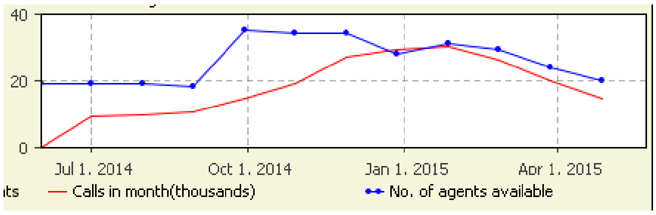


Figure 16 Severe Winter Simulation Calls by Month

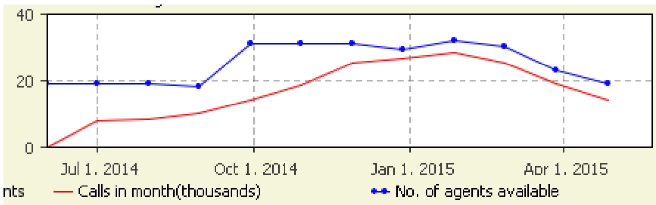


Figure 17 Average Simulation Calls by Month

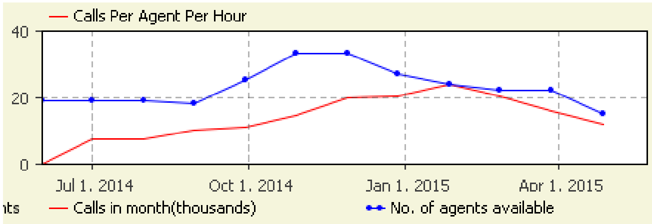


Figure 18 Low Simulation Calls by Month

SIMULATING RECRUITMENT PROCESS

The weather experimentation was designed to be a feature of the User Tool version of the simulation, due to the increased validity of the data and assumptions, and also because it was the most useful for contact centre managers. In comparison, experimentation with contract length was handled manually by changing the key variables in the objects of the simulation for each variation.

The recruitment process was run as a parameter variation of the simulation that can be accessed through the AnyLogic files ,but not through the applet.

4.1.4 VALIDATION

DIFFERENT TYPES OF VALIDATION

Validation "is the process of ensuring that the model is sufficiently accurate for the purpose at hand" (Robinson, 2004, p.209); because models are built to a specific purpose, validation is about how well it meets that purpose. Accordingly, there are several types of validation dealing with different aspects of the purpose.

Conceptual Model validation is about whether the conceptual model contained everything needed to meet the objectives (Robinson, 2004, p.210). Although the conceptual model presented in this report does meet this criterion, that is chiefly because it has been constantly revised during the implementation phase. For example, when it was not possible to get the resource object to use the CCAgents as entities in the time available then determining each individual agent's idle and working time became impossible, so the conceptual model was updated to reflect this. Due to delays with authorisation at the company meant that assumptions were checked informally with experienced contact centre managers along the way instead of formal interviews and circulation of the conceptual model for sign off.

Data validation is about ensuring that the data used to build the model is reliable (Robinson, 2004, p215). The data used for the temperature and degree-days should be reliable considering their sources, it is possible degree-days may have had a stronger relationship with a different base level. The real world company data could have been tested more, but it was sufficient to get the simulation made and generating reasonably realistic results. The model is suitably parameterised to allow future replacement of the input data if required.

White-box validation is micro level checks on the content of the model (Robinson, 2004, p.215). This includes verification that the object works as designed in the conceptual model and validation that it reflects the real world. It should be clear from the implementation chapter that objects were checked as they were created, that the output was as expected in line with the research and that the requirements of the conceptual model were met to a reasonable standard. Informal and formal conversations took place during the implementation stage when findings and assumptions were discussed. For example, the decision as to how to split the available staff across a week was discussed with a manager and it was agreed that the proportion of staff available per day was a good enough fit for reality when averaged over a year.

Black-Box validation is where the model is compared to the real world (Robinson, 2004, p.217). The Call Process looks to be close enough to reality to meet the objectives of the study. The Recruitment process, particularly advisor behaviour, could be improved, however again it is good enough for purpose. Though other statistical tests have been taken, the suggestion to compare results to real world data to a statistically significant degree of similarity (Robinson, 2004, p.218), has not been followed as the paucity of suitable comparative data makes this impossible.

Solution validation is where the suggested solution is applied to the real world and the data gathered to compare the changes to see if it matches the simulation results (Robinson, 2004, p.221). This is not often carried out and has not been completed as part of this project due to time constraints.

Experimental validation is about the necessary run-length and replications needed to provide trust-worthy data (Robinson, 2004p. 220). To carry out the experimental validation the static model was run 20 times for 200 days.

WARM-UP PERIOD[6]

Simulations that start with an empty world need time to get to normal operation – this is called the warm-up period (Robinson, 2004, p.141). Following Robinsons recommendation of a time-series inspection approach, the simulation was tested on the outputs of [7]:

1) calls through system, so could also check that the results were matching the Rostering website at the same time
2) calls per agent per hour – key output, likely to be very variable
3) average time in system – key output, likely to be more stable

The moving averages of the results were plotted and the results determined visually. Welch's method was also applied, where the moving average is plotted instead, to help smooth out the noise (Robinson, 2004, p.147).

[6] Warm-up period and replication working was heavily reliant on resources from module G54SIIM UK
[7] There is no sensitive data from this output so the results file with all of the corresponding statistical analysis can be found with the CD provided with the paper copy of this dissertation.

The average time in system and throughput were the most variable but both settled by 17 days. This means that the simulation should run for at least 10 x the run length i.e. 187 days (including the original 17 days) (Robinson, 2004, p.152). See figures 19 and 20.

As the length of the simulation was decided by the passing of the winter months, the warm up period was needed in order to know how far in advance of September to begin the simulation. It was decided to start the simulation in June as winter had always finished by May at the latest; this was more than enough time for the simulation to reach normal working. It was not possible to run this test on the seasonal version of the model because of the missing long term employment information; the simulation could not be run for more than one winter.

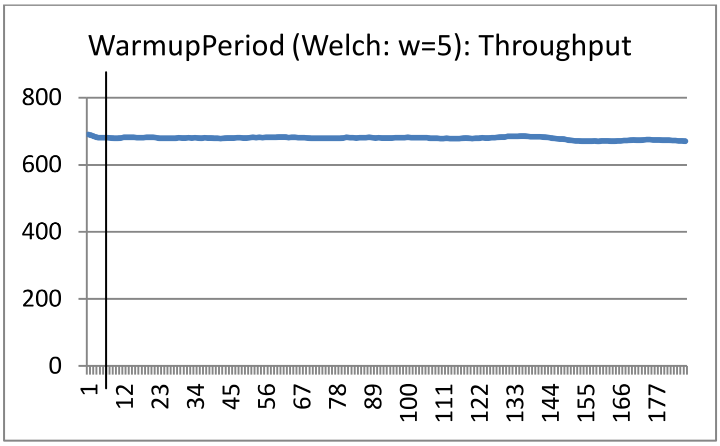


Figure 19 Throughput Warm-Up Period

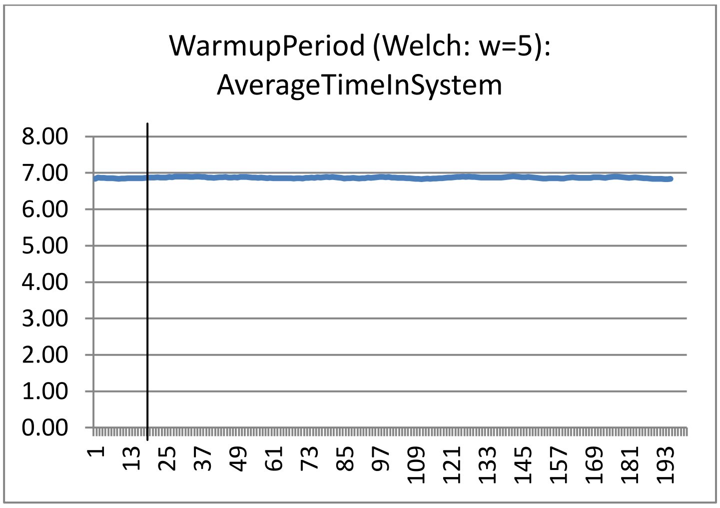


Figure 20 Average Time in System Warm-Up Period

NUMBER OF REPLICATIONS

The reliance on simulation for using random numbers means that multiple runs of the simulation are needed to get enough data for reliable experimentation (Robinson, 2004, p.152).

A graphical method was used, where the cumulative mean is plotted after removing the data from the warmup period (Robinson, 2004, p.152). When the line converges on a steady state then enough replications have been run. Here, calls per agent per hour and time in system where were used for replication checking. Figures 21 and 22 show that 10 replications were enough for this data; although the line may still look noisy, it is important to bear in mind the scale of the graphs. There is actually very little variation after the first couple of runs.

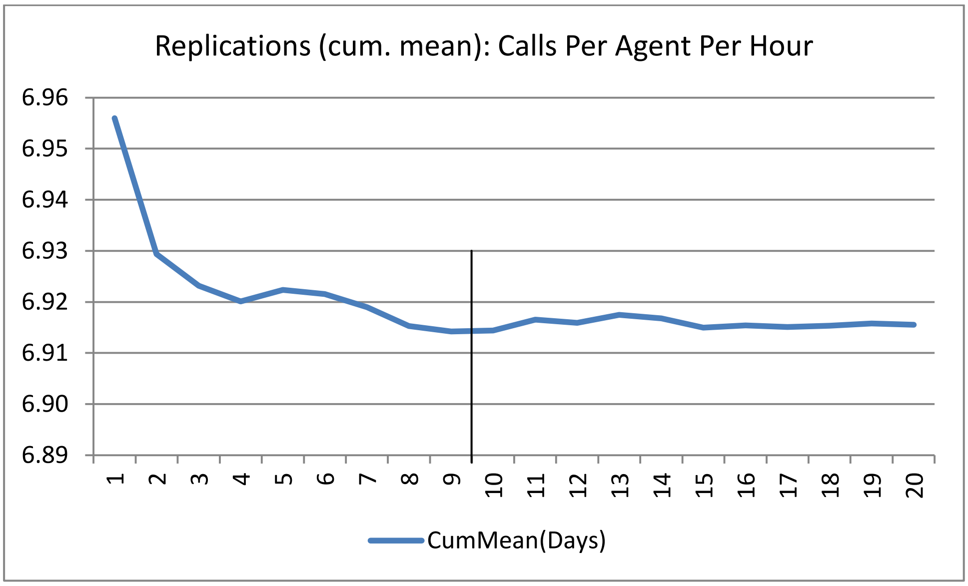


Figure 21 Calls per Agent Required Replications Graphical Method

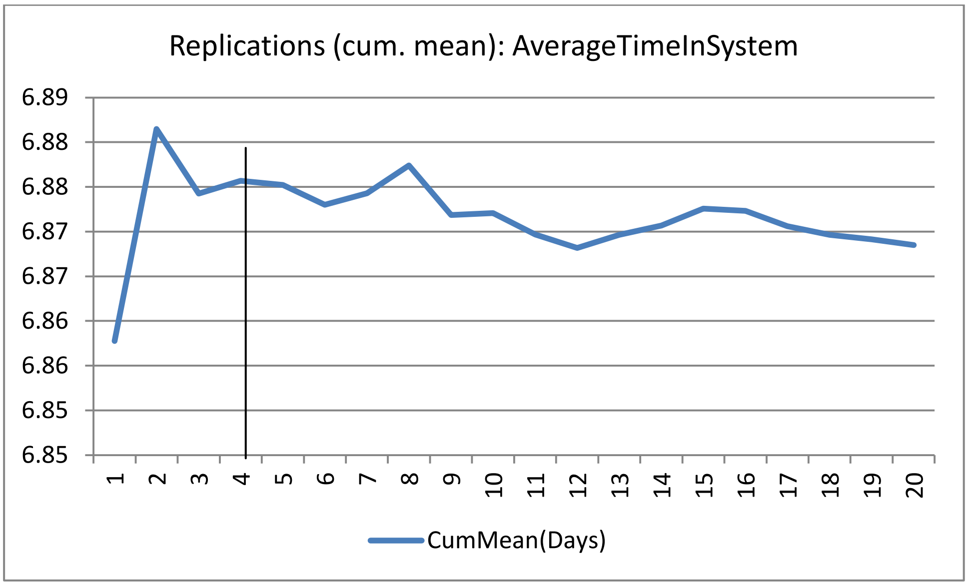


Figure 22 Average Time in System Required Replications Graphical Method

These results were confirmed by the confidence interval method which is a method "for showing how accurately the mean average of a value is being estimated" (Robinson, 2004, p.154). Accurate estimates are the narrower intervals. Figures 23 and 24 show that the 95% confidence interval was found after just one or two replications.

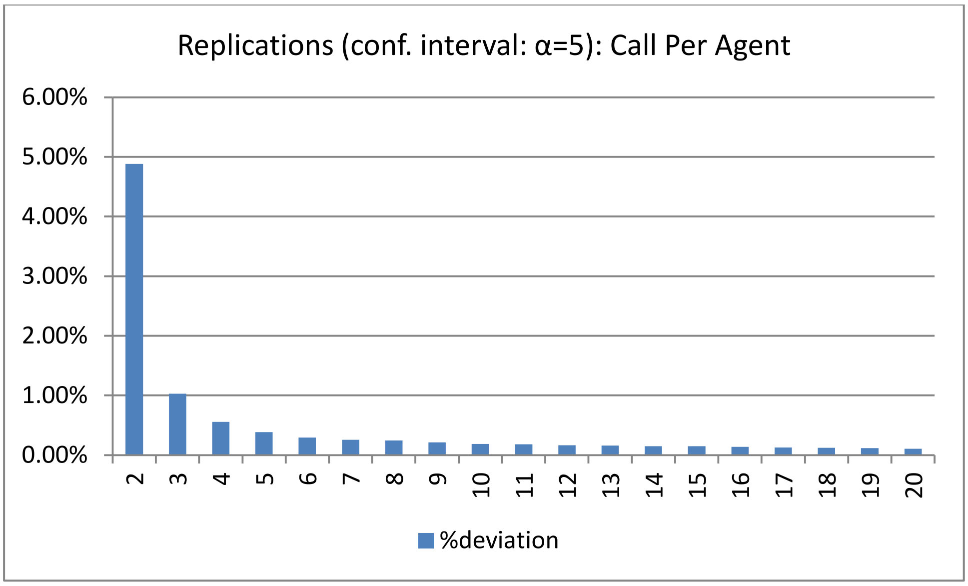


Figure 23 Calls per Agent Required Replications Confidence Interval Method

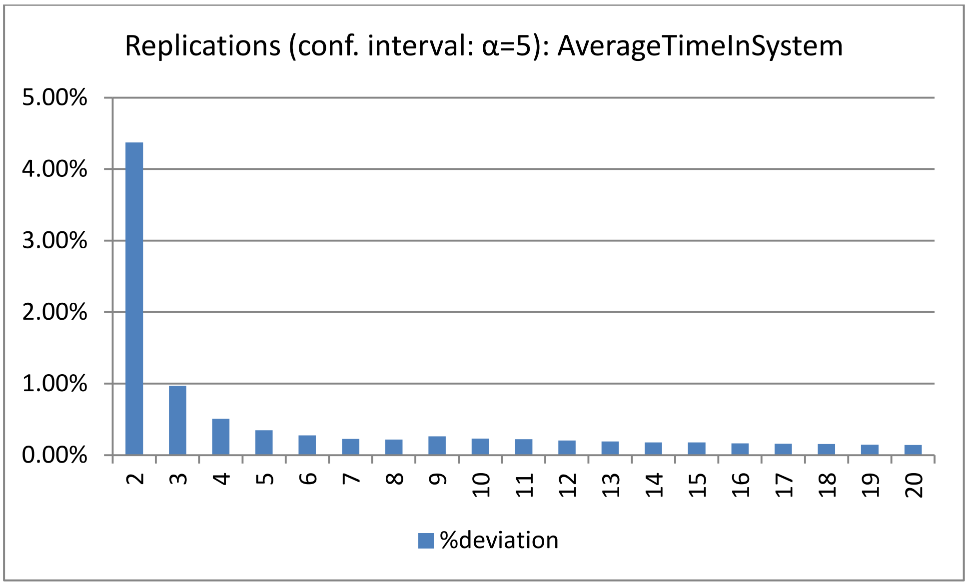


Figure 24 Average Time in System Required Replications Confidence Interval Method

CONCLUSION TO VALIDATION

A short run time and a couple of replications would be fine for the static model but it would be risky to assume that the seasonal model would be as stable. As the length of the simulation was decided by the scenario, and the whole run takes less than ten seconds to run then there were no technological or resource issues with having a long run length and doing lots of replications.

Therefore, the data for the experiments in the experimentation section was gathered from 20 replications and 305 days (not including a 30 warm-up period that was discounted). Of this the first 10 replications were used. This should mean that any variation due to the random number variables, as opposed to the experimental condition, has been removed.

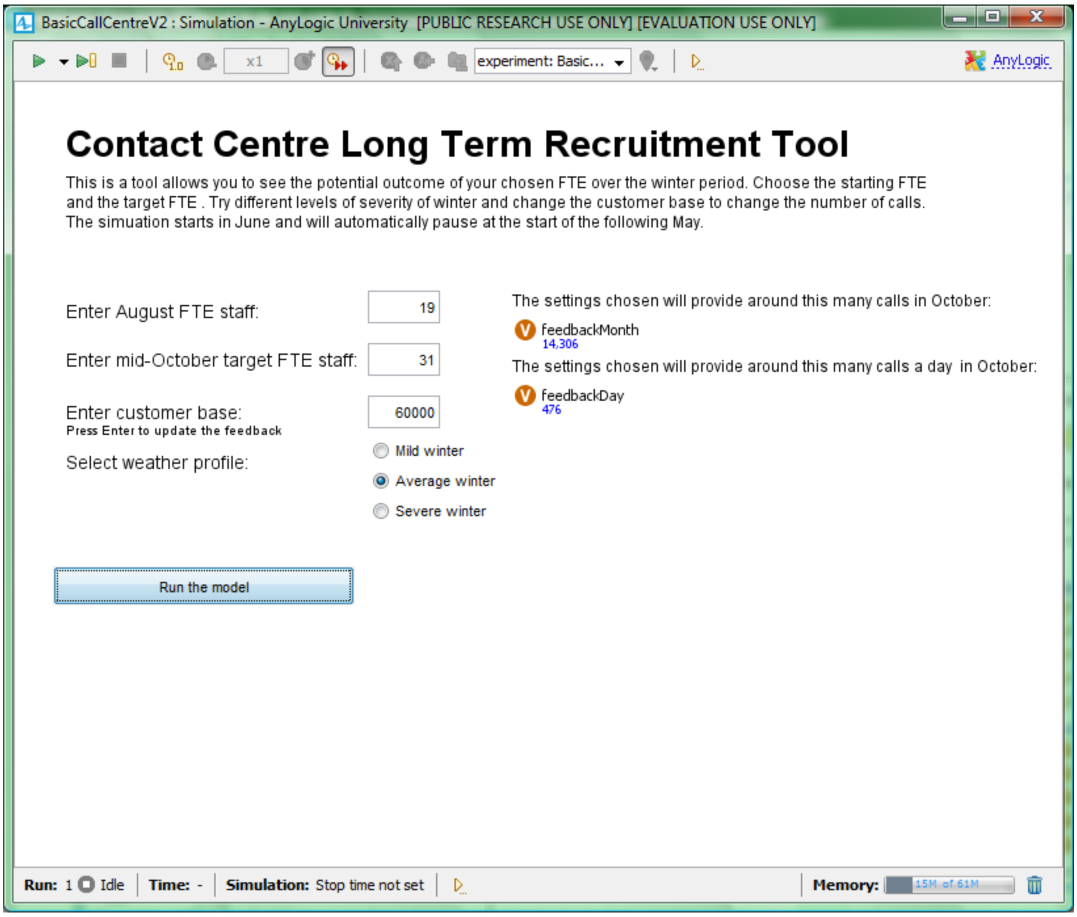
4.2 DESIGN

The Recruitment Tool was designed around Nielson's heuristic usability principles. These are (Nielson, 1993, p.20):

- **Simple and natural dialogue**: all information provided should be in a logical order and no irrelevant text should be used
- **Speak the users language**: the language should match the users
- **Minimise the users' memory load**: the user shouldn't have to remember instructions from one screen to another
- **Consistency**: words, situations and actions should always be identical when the same behaviour occurs in different locations
- **Clearly marked exits**: if something goes wrong then it should be easy to find a way to leave the system
- **Shortcuts**: experienced users should be able to navigate the system quickly, whilst novice's should not be left to get lost
- **Good error messages**: error messages should be clear, non-technical and offer solutions
- **Help and documentation**: should be concise and largely unnecessary

A user test was carried out with one participant and the findings are discussed with each associated section. The test and full results can be found in *Appendix 1 – Interviews, Questionnaires and User Tests*.

INPUTS

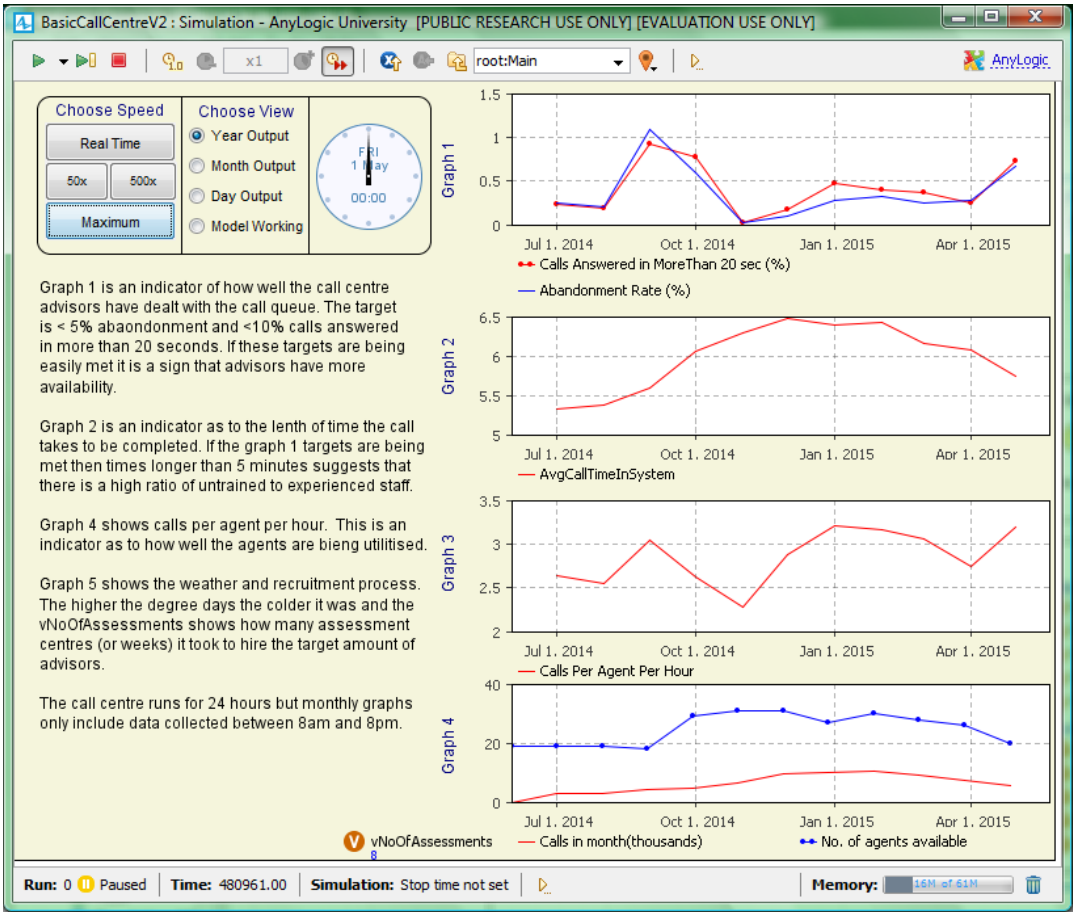


The user has four inputs to the simulation. The first two relate to the staffing levels, while the second two govern the incoming call rate. As requested, the inputs are simple. It is also in the language of the user – FTE means full time equivalent and is used instead of number of advisors because of part-time staff.

The positives of this screen are that it speaks the user's language (except for feedback which just used the AnyLogic default variable image for convenience) and the clear interaction flow through the inputs followed by the button to make the simulation run.

This style worked too well. During the user test phase the user moved so quickly through the task of setting the scenario that they missed that the feedback variables would change in response to the settings chosen. This could be fixed by moving the 'Run the Model' button to the right, so as to draw the eye across the feedback section and emphasise its importance. It could also be helped by changing the feedback variables into proper boxes and/or larger text.

OUTPUTS

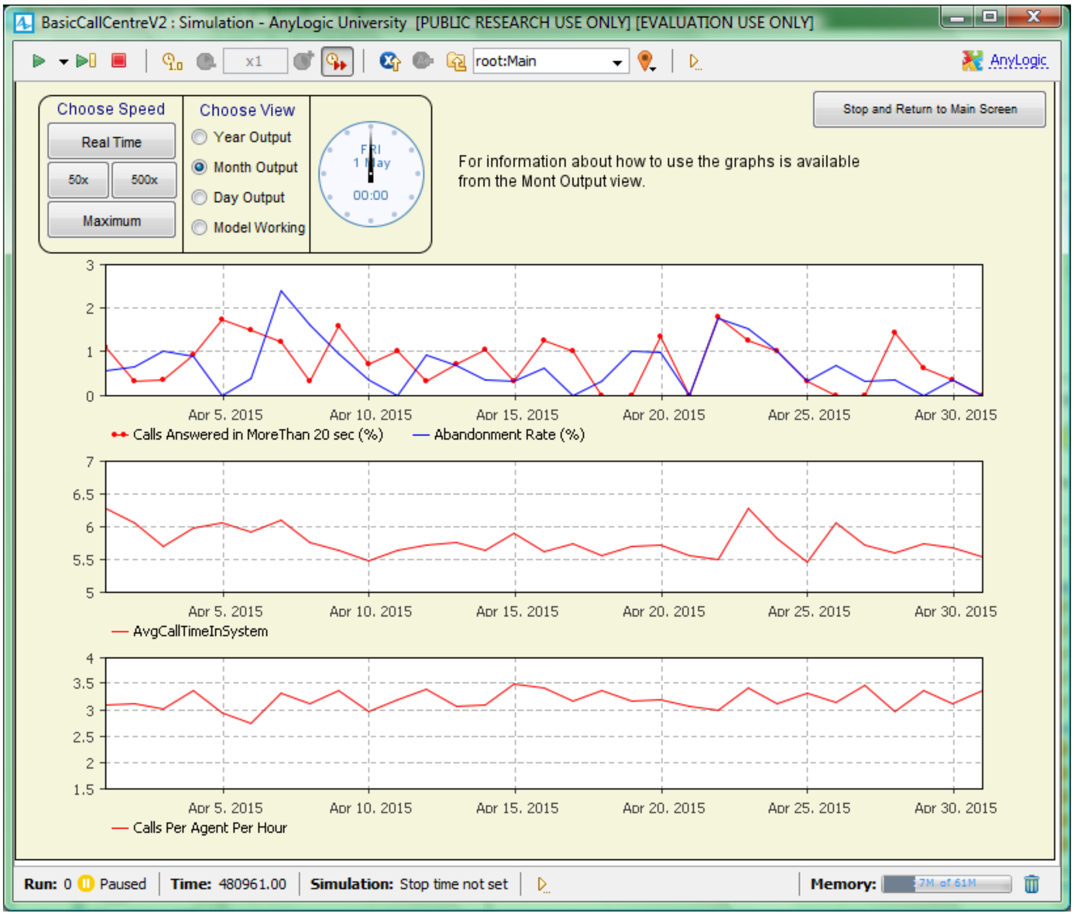


Once the model has been started the application defaults to the Year Output screen, as it is the most useful for experienced users who just want to try a scenario and see the results. This screen contains a lot of text that is necessary for new users to understand the graphs. The change to grey colouring was used to differentiate the output screens.

This worked during the user testing as the user became aware that the simulation was running (also emphasised by the clock). They avoided reading the text, but found it useful once they had. Also, some errors found were the lack of units associated with some of the graphs, making it difficult to put the results into context. A response to these issues could be information appearing when the curser hovers over an object.

For example, showing the calculation for outputs when the associated graph is selected. The ability to zoom in for more detail may also be useful.

One key oversight was not including the inputs selected by the user in any of the other screens. This directly contravenes the memory load principle as the user has to remember what was selected.



The monthly screen turned out to be unnecessary after meetings with stakeholders but this is an opportunity to discuss decisions made about the graphs and general navigation.

The Choose view button always defaults to the current screen, this serves two purposes:

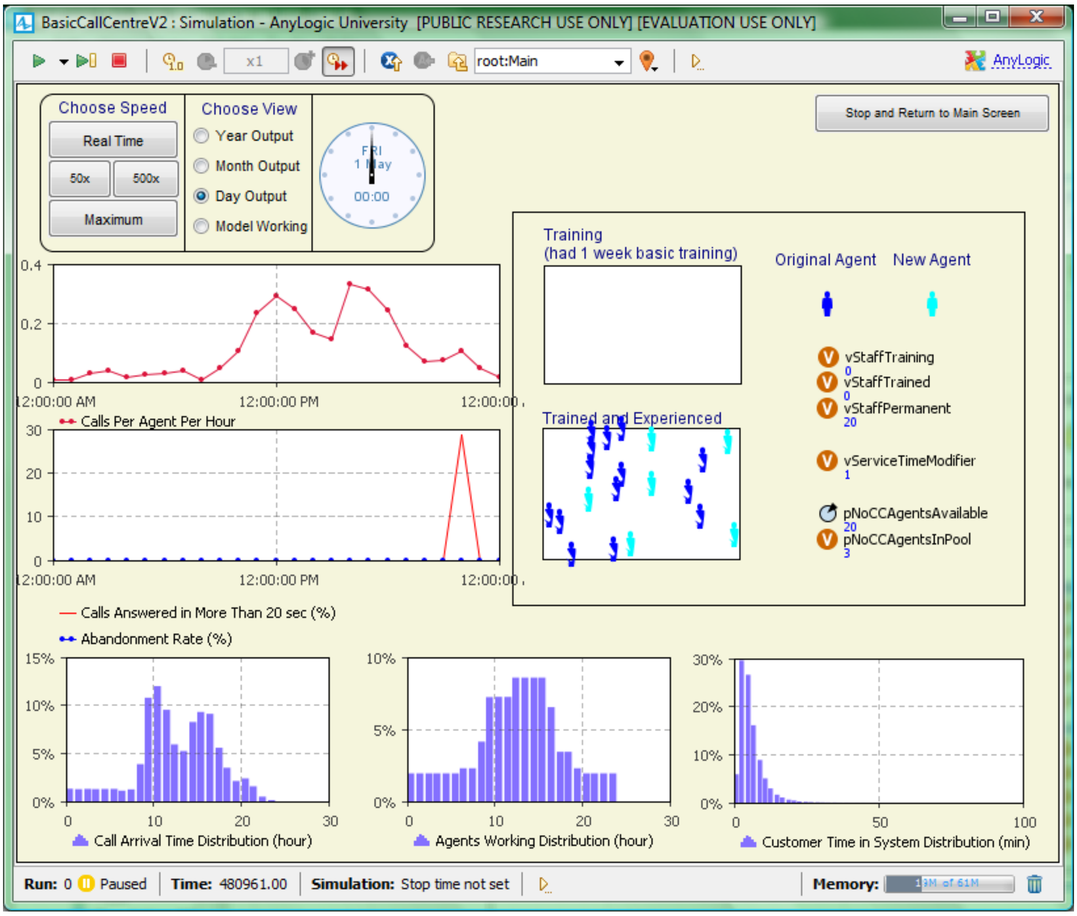
1) it shows where the user is without wasting time on a title
2) it shows the user what other options are available

During the user test the participant did look at the one other screen but they did not feel the need to change screens during the first part of the test as they thought they could get all the information they wanted from the year output screen, however, when it was necessary to change screen they had no questions and made no errors.

The colours used on the graphs should be accessible to people who are colour-blind. The application does not have features such as screen reader compatibility but at some least steps were taken during the design phase to take into account those with sight difficulties. This is also the reason why one line will always have markers plotted when there are two outputs on one graph.

A clear design error should be visible from this screen as the clearly marked exit button is not available from the default year output view. This was caused by trying to fit all needed information onto that screen.

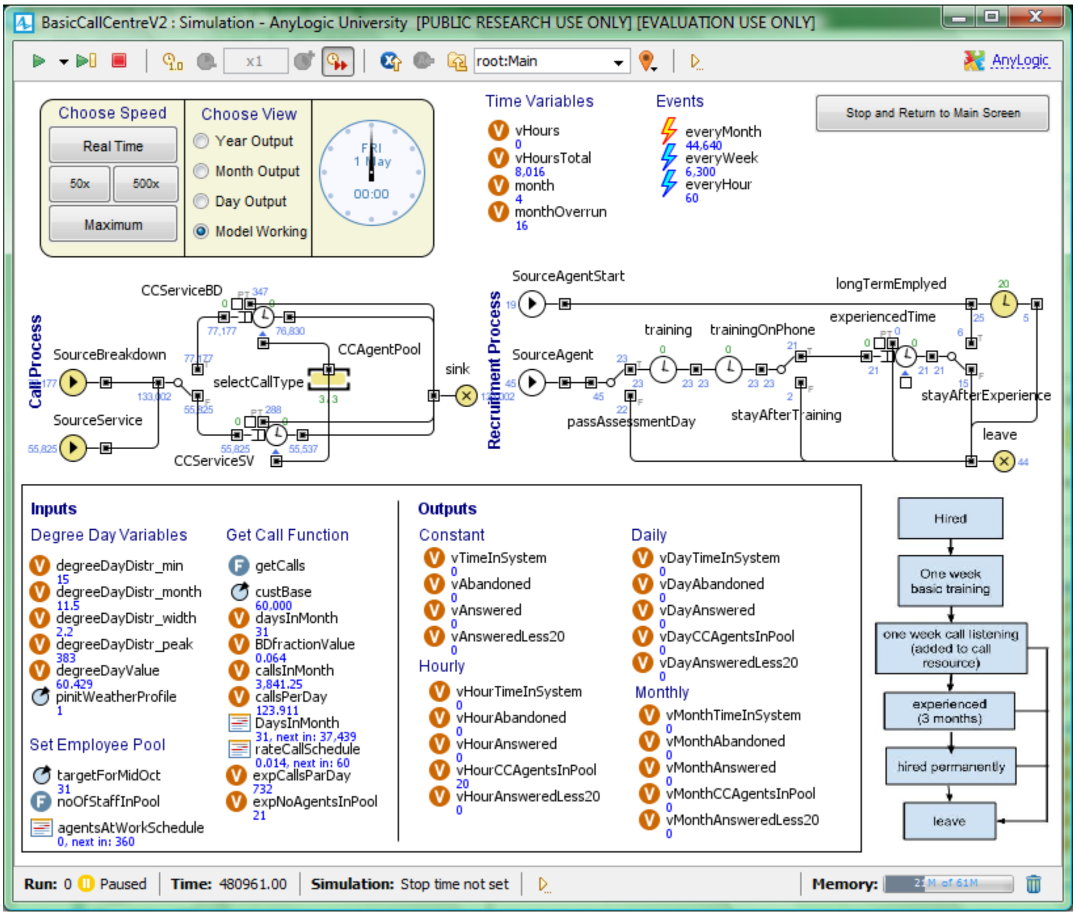
Finally, this screen shows the relationship between calls answered in 20 seconds and abandonment rate as was discussed in the validation section.



The final output screen is the Day Output. The core reason for this screen is to help the manager feel more confident of the simulation results. This worked to a disadvantage during the discussion with the call centre manager as they wanted a closer representation of the companies daily staffing distribution.

The 12AM 12PM graphs are also difficult to read but the movement of the staff from one type to another is a simple but effective way to visually show the transition through the recruitment process.

THE PROCESSES



The flow chart was added to the Model Working screen to help the user understand the simulation assumptions. In practice, this screen is of limited use to the contact centre manager end user as it is technical in nature, but it is useful as part of the system documentation, particularly for maintaining the code in the future.

5. EXPERIMENTATION

Two experiments were carried out. Abandonment distribution was tested with the static model to see what effect, if any, a distribution had on the simulation outputs. Variants on the length of contract were tested with the aim of providing a staffing recommendation to the company.

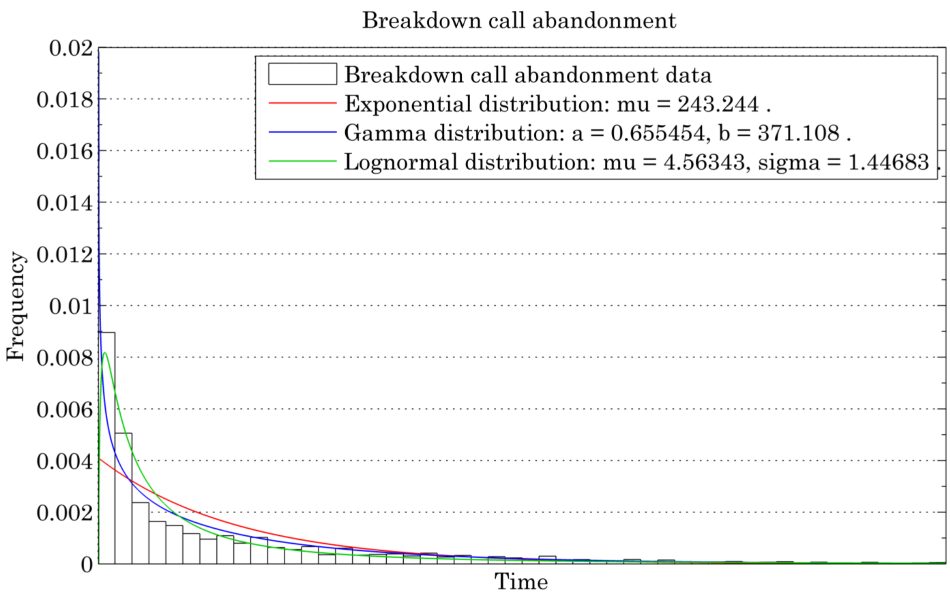
5.1 ABANDONMENT DISTRIBUTION

The different types of distributions were compared by running a “parameters version” of the static model, using an AnyLogic feature which allows a simulation to be run to investigate the effect of variations in parameters. Each distribution was run 10 times and the first 30 data points were disregarded. The results were compared using Student’s Paired t-test, as seen in Figure 25. Four distributions were chosen:

- Lognormal because it was indicated as the best fit from MATLAB, it also matches the Brown et al study as discussed earlier
- Gamma because it is the closest match to Erlang, which is heavily used in all aspects of call modelling
- Exponential because it was described as being used in the literature review section of the Mehrota study
- Triangular because it is the standard distribution when there is very little data to work from

There was a statistically significant difference found for each of the distribution types, as seen in figure 25. This means that modellers should ensure that they are using the correct distribution for the data as each option could have an unintended effect on the simulation outputs. The findings also mean that the distributions can be ranked in terms of severity, so a more if there is no choice but to use a triangular distribution then it can be inferred what sort of impact this would have on the model output.

It is not surprising that the distribution has such an impact when looking again at the graphical representation of the distributions (printed here again for convenience). As can be seen in the figure the exponential distribution clearly does not adequately represent the distribution at low time values. Gamma distribution is a much closer fit but its extremely high peak at zero time is inaccurate. The log normal distribution is the best compromise and captures the initial peak without extreme values.



	n				10									
	alpha				0.0083									
	t-value at n-1, a/6				3.3642									
	Results				**log v exp**		**log v gamma**		**log v triangle**		**exp v gamma**		**exp v triangle**	
Replication	**lognormal**	**exponential**	**gamma**	**triangle**	**D**	**SD cmpnt**	**D**	**SD cmpnt**	**D**	**SD cmpnt**	**D**	**SD cmpnt**	**D**	**SD cmp**
1	13.2%	11.5%	12.8%	8.4%	1.6%	7.73E-08	0.3%	3.73E-06	4.8%	2.71E-06	-1.3%	4.88E-06	3.1%	3.70E-
2	13.5%	11.5%	12.5%	8.4%	2.0%	1.32E-05	1.0%	2.43E-05	5.1%	3.91E-06	-1.0%	1.67E-06	3.2%	2.76E-
3	13.0%	11.5%	12.9%	8.1%	1.5%	2.16E-06	0.1%	1.53E-05	4.9%	8.22E-07	-1.3%	5.97E-06	3.4%	3.19E-
4	13.2%	11.9%	12.6%	8.1%	1.4%	6.56E-06	0.6%	2.90E-07	5.1%	1.63E-06	-0.8%	9.60E-06	3.7%	1.47E-
5	13.3%	11.6%	12.7%	8.3%	1.8%	2.20E-06	0.6%	1.22E-06	5.0%	2.57E-07	-1.1%	1.43E-07	3.2%	9.52E-
6	13.2%	11.6%	12.7%	8.2%	1.7%	4.60E-07	0.6%	2.58E-07	5.1%	1.49E-06	-1.1%	2.90E-08	3.4%	2.95E-
7	13.1%	11.5%	12.4%	8.1%	1.5%	5.31E-07	0.6%	1.19E-06	5.0%	5.82E-08	-0.9%	3.31E-06	3.4%	9.40E-
8	13.4%	11.7%	12.7%	8.4%	1.8%	1.98E-06	0.7%	3.71E-06	5.1%	1.46E-06	-1.0%	2.68E-07	3.3%	3.91E-
9	12.7%	11.7%	12.8%	8.1%	1.0%	3.35E-05	-0.1%	3.81E-05	4.6%	9.81E-06	-1.1%	1.53E-07	3.6%	7.04E-
10	13.3%	11.4%	12.6%	8.4%	1.9%	9.37E-06	0.7%	3.71E-06	4.9%	5.60E-07	-1.2%	1.29E-06	2.9%	1.45E-
	Dbar					0.01614		0.00518		0.04943		-0.01096		0.033
	Total (X1-X2-Dbar)^2					7.00E-05		9.19E-05		2.27E-05		2.73E-05		4.53E-
	SD					0.00279		0.00320		0.00159		0.00174		0.002
	Lower CI					**0.01317**		**0.00178**		**0.04774**		**-0.01281**		**0.030**
	Upper CI					**0.01911**		**0.00858**		**0.05112**		**-0.00911**		**0.035**
	To a confidence level of 95% every one distribution has a statistically significant difference													

Figure 25 Comparison of Abandonment Distributions

5.2 LENGTH OF FIXED TERM CONTRACT

Three lengths of contract length were tested through the seasonal model and the outputs subjected to the Student t-distribution test of statistical significance. Again the simulation was run 10 times and the first 30 data points were disregarded for each scenario.

Calls answered less 20 seconds	Low weather severity	Avg. Weather severity	High weather severity
3 month contract	93.1%	90.8%	89%
6 month contract	92%	89.7%	88%
9 month contract	93.3%	89.7%	88.2%

As expected severe weather always has a negative impact on targets and although it looks like the 3 month contract may be slightly better, no statistical significance found.

Calls per agent per hour	Low weather severity	Avg. Weather severity	High weather severity
3 month contract	4.3	4.6	4.8
6 month contract	4.4	4.7	4.9
9 month contract	4.3	4.7	4.9

3 month contracts were found to be statistically significantly worse than both 6 and 9 month contracts during an average winter.

Avg. time in system (minutes)	Low weather severity	Avg. Weather severity	High weather severity
3 month contract	6.01	6.14	6.2
6 month contract	6.01	6.11	6.18
9 month contract	6	6.13	6.2

9 month contracts were found to be statistically better for service times than 3 month contacts, but no different than 6 month contracts during an average winter.

In conclusion 9 month contracts give time for advisors to become better trained over the winter period and this may result in better service times for customers. However, 3 month contracts are more adaptable and allow managers to lose staff quickly if a winter turns out to be mild. A full cost benefit analysis is recommended as it is suggested that contract length has an effect on contact centre key performance indicators.

Although these figure are based on rough estimates of data, that the contract length has an impact on the outputs means that this should be considered as the assumed distributions are as likely to have underestimated the difference between contracts as overestimated.

6. RESULTS AND RECOMENDATIONS

6.1 MAIN MODEL

The results of the simulation fall into different categories:

The data collection and analysis phase showed that the different types of queue were made of customers with different levels of patience and with problems that took different amounts of time. This was useful for the implementation of the model, but also emphasises the importance for call centres to collect this type of information and act on it by prioritising different types of calls with call routing software. The implementation of the weather function also means that the company has an equation that can be directly applied to call forecasting, no matter which method is used. The simulation itself can be used as a tool to help inform decision making and thinking through the implications of FTE decisions over the winter period and the experiments suggest that contract length should be considered as a way to improve meeting KPIs.

Fitting into the current work on simulation, this project showed the importance of trying different types of distributions and replicated some of the findings from other studies that have been based on real-world data. It also gives an indication of the challenges faced for the modelling of long term scenarios with interacting highly variable processes.

6.2 USER DESIGN

In line with all usability and design principles, applications are more usable and understandable when simple to use. This tool has limited functionality when in applet form but it can still be very useful. From this it would be possible to implement a completely original and standalone piece of software, and though this would provide more control over form and function, the ability to make the tool more complex would not necessarily be the best decision.

6.3 LIMITATIONS

There are key limitations to this project. Firstly the degree-day research showed that different demographics use heating in different ways. If the make-up of the customer base were to change then the equation defining the relationship between breakdown calls and degree-days should be revisited.

Similarly to other models, this simulation does not reflect real customer behaviour at abandonment. It assumes that the customer leaves the call queue and does not call back. It would be better to model this relationship properly as working with the data suggests that this is a key reason why peak call times during the day expand and contract, but that intuition is for another study.

Finally, as discussed at length during this dissertation, more category A recruitment and employee life-cycle data would allow for a simulation with a higher level of data validity.

7. CONCLUSION

This project has conducted a literature review of current call centre staffing forecast tools and found a lack of support decision support of long term options. It was determined that simulation may offer a solution to this problem and a discrete event simulation was created.

The process was useful because all stages of the modelling process from conceptual modelling, data collection and implementation provided the company with a fresh way of assessing processes and providing recommendations to improve performance.

Through conducting experiments it was found that distributions have statistically significant effects on outputs and models of all types that assume distribution type from other studies should be wary and rely more heavily on real-world data. Software packages make it very easy to try multiple distributions quickly and conveniently and simulation designers should include distribution support when choosing a methodology for the model implementation.

Degree-days have also been shown to be a viable data source as an alternative to mean temperature for modelling weather effects.

8. FURTHER STUDY

In terms of modelling call centre schedules, more work needs to be carried out into abandonment rates. This information could be used to turn this simulation into a full agent-based solution. The benefit of this may be a more realistic model or a simulation that allows for the evaluation of behaviour not possible within the limits of discrete event simulation. For example, consider a customer abandoning a call, does that make their level of patience lower when they call in again meaning that peak periods are made worse by not meeting customer needs on the first call; or an advisor that starts work during a particularly busy week, how does this affect their chance of leaving if the centre is not fully staffed?

In terms of software, all of the objects, created for this project or available from the default AnyLogic software could be turned into a standalone application. This project has identified the variables required for a long term model which could be fully realised as a profitable piece of software, especially of it also handled weekly advisor scheduling.

9. REFLECTIONS

From this project I have learnt that it is important to keep things simple. It’s the only way to build a simulation in a short time period and it helps the end-user to follow the assumptions made. It was challenging to keep the stakeholders involved when assumptions had to be made when people were not available but this was accomplished building in such a way that changes could be made without too much time being lost or by explaining that it was not possible to meet the request and manager the stakeholders’ expectations.

My aim was to create something that would be useful in a way that would improve my own skills. Although it has been a challenge to complete this project, and of course improvements can be made, I am happy that I have met this personal target.

REFERENCES

A. T ERNST, JIANG, H. & KRISHNAMOORTHY, M. 2004. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research,* 153**,** 3-27.

BORSCHEV, A. 2013. *The Big Book of Simulation Modelling: Multimethod Modelling with AnyLogic 6*, AnyLogic North America.

BRITISH GAS. 2014. *Contact Telephone Numbers* [Online]. Available: http://www.britishgas.co.uk/help-and-advice/contactus-personal-details/contact-us.html [Accessed 6/07/2014 2014].

BROWN, L., GANS, N., MANDELBAUM, A., SAKOV, A., SHEN, H., ZELTYN, S. & ZHAO, L. 2005. Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective *Journal of the American Statistical Association,* 100**,** 36-50.

CCA GLOBAL. 2014. *FAQs* [Online]. Available: http://www.cca-global.com/gsx/content/learningzone/?contentid=199 [Accessed 11/08/2014 2014].

CFA. 2012. *Contact Centre Operations Labour Market 2012* [Online]. Business Skills @ Work. Available: http://www.oph.fi/download/145729_Contact_Centres_LMI_2012_.pdf [Accessed 4 July 2014 2014].

GOVERNMENT, S. 2003. Benchmarking and Definitions Winthin Contact Centres. *In:* GOVERNMENT, T. S. (ed.). The Scottish Government.

HADLEY CENTRE 2014. HadCET diagnostics and data for the current and previous years. *In:* CHANGE, M. O. H. C. F. C. (ed.). http://www.metoffice.gov.uk/hadobs/hadcet/graphs/index.html: Met Office.

HARRIS, S. 2014. *A Primer On Call Center Staffing Methods* [Online]. Portage Communications. Available: http://www.portagecommunications.com/index.php?option=com_content&task=view&id=24&Itemid=37 [Accessed 11/08/2014 2014].

MEHROTA, V. & FAMA, J. Call Centre Simulation Modelling: Methods, Challenges and Opportunities. 2003 Winter Simulation Conference, 2003 The Fairmont New Orleans, New Orleans, LA.

MEIERA, H. & REHDANZB, K. 2010. Determinants of residential space heating expenditures in Great Britain. *Energy Economics,* 32**,** 949-959.

MYRON, D. 2002. *A Day in the Life of a Call Center* [Online]. CRM.com. Available: http://www.destinationcrm.com/Articles/Editorial/Magazine-Features/A-Day-in-the-Life-of-a-Call-Center-47993.aspx [Accessed 3/08/2014 2014].

NIELSON, J. 1993. *Usability Engineering*, Margan Kaufman: San Francisco.

NPOWER. 2014. *Conact Us* [Online]. Available: https://www.npower.com/home/help-and-support/contact-us/#bills [Accessed 14/07/2014 2014].

PEER-OLAF SIEBERS, U. A. 2011. A First Approach on Modelling Staff Proactiveness in Retail Simulation Models. *Journal of Artificial Intelligence and Societal Simulation,* 14**,** 1-25.

PORTAGECOMMUNICATIONS. 2014. *Call Center Workforce Management Software for Small and Medium Sized Call Centers* [Online]. Available: http://www.portagecommunications.com/ [Accessed 28/08/2014 2014].

RESEARCH, C. 2004. Best Practice In Call Centres 2004.

RESOURCE ANALYSTS. 2014. *Call Centre* [Online]. Resource Analyists: Shift Pattern and Implimentation. Available: http://www.rostering.co.uk/workload-analysis-call-centre [Accessed 20/06/2014 2014].

ROBINSON, S. 2004. *Simulation: The Practice of Model Development,* Chichester, John Wiley & Sons Ltd.

SCAFAIO, C. Defining a World-Class Customer Service Call Center. DVIRC.

SHANNON, R. E. Introduction to the Art and Science of Simulation. *In:* D.J. MEDEIROS, E.F. WATSON, J.S. CARSON & MANIVANNAN, M. S., eds. 1998 Winter Simulation Conference, 1998 Grand Hyatt Washington, Washington DC.

VESMA, V. *Degree day data* [Online]. Degree Days Direct Limited. Available: http://www.vesma.com/ddd/welcome.htm [Accessed 3/08/2014 2014].

WEISSTEIN, E. W. 2014. *Erlang Distribution* [Online]. MathWorld--A Wolfram Web Resource. Available: http://mathworld.wolfram.com/ErlangDistribution.html.

WESTBAY ENGINEERS. *Call Centre Calculator Help* [Online]. Available: http://www.erlang.com/calculator/call/help.html [Accessed 20/08/2014 2014].

WHICH. 2013. *Energy companies putting customer service on hold* [Online]. Available: http://www.which.co.uk/news/2013/07/energy-companies-putting-customer-service-on-hold-325177 [Accessed 1/07/2014 2014].

APPENDIX 1 – INTERVIEWS, QUESTIONNAIRES AND USER TESTS

HR QUESTIONNAIRE

1. How long does it take from the point of getting permission to hire to run an assessment centre? **Estimate x xxxxxx**
2. What sort of percentage of agents interviewed at an assessment centre get hired and accept the position? **Aprox xx**
3. What is the shortest period of time a new hire will stay for? **xxxxxx**
4. What is the longest time a new hire will stay for? **Normally FTC so hard to say. We keep about xxxxxx**
5. How long do most agents stay? **xxxxxx**

Any other information you think may me relevant? **Notice that the take up drops when we advertise FTC as against permanent**

TEAM MANAGER 1 INTERVIEW

Team Manager 1 Interview

The session began by showing the manager a very early draft of the simulation. The answers given to the questions have been summarised and agreed with the participant.

1. How do you measure how a contact centre (CC) is successful?

Targets are set for the contact centre as a whole and for individual agents. These include: calls per hour, wrap up times [where the customer has left the call but the agent has time to finish writing up notes], hold time [how long the customer is placed on hold during the call], personal time; and productivity [time agents are able to make or receive calls] is generally at **xxxxxx**

2. What is the system for hiring a large number of agents?

Assessment days are used to hire a number of agents at a time. It can be difficult to get enough applications but pre-assessment day tests can improve the number of successful applicants. xxx potential applicants are invited and xxxx may be successful. However, there can be low turnout and assessment days organised can have as few as 2 applicants attend. Assessment days will be spread out to improve the chances of getting enough applicants.

3. How long does it take an agent to become experienced?

New agents are given xxxx weeks training: xxxxx of classroom training and xxxxx call listening and answering easier calls. About xxx months of work are needed before the agent becomes experienced. The level of experience will depend on the individual and their performance, this will include feedback on current performance and any errors made to improve skills.

4. What effect does an inexperienced agent have on CC measures?

Inexperienced agents will have: longer hold times, manage less calls per hour, customers will have longer call wait times, and more call swill be escalated [escalations are where a customer asks to speak to a supervisor]

Not all customers will escalate when speaking with a new hire, this can occur with tenured agents also, with new hires this can be where they have had less experience of how to manage a complaint call or how to explain a solution/outcome confidently, this does come with experience of having to manage different calls.

5. What is the shortest, longest and average that a CC agent will stay at work for?

Very few agents leave during their training period. Agents on fixed term contracts may leave between xxxxx months. Permanent agents will stay for about a year, though some stay for longer.

We may still have experienced agents on a rolling/FTC, this is when we haven't made positions permanent and still need to have a good level of resourcing.

6. Here is a version of a CC simulation – what information would you want to try to change?

Keep it as simple as possible. I would like to be able to set the starting amount of FTE [Full time Equivalent] and the target FTE. I also want to know if I can change the number of calls per day in case the number of customers changes.

7. What experience do you have of managing CC staffing and performance?

Over 5 years team manager experience

CONTACT CENTRE MANAGER INTERVIEW AND TOOL FEEDBACK SESSION

The session began with the researcher explaining the assumptions behind the model and giving an example of using the simulation. The answers given to the questions have been summarised and agreed with the participant.

1. So what are your first thoughts?
    - Looks good
    - Like the graphical feedback
    - would like more information about the staff and call distributions
2. [Returned to the Day Output screen and used graphs to explain assumptions and how staff and calls were distributed through the day] Does that answer your questions?
   Yes, though I'd prefer to see a version that had data I know is accurate to this company, I understand that isn't possible at the moment though.
3. So what are the positives?
    - The explanations and the graphs
    - the clear front page
    - Daily and yearly pages are the best
4. What can be improved?
    - The name – I was expecting it to tell me how many people to hire and when
    - Data targeted to this business
    - Easier to understand model process
5. Can you see a use for this tool?
    - Definitely, not for say-to-day use but when thinking about hiring levels
6. While undertaking the model I found the following… [show abandonment data and show how point out abandonment and calls answered in 20 seconds graphs]… how does this sound?
   I'm surprised the relationship is that close, but it does match with the impact of new current new starters that are working service only queues. That is definitely something I want to pursue. Can I have a copy of the tool to play with?
   [link to runTheModel.com provided with instructions as to how to use]

USER TEST

One test was carried out to check the placement of buttons. The participant was chosen by convenience and is inexperienced with call centres but experienced with computer software.

The investigator's script was as follows:

"I'm going to present you with a scenario. Please follow the instructions; I will make notes as we go."

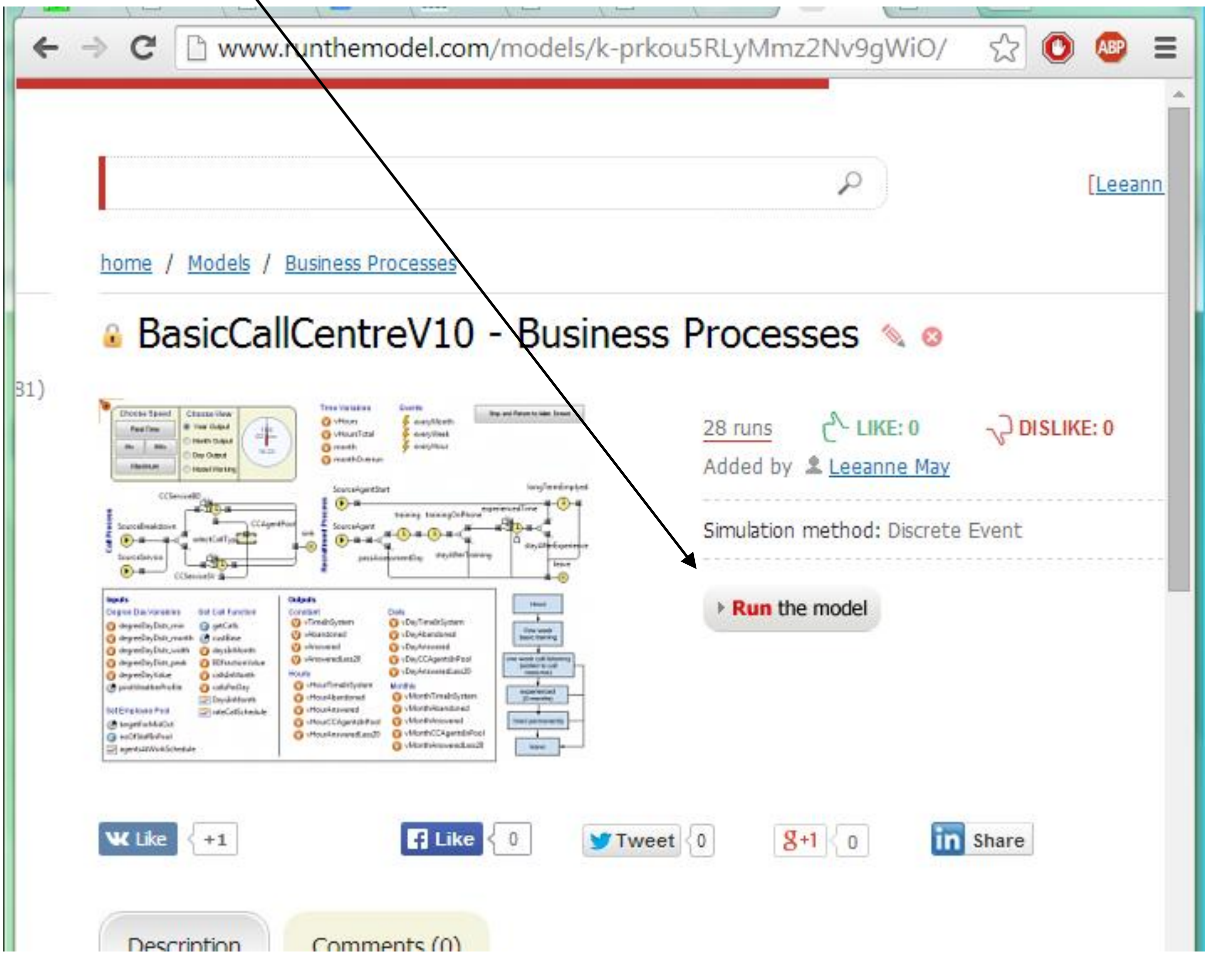
Action	What happened	User comments
Start with 15 FTE	Immediately added figure to correct box	
Target is 20 FTE	Immediately added figure to correct box	
Customer base is 100,000	Immediately added figure to correct box	
Can you tell me what the average calls per day is?	Clicked run the model box before the question was finished and so couldn't answer	"Sorry – I didn't think there was anything else on that page"
Run the model at full speed	Looked at navigation group and chose correct button	
What do you interpret from the graphs?	Studied the current page, did not attempt to use any other pages	"I can see how the number of call and staff increased in October but I think we may have hired too early because the number of calls per person drops suddenly. I suppose all of the targets look like they've been met – hang on what does this say, oh so that was wrong, none of the targets have been met after October. I don't know what the units of the average time in system so I'm not really sure what to do with that "
Can you get back to the front page?	Looks at page, clicks on Day Output button and then clicks on the Stop and Return to Main Screen button	"Not sure… cool there it is. Oh and there are the calls per day"

APPENDIX 2 – HOW TO/DOCUMENTATION

RUNNING THE MODEL FROM RUNTHEMODEL.COM

This is the most likely version to work without having Anylogic v7 software. It is recommended to use Chrome or Firefox and to check the troubleshooting section first.

1. Go to http://www.runthemodel.com/models/k-prkou5RLyMmz2Nv9gWiO/
2. Click Run the Model



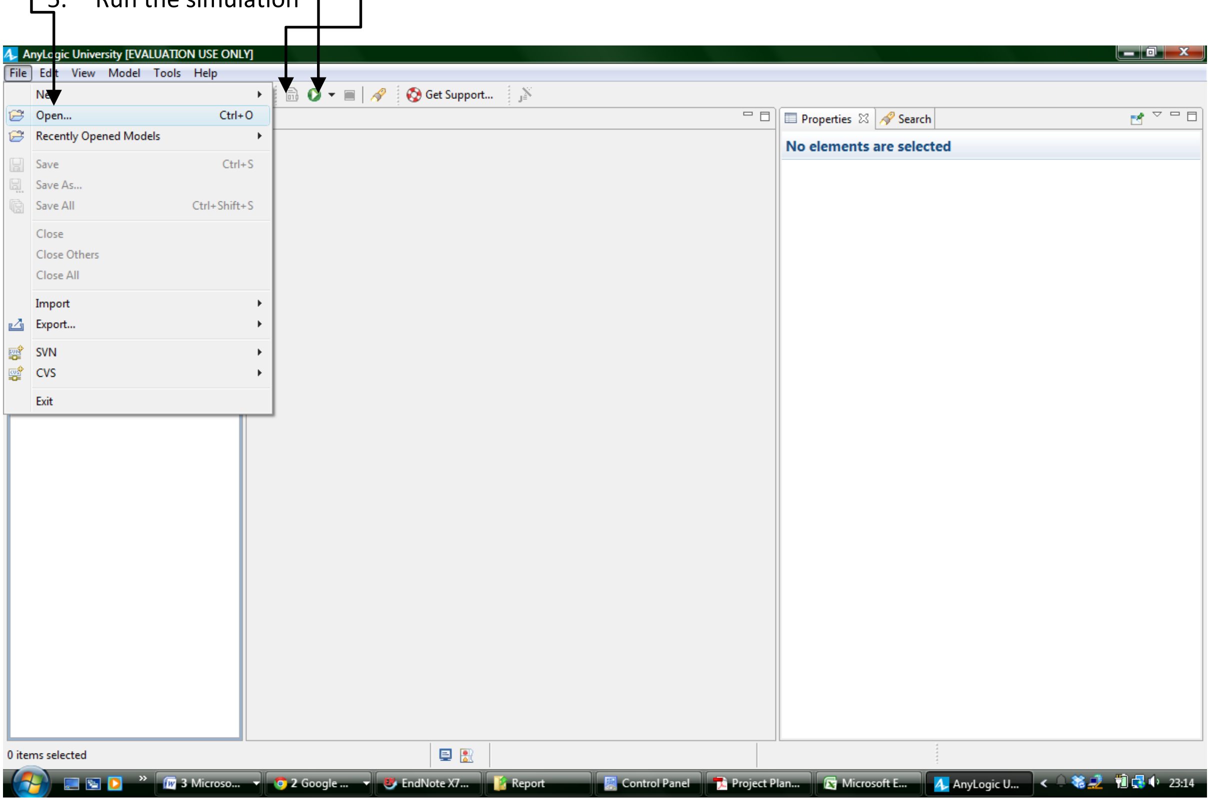
3. The model will then run. For problems with Java see Troubleshooting

USING ANYLOGIC

A trial version of AnyLogic 7.0.3 can be downloaded for free from http://www.anylogic.com/downloads

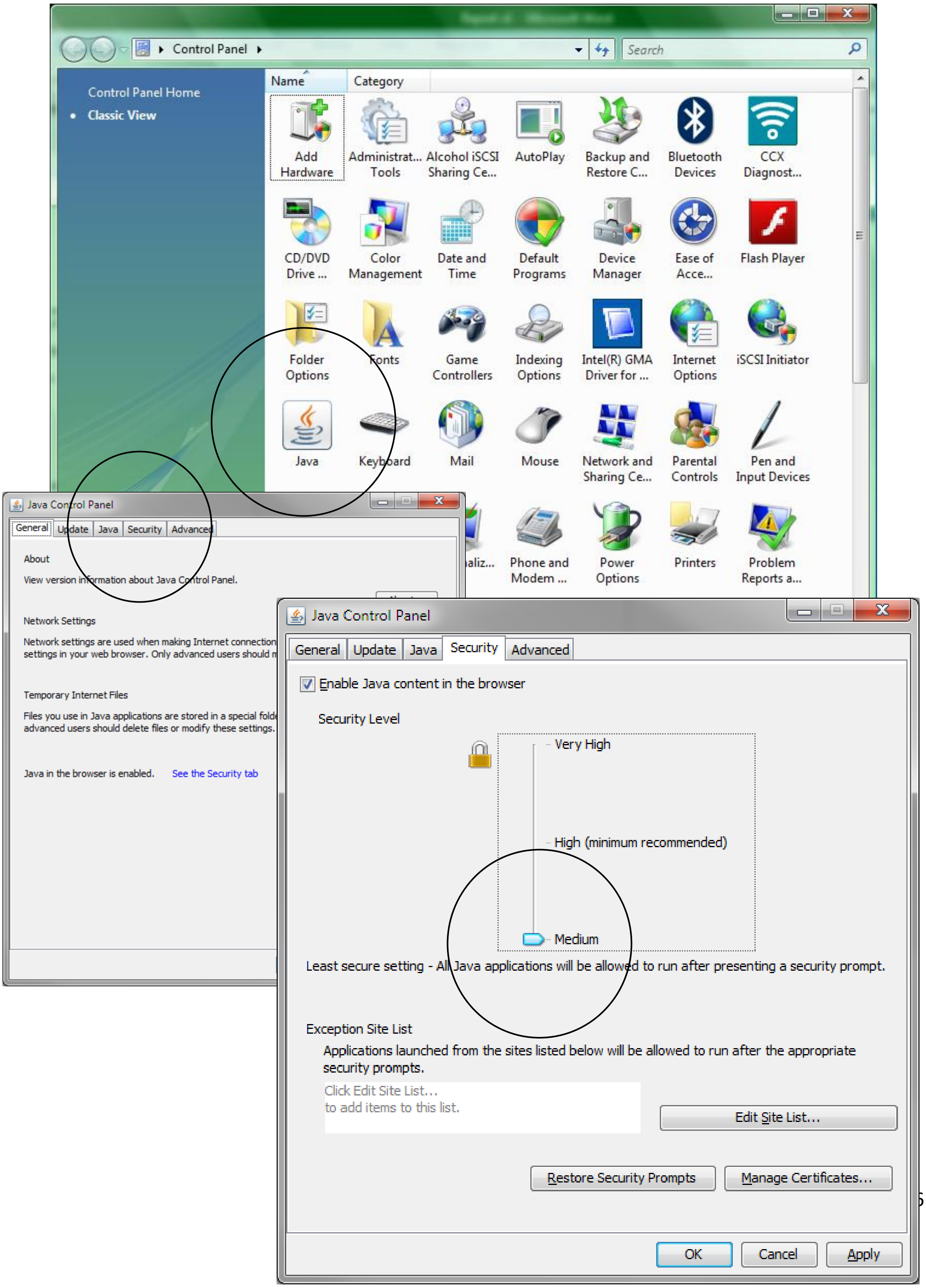
Choose the version for your operating system and follow the Activation Guide instructions

1. Once operational, open AnyLogic
2. Close the welcome page
3. Open the model files from the CD provided with the paper version of this dissertation
4. Compile the simulation
5. Run the simulation



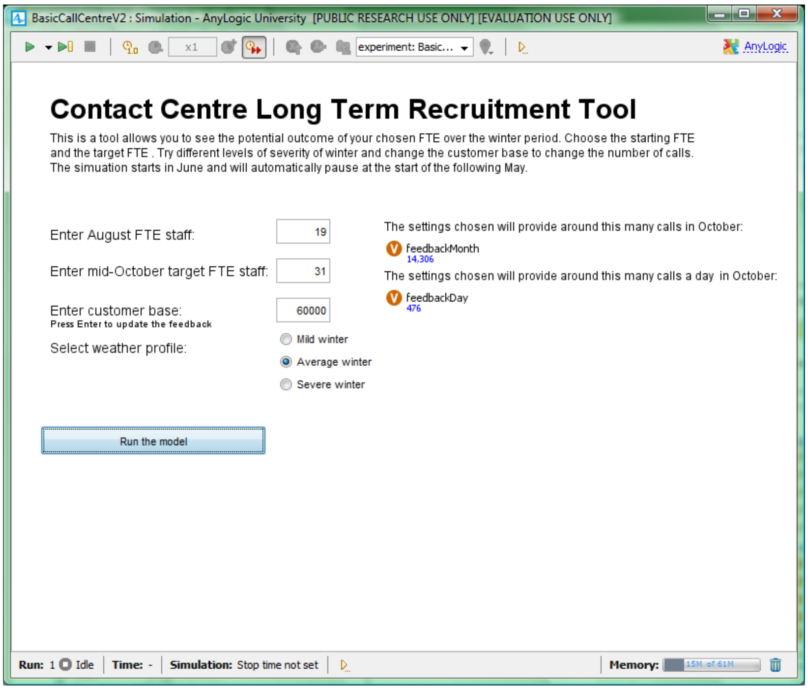
TROUBLESHOOTING

1. Needs an up-to-date version of Java
    a. follow online instructions to get most up-to-date version of Java and remove old versions
    b. restart browser
2. Security error
    a. go to the Java control panel (can be accessed through the control panel)
    b. go to the security tab
    c. change security to medium (it is recommended to set security back to high once you have finished with the simulation)
    d. restart browser



USING THE MODEL

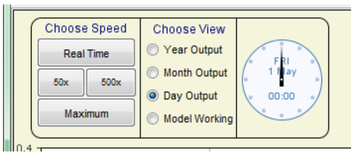
INITIALISATION

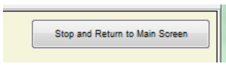


Set the scenario as desired. The feedback variables will change to give an indication on the affect of the decisions made.

NAVIGATION

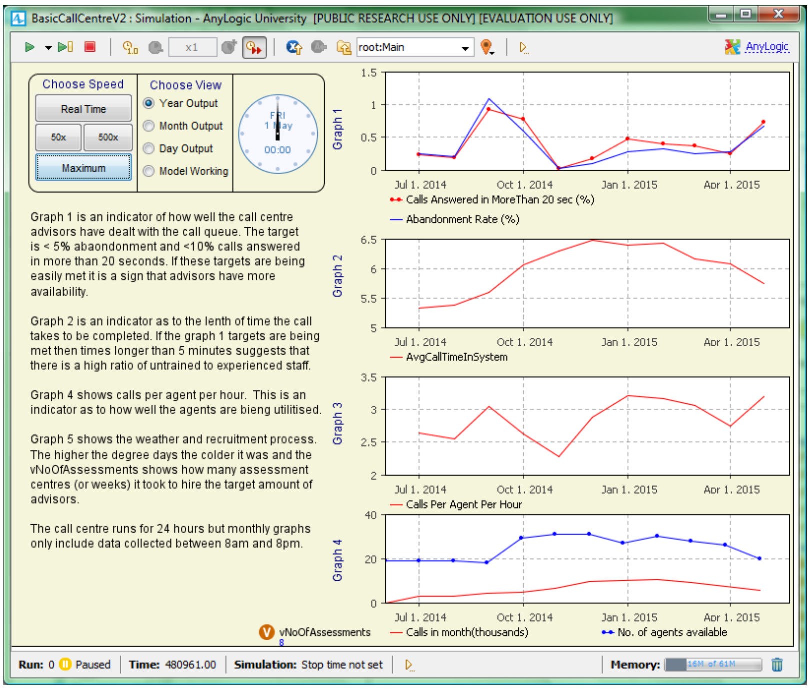
On each page there is a set of navigation pages, and on every page other than 'Year Output' there is a button to return to the front screen.



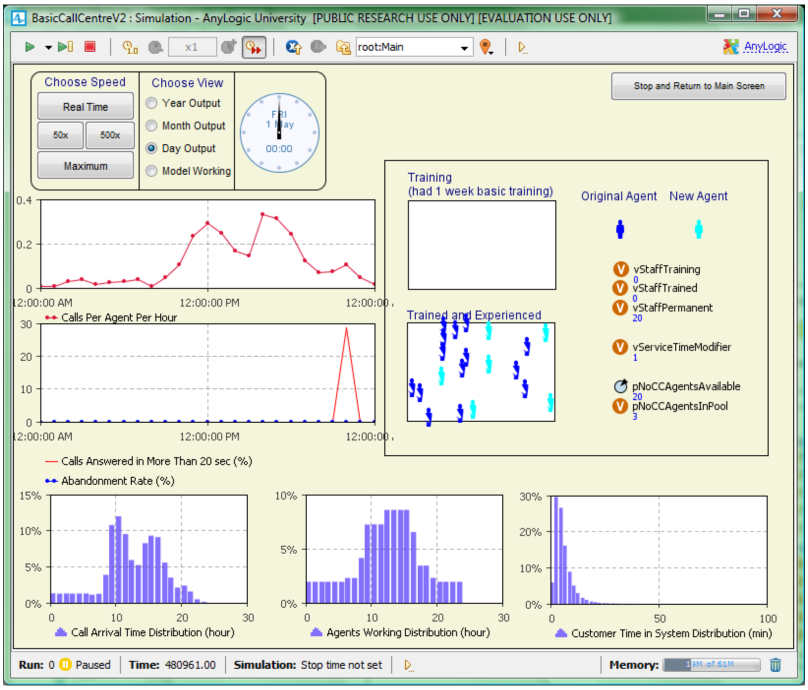


OUTPUT

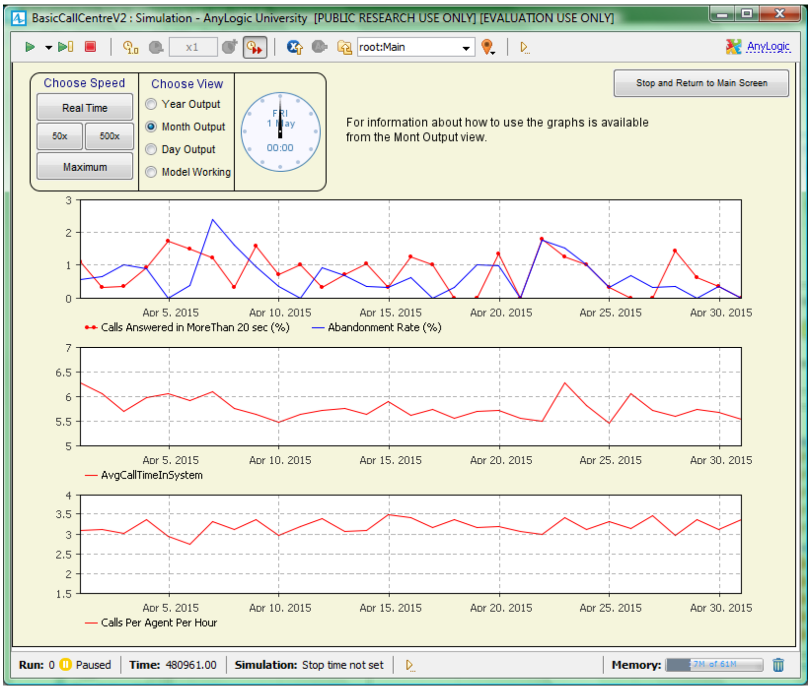
Each Output page provides information that is useful in slightly different ways. Data is only shown from between the hours of 8am and 8pm.



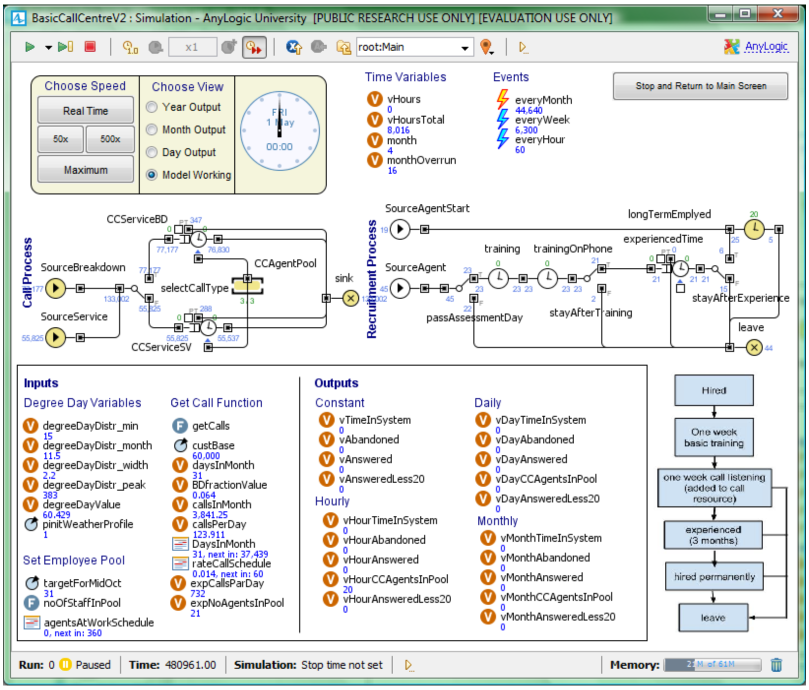
Use the Yearly Output screen for assessing the overall impact of the scenario.



The Daily screen shows daily output plus information as to how calls and advisors are distributed during the days. There is also a training area that shows advisors transitioning through the recruitment process.



The monthly screen shows the level of variation throughout the month – this is not representative of realistic monthly patterns, only to be used to get an idea of the consistency of the targets being met.



Use the Model Working screen to follow the stages of the processes.

APPENDIX 3 – CODE

This code is the result of an export from AnyLogic and has been adapted to show only relevant information.

Code

Simulation Experiment: Simulation

Name	Value
General	
Maximum Available Memory	64
Active Object Class	Main
Model time	
Execution Mode	realTimeScaled
Real Time Scale	1.0
Use Calendar	true
Stop Option	Never
Initial Time	0.0
Initial Date	Sat May 31 23:59:00 GMT 2014
Randomness	
Random Number Generation Type	randomSeed
Selection Mode For Simultaneous Events	LIFO

Name	Type	Initial value	Controlled by
initpNoOfAgentsAvailable	int	18	Editbox1 Set starting FTE
initpNoOfAgentsTarget	int	30	Edit box 2 Set Target FTE
initCustBase	int	80000	Edit box 3 Set customer base
initWeatherProfile	int	1	Radio Select weather profile
feedbackDay	double	476	Function fFeedback
feedbackMonth	double	14306	Function fFeedback

Edit Box: editbox1 Set Starting FTE

Name	Value
General	
Maximum Value	100
Minimum Value	0
Link	initpNoOfAgentsAvailable
Link To	true

Edit Box: editbox2 Set Target FTE

Name	Value
General	
Maximum Value	100
Minimum Value	0
Link	initpNoOfAgentsTarget

Edit Box: editbox3 Set Customer Base

Name	Value
General	
Maximum Value	200000
Minimum Value	0
Link	initCustBase
Link To	true
Action	
Action	fFeedback()

Radio Buttons: radio Select Weather Profile

Name	Value
General	
Orientation	VERTICAL
String Values	Mild winter, Average winter, Severe winter
Link	initWeatherProfile
Link To	true
Action	
Action	fFeedback();

Function: fFeedback

Name	Value
General	
ReturnModificator	VOID
Function body	

Body	``` int month = 9; int daysInMonth = 30; int weatherProfile = initWeatherProfile; double degreeDayDistr_min = 0; double degreeDayDistr_month = 0; double degreeDayDistr_width = 0; double degreeDayDistr_peak = 0; switch(weatherProfile){ case 0 : degreeDayDistr_min = 10 ; degreeDayDistr_month = 13; degreeDayDistr_width = 2.2; degreeDayDistr_peak = 290; break; case 1 : degreeDayDistr_min = 20 ; degreeDayDistr_month = 12; degreeDayDistr_width = 2.2; degreeDayDistr_peak = 383; break; case 2 : degreeDayDistr_min = 25 ; degreeDayDistr_month = 11; degreeDayDistr_width = 2.2; degreeDayDistr_peak = 421; break; } /*get degree day*/ double degreeDayVal = (degreeDayDistr_peak-degreeDayDistr_min) *exp(-pow(month-degreeDayDistr_month,2) /(2*pow(degreeDayDistr_width,2))) +degreeDayDistr_min; /*get breakdown call total for month*/ double bdFrac = 0.0406304264574+(0.00035647283404701*degreeDayVal); feedbackMonth = (int)((initCustBase*bdFrac)+(initCustBase*0.08)); /*get BD call total for day*/ double e = feedbackMonth/30; feedbackDay = (int)e; ```

Model: BasicCallCentreV10

Name	Value
General	
Model Time Units	Minute
Java Package Name	basiccallcentrev2
File Name	G:\Dissertation\anylogic\BasicCallCentreV10\BasicCallCentreV10.alp

Agent Type: Main

Name	Value
Agent actions	

Startup Code	SourceAgentStart.inject(pNoCCAgentsAvailable); initialise(); viewAreaGraphsYear.navigateTo();

List of Parameters:

Name	Value		
Name	Type	Initial value	Controlled by
pNoCCAgentsAvailable	int	18	Function initialise
pinitWeatherProfile	int		Function initialise
targetForMidOct	double		Function initialise
custBase	int		Function initialise

List of Variables Used in Main

Name	Type	Initial value	Controlled by
vHoursTotal	Double	18	event
vHours	double		event
noOfStaffInPool	double		Recruitment process
DaysInMonth	int	30	event
degreeDayDistr_month	double		Function getCalls
degreeDayDistr_peak	double		Function getCalls
degreeDayDistr_width	double		Function getCalls
degreeDayDistr_min	Double		Function getCalls
monthOverrun	double		event
callsInMonth	double		Function getCalls
callsPerDay	double		Function getCalls
degreeDayValue	double		Function getCalls
BDfractionValue	double		Redundant (used during testing)
vStaffTrained	double		Recruitment Process
vStaffPermanent	double		Recruitment Process
vStaffTraining	double		Recruitment Process
vServiceTimeModifier	double	1	event
pNoCCAgentsInPool	int	3	event
vNoOfAssessments	double		Recruitment Process
vHourAbandoned	double		event
vHourAnswered	double		event
vHourCCAgentsInPool	double		event
vHourAnsweredLess20	double		event
vTimeInSystem	double		Call process

vAnsweredLess20	double		Call process
vAnswered	double		Call process
vAbandoned	double		Call process
vDayTimeInSystem	double		event
vDayAnsweredLess20	double		event
vDayCCAgentsInPool	double		event
vDayAnswered	double		event
vDayAbandoned	double		event
vMonthTimeInSystem	double		event
vMonthAnsweredLess20	double		event
vMonthAbandoned	double		event
vMonthCCAgentsInPool	double		event
vMonthAnswered	double		event
chosenView	int	0	Navigation buttons

Schedule: rateCallSchedule

Name	Value
General	
Data	
Value type	double_
The schedule defines	intervals
Representation type	calendar
Repeat time	1
Repeat time interval	days
Advanced	
Unit	minute()

Schedule intervals:

Start	End	Value
00:00	01:00	0.014
01:00	02:00	0.014
02:00	03:00	0.014
03:00	04:00	0.014
04:00	05:00	0.014
05:00	06:00	0.014
06:00	07:00	0.0123
07:00	08:00	0.0137
08:00	09:00	0.0423
09:00	10:00	0.1161
10:00	11:00	0.1284
11:00	12:00	0.1025
12:00	13:00	0.0628
13:00	14:00	0.056

14:00	15:00	0.0888
15:00	16:00	0.1011
16:00	17:00	0.097
17:00	18:00	0.0615
18:00	19:00	0.0383
19:00	20:00	0.0219
20:00	21:00	0.0246
21:00	22:00	0.0178
22:00	23:00	0.0055
23:00	00:00	0.0014

Schedule: agentsAtWorkSchedule

Name	Value
Data	
Value type	double_
The schedule defines	intervals
Representation type	calendar
Repeat time	1
Repeat time interval	days

Schedule intervals:

Start	End	Value
00:00	06:00	0.0
06:00	08:00	0.22
08:00	09:00	0.39
09:00	11:30	0.67
11:30	16:00	0.78
16:00	17:00	0.61
17:00	19:00	0.33
19:00	20:00	0.22
20:00	00:00	0.0

Schedule: DaysInMonth

Name	Value
General	
Show At Runtime	true
Show name	true
Data	
Value type	integer
The schedule defines	exactTimes
Representation type	timeUnits
Repeat time	1
Time units	months
Advanced	
Use Units	false

Schedule time moments:

Time	Value

0	31
1	28
2	31
3	30
4	31
5	30
6	31
7	31
8	30
9	31
10	30
11	31

Java Class: Call

Name	Value
General	

```
public class Call extends com.xj.anylogic.libraries.enterprise.Entity {

		double dblTimeEnterCallSystem;
		double dblTimeLeaveCallSystem;
		double dblTimeEnterCallQueue;
		double dblTimeLeaveCallQueue;
		double dblTimeTillAbandon;
		boolean didTimeOut;


    public Call(){
    }

    public Call(boolean breakdown,double dblTimeEnterCallSystem,double
dblTimeLeaveCallSystem ,double dblTimeEnterCallQueue,double dblTimeLeaveCallQueue
,double dblTimeTillAbandon,boolean didTimeOut){
				this.breakdown = breakdown;
				this.dblTimeEnterCallSystem = dblTimeEnterCallSystem;
				this.dblTimeLeaveCallSystem =dblTimeLeaveCallSystem;
				this.dblTimeEnterCallQueue = dblTimeEnterCallQueue;
				this.dblTimeLeaveCallQueue = dblTimeLeaveCallQueue;
				this.dblTimeTillAbandon =dblTimeTillAbandon;
				this.didTimeOut = didTimeOut;
    }

		@Override
		public String toString() {
				return super.toString();
		}

}
```

Java Class: CCAgent

Name	Value
General	

Text	``` /** * CCAgent */ public class CCAgent extends com.xj.anylogic.libraries.enterprise.Entity implements java.io.Serializable { boolean fixedTerm; boolean experienced; double dblTimeTillLeave; double dblWorkTimeEnd; double dbleIdleTime; boolean canWork; /** * Default constructor */ public CCAgent(){ } /** * Constructor initializing the fields */ public CCAgent(boolean fixedTerm, boolean experienced, double dblTimeTillLeave, double dblWorkTimeEnd, double dbleIdleTime, boolean canWork){ this.fixedTerm = fixedTerm; this.experienced = experienced; this.dblTimeTillLeave = dblTimeTillLeave; this.dblWorkTimeEnd = dblWorkTimeEnd; this.dbleIdleTime = dbleIdleTime; this.canWork = canWork; } @Override public String toString() { return "fixedTerm = " + fixedTerm +" " + "experienced = " + experienced +" " + "dblTimeTillLeave = " + dblTimeTillLeave +" " + "dblWorkTimeEnd = " + dblWorkTimeEnd +" "+ "dbleIdleTime = " + dbleIdleTime +" " + "canWork = " + canWork +" "; } ```

Function: initialise

Name	Value
Function body	
Body	switch(pinitWeatherProfile){ case 0 : degreeDayDistr_min = 10 ; degreeDayDistr_month = 11.5; degreeDayDistr_width = 2.2; degreeDayDistr_peak = 290; break; case 1 : degreeDayDistr_min = 15 ; degreeDayDistr_month = 11.5; degreeDayDistr_width = 2.2; degreeDayDistr_peak = 383; break; case 2 : degreeDayDistr_min = 20 ; degreeDayDistr_month = 11.5; degreeDayDistr_width = 2.2; degreeDayDistr_peak = 421; break; }

Function: noOfStaffInPool

Name	Value
Function body	
Body	/*no. of agents available in day*/ pNoCCAgentsInPool = (int)(pNoCCAgentsAvailable * 0.71); /*get how many available at time of day*/ double a = agentsAtWorkSchedule.getValue(); if (a == 0) { pNoCCAgentsInPool=3; } else { a = a * pNoCCAgentsInPool; if (a<=3){ a = 3;} pNoCCAgentsInPool = (int)a;} CCAgentPool.set_capacity(pNoCCAgentsInPool);

Function: getCalls

Name	Value
Function body	

```
int a = getMonth();
month = a;
daysInMonth = DaysInMonth.getValue();

/* Create special monthOverrun
 (measured from Jan of prev yr) */
if (month <= 4) {
		monthOverrun = month + 12;
} else {
		monthOverrun = month;
}

/*get degree day*/
double degreeDayVal = (degreeDayDistr_peak-degreeDayDistr_min)
		*exp(
				-pow(monthOverrun-degreeDayDistr_month,2)
				/(2*pow(degreeDayDistr_width,2)))
		+degreeDayDistr_min;
degreeDayValue = degreeDayVal;
if (month <= 5 ) {
		degreeDayVal = degreeDayVal+ random()*20;
} if (month>5 && monthOverrun<= 10  ) {
		degreeDayVal = degreeDayVal+ random()*40-20;
} if (month == 11 ) {
		degreeDayVal = degreeDayVal+ random()*80-50;
} if (month == 12 ) {
		degreeDayVal = degreeDayVal+ random()*70-40;
} if (month == 13 ) {
		degreeDayVal = degreeDayVal+ random()*50-20;
} if (month == 14 ) {
		degreeDayVal = degreeDayVal+ random()*70-20;
} if (month == 15 ) {
		degreeDayVal = degreeDayVal+ random()*50-20;
} if (month == 16 ){
		degreeDayVal = degreeDayVal + random()*30-30;
}


/*get breakdown call total for month*/
double bdFrac = 0.0406304264574+(0.00035647283404701*degreeDayVal);
BDfractionValue = bdFrac;
callsInMonth = custBase*bdFrac;

/*get BD call total for day*/
```

Function: fSetServiceTimeModifier

Name	Value
Function body	
Body	vServiceTimeModifier = ((vStaffTraining*1.4)+ (vStaffTrained*1.2)+ (vStaffPermanent*1))/pNoCCAgentsAvailable;

Event: everyHour

Name	Value
General	
Event Recurrence Time Unit	MODEL_TIME_UNIT
Recurrence	60
Occurence Time	hour()
Occurs At Time	true
Mode	cyclic
Action	

Action	/*in use*/ vHours++; vHoursTotal++; if (vHours == 24) {vHours = 0;} noOfStaffInPool(); /*hourly*/ if (vHourAnswered != 0){ HourCallsPerAgent.updateData(); HourPerCentAnsweredLess21.updateData(); } vHourTimeInSystem = vTimeInSystem; vTimeInSystem = 0; vHourAbandoned = vAbandoned; vAbandoned = 0; vHourAnswered = vAnswered; vAnswered = 0; vHourAnsweredLess20 = vAnsweredLess20; vAnsweredLess20 = 0; vHourCCAgentsInPool = pNoCCAgentsAvailable; /*Daily and Monthly*/ if (vHours >=8 && vHours<= 20){ vDayTimeInSystem = vDayTimeInSystem + vHourTimeInSystem; vDayAbandoned = vDayAbandoned + vHourAbandoned; vDayAnswered = vDayAnswered + vHourAnswered; vDayAnsweredLess20 = vDayAnsweredLess20 + vHourAnsweredLess20; if (vHourCCAgentsInPool > vDayCCAgentsInPool) { vDayCCAgentsInPool = vHourCCAgentsInPool; } } if (vHours ==23){ DayAbandonment.updateData(); DayAvgTimeInSystem.updateData(); DayPerCentAnsweredLess22.updateData();

```
        vMonthTimeInSystem = vMonthTimeInSystem +
vDayTimeInSystem;
        vDayTimeInSystem = 0;
        vMonthAbandoned = vMonthAbandoned +
vDayAbandoned;
        vDayAbandoned = 0;
        vMonthAnswered = vMonthAnswered + vDayAnswered;
        vDayAnswered = 0;
        vMonthAnsweredLess20 = vMonthAnsweredLess20 +
vDayAnsweredLess20;
        vDayAnsweredLess20 = 0;
        vMonthCCAgentsInPool =
vMonthCCAgentsInPool+vDayCCAgentsInPool;
        vDayCCAgentsInPool = 0;
        /*if (vDayCCAgentsInPool > vMonthCCAgentsInPool) {
                vMonthCCAgentsInPool =
vDayCCAgentsInPool;
                vDayCCAgentsInPool = 0;
        }*/
}




/*extra*/


for (int i=0; i<pNoCCAgentsInPool;i++){
dataRS1.add( getHourOfDay() );}
```

Event: everyMonth

Name	Value
General	
Event Recurrence Time Unit	MONTH
Recurrence	1
Event Occurrence Date	Sun Jun 01 00:00:00 GMT 2014
Occurs At Time	false
Mode	cyclic
Trigger Type	timeout
Show At Runtime	true
Show name	true
Action	
Action	getCalls(); if (vMonthAbandoned != 0) { YearAvgTimeInSystem.updateData(); YearAbandonment.updateData(); YearPerCentAnsweredLess20.updateData(); } vMonthTimeInSystem = 0; vMonthAbandoned = 0; vMonthAnswered = 0; vMonthAnsweredLess20 =0; vMonthCCAgentsInPool = 0; if (monthOverrun==16) { pauseSimulation(); }

Event: everyWeek

Name	Value
General	
Event Recurrence Time Unit	WEEK
Recurrence	1
Event Occurrence Date	Tue Jun 03 09:00:00 GMT 2014
Occurs At Time	false
Mode	cyclic
Trigger Type	timeout
Show At Runtime	true
Show name	true
Action	
Action	if (pNoCCAgentsAvailable < targetForMidOct && monthOverrun >=8 && monthOverrun <=10) { int agentsToEmploy = uniform_discr(2, 10); SourceAgent.inject(agentsToEmploy); vNoOfAssessments++;} fSetServiceTimeModifier();

Source: SourceBreakdown

Name	Value
General	
Generic Parameters Substitute	Call
Advanced	
Replication	false
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Arrivals defined by	com.xj.anylogic.libraries.enterprise.Source. RATE_SCHEDULE
Arrival rate	(callsPerDay*rateCallSchedule.getValue())/1*minute()
Interarrival time	exponential(1)
Rate schedule	rateCallSchedule
Modify rate	true
Rate expression	(baseRate * callsPerDay)/60
Entities per arrival	1
Limited number of arrivals	false
Maximum number of arrivals	Integer.MAX_VALUE
New entity	new Call()
On exit	entity.breakdown = true; entity.dblTimeEnterCallSystem=time(); entity.dblTimeTillAbandon = lognormal(4.56343, 1.44683, 0)/60;/*exponential(0.5*minute(), 0);*/ entity.didTimeOut = false; dataRS.add(getHourOfDay());
Unique shape for each entity	false
Enable rotation	false

Enable vertical rotation	true

Service: CCServiceBD

Name	Value
General	
Generic Parameters Substitute	Call
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Resource quantity	1
Delay time	(lognormal(5.79643, 0.614067, 0)/60)*vServiceTimeModifier
ResourcePool object	null
On enter	entity.dblTimeEnterCallQueue = time();
On enter delay	entity.dblTimeLeaveCallQueue = time();
Queue capacity	100
Maximum queue capacity	true
Enable exit on timeout	true
Timeout	entity.dblTimeTillAbandon
On exit (timeout)	entity.didTimeOut = true;

Service: CCServiceSV

Name	Value
General	
Generic Parameters Substitute	Call
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Resource quantity	1
Delay time	(lognormal(5.47917, 0.506602, 0)/60)*vServiceTimeModifier
ResourcePool object	null
On enter	entity.dblTimeEnterCallQueue = time();
On enter delay	entity.dblTimeLeaveCallQueue = time();
Queue capacity	100
Maximum queue capacity	false
Enable exit on timeout	true
Timeout	entity.dblTimeTillAbandon
On exit (timeout)	entity.didTimeOut = true;

SelectOutput: selectCallType

Name	Value
General	

Generic Parameters Substitute	Call
Advanced	
Replication	false
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Select True output	false
Condition	entity.breakdown
Probability	1

ResourcePool: CCAgentPool

Name	Value
General	
Generic Parameters Substitute	ResourceUnit
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Resource units are	true
Capacity defined	com.xj.anylogic.libraries.enterprise.ResourcePool.CAPACITY_DIRECT
Capacity	pNoCCAgentsInPool-3
Capacity when "On"	1
New resource unit	new ResourceUnit()

Sink: sink

Name	Value
General	
Generic Parameters Substitute	Call
Advanced	
Replication	false
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value

On enter	/*Update entity variable*/ entity.dblTimeLeaveCallSystem =time(); /*Update simulation variable*/ if (entity.didTimeOut!=true) { vAnswered++; vTimeInSystem = vTimeInSystem + (entity.dblTimeLeaveCallSystem-entity.dblTimeEnterCallSystem); if((entity.dblTimeLeaveCallQueue-entity.dblTimeEnterCallQueue)<0.33){ vAnsweredLess20++; } } else { vAbandoned++; }
	/*other*/ /*run functions*/ //update general usage stats histWait1.add(entity.dblTimeLeaveCallSystem-entity.dblTimeEnterCallSystem);

Source: SourceAgent

Name	Value
General	
Generic Parameters Substitute	CCAgent
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Arrivals defined by	com.xj.anylogic.libraries.enterprise.Source.MANUAL
Arrival rate	week()
Interarrival time	week()*1
Modify rate	false
Rate expression	baseRate
Entities per arrival	(int)random()*8+2
New entity	new CCAgent()
On exit	entity.fixedTerm=true; entity.experienced = false; entity.dblTimeTillLeave = triangular(2*week(),6*month(),7*month()); /*liklihood to leave etc set here*/
Entity animation shape	ccAgentShapeTrain

Source: SourceService

Name	Value
General	
Generic Parameters Substitute	Call
Replication	false
Show name	true

Advanced	
Replication	false
Java Package Name	com.xj.anylogic.libraries.enterprise
Show At Runtime	true
Public	false

Agent Parameters:

Name	Value
Arrivals defined by	com.xj.anylogic.libraries.enterprise.Source.RATE_SCHEDULE
Arrival rate	(callsPerDay*rateCallSchedule.getValue())/1*minute()
Interarrival time	100
Rate schedule	rateCallSchedule
Modify rate	true
Rate expression	(baseRate * (0.08* custBase/daysInMonth))/60
Entities per arrival	1
Limited number of arrivals	false
Maximum number of arrivals	Integer.MAX_VALUE
New entity	new Call()
On exit	entity.breakdown=false; entity.dblTimeEnterCallSystem=time(); entity.dblTimeTillAbandon = lognormal(4.11051, 1.36609, 0)/60;/*exponential(0.5*minute(), 0);*/ entity.didTimeOut = false; dataRS.add(getHourOfDay());
Unique shape for each entity	false
Enable rotation	false
Enable vertical rotation	true

Source: SourceAgentStart

Name	Value
General	
Generic Parameters Substitute	CCAgent
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Arrivals defined by	com.xj.anylogic.libraries.enterprise.Source.MANUAL
Arrival rate	week()
Interarrival time	week()*1
Modify rate	false
Rate expression	baseRate
Entities per arrival	10
Limited number of arrivals	false
Maximum number of arrivals	Integer.MAX_VALUE
New entity	new CCAgent()

On exit	entity.fixedTerm=false; entity.experienced = true; entity.dblTimeTillLeave = triangular(4*week(),1*year(),3*year()); noOfStaffInPool(); vStaffPermanent++;
Entity animation shape	ccAgentShapeExp

SelectOutput: stayAfterTraining

Name	Value
General	
Generic Parameters Substitute	Entity
Replication	false
Show name	true
Advanced	
Replication	false
Java Package Name	com.xj.anylogic.libraries.enterprise
Show At Runtime	true
Public	false

Agent Parameters:

Name	Value
Select True output	true
Condition	randomTrue(0.5)
Probability	0.95
On exit (true)	vStaffTrained++; vStaffTraining= vStaffTraining-1;
On exit (false)	vStaffTraining =vStaffTraining-1; pNoCCAgentsAvailable=pNoCCAgentsAvailable-1

Service: experiencedTime

Name	Value
General	
Generic Parameters Substitute	CCAgent
Advanced	
Replication	false
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Resource quantity	0
Delay time	week()*12
ResourcePool object	null
Queue capacity	100
Maximum queue capacity	false
Enable exit on timeout	true
Timeout	entity.dblTimeTillLeave
On exit (timeout)	vStaffTrained =vStaffTrained-1; pNoCCAgentsAvailable=pNoCCAgentsAvailable-1
Enable preemption	false

Entity priority	0
Animation guide shape (queue)	ccAgentStorageExperienced
Animation type (queue)	com.xj.anylogic.libraries.enterprise.Animator.BAG
Animation direction (queue)	true
Animation guide shape (delay)	ccAgentStorageExperienced
Animation type (delay)	com.xj.anylogic.libraries.enterprise.Animator.BAG
Animation direction (delay)	true
Enable statistics	false

SelectOutput: stayAfterExperience

Name	Value
General	
Generic Parameters Substitute	CCAgent
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Select True output	true
Condition	randomTrue(0.5)
Probability	0.25
On exit (true)	entity.experienced = true;
	entity.fixedTerm = false; vStaffPermanent++; vStaffTrained= vStaffTrained-1; entity.dblTimeTillLeave = triangular(4*week(),1*year(),3*year());
On exit (false)	pNoCCAgentsAvailable= pNoCCAgentsAvailable-1; vStaffTrained= vStaffTrained-1;

Delay: longTermEmplyed

Name	Value
General	
Generic Parameters Substitute	CCAgent
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Delay time	entity.dblTimeTillLeave
Speed	10
Capacity	1
Maximum capacity	true
On exit	pNoCCAgentsAvailable= pNoCCAgentsAvailable-1; vStaffPermanent= vStaffPermanent-1;
Animation guide shape	ccAgentStorageExperienced
Animation type	com.xj.anylogic.libraries.enterprise.Animator.BAG

SelectOutput: passAssessmentDay

Name	Value
General	
Generic Parameters Substitute	Entity

Agent Parameters:

Name	Value
Select True output	true
Condition	randomTrue(0.5)
Probability	0.7

Delay: training

Name	Value
General	
Generic Parameters Substitute	CCAgent
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Agent Parameters:

Name	Value
Delay time is	false
Delay time	1*week()
Speed	10
Capacity	1
Maximum capacity	true
Animation guide shape	ccAgentStorageTraining
Animation type	com.xj.anylogic.libraries.enterprise.Animator.BAG
Animation direction	true
Enable statistics	false

Delay: trainingOnPhone

Name	Value
General	
Generic Parameters Substitute	CCAgent
Replication	false
Show name	true
Advanced	
Replication	false
Java Package Name	com.xj.anylogic.libraries.enterprise
Show At Runtime	true
Public	false

Agent Parameters:

Name	Value
Delay time is	false
Delay time	1*week()
Speed	10
Capacity	1
Maximum capacity	true

On enter	vStaffTraining++; pNoCCAgentsAvailable++;
Animation guide shape	ccAgentStorageTraining
Animation type	com.xj.anylogic.libraries.enterprise.Animator.BAG
Animation direction	true
Enable statistics	false

Sink: leave

Name	Value
General	
Generic Parameters Substitute	CCAgent
Advanced	
Java Package Name	com.xj.anylogic.libraries.enterprise

Time Plot: YearAvgTimeInSystem

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	false
Dataset Samples To Keep	200
Scale	
Time Window	356*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
AvgCallTimeInSystem	vMonthTimeInSystem/vMonthAnswered	NONE	red	true	1	LINEAR

Time Plot: YearPerCentAnsweredLess20

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	false
Dataset Samples To Keep	200
Scale	
Time Window	356*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Calls Answered in MoreThan 20 sec (%)	(1-(vMonthAnsweredLess20)/vMonthAnswered)*100	CIRCLE	red	true	1	LINEAR

Abandonment Rate (%)	(vMonthAbandoned/(vMonthAnswered+vMonthAbandoned))*100	NONE	blue	true	1	LINEAR

Time Plot: YearAbandonment

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	false
Dataset Samples To Keep	200
Scale	
Time Window	356*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Calls Per Agent Per Hour	(vMonthAnswered/(vMonthCCAgentsInPool*0.71*6.66))callsInMonth/1000	NONE	red	true	1	LINEAR

Time Plot: Utilisation1

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	true
Recurrence	day()*30
Dataset Samples To Keep	200
Scale	
Time Window	356*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Calls in month (thousands)	callsInMonth/1000	NONE	red	true	1	LINEAR
No. of agents available	pNoCCAgentsAvailable	CIRCLE	blue	true	1	LINEAR

Time Plot: HourPerCentAnsweredLess21

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	false
Dataset Samples To Keep	200

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Calls Answered in More Than 20 sec (%)	(1-(vHourAnsweredLess20)/vHourAnswered)*100	NONE	red	true	1	LINEAR
Abandonment Rate (%)	(vHourAbandoned/(vHourAnswered+vHourAbandoned))*100	CIRCLE	blue	true	1	LINEAR

Time Plot: HourCallsPerAgent

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	false
Dataset Samples To Keep	200
Scale	
Time Window	1*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Calls Per Agent per hour	(vHourAnswered/(vHourCCAgentsInPool*6.66))	CIRCLE	crimson	true	1	LINEAR

Time Plot: DayAvgTimeInSystem

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	false
Dataset Samples To Keep	200
Scale	
Time Window	30*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
AvgCallTimeInSystem	vDayTimeInSystem/vDayAnswered	NONE	red	true	1	LINEAR

Time Plot: DayPerCentAnsweredLess22

Name	Value
General	
Public	true
Data update	
Analysis Auto Update	false

Dataset Samples To Keep	200
Scale	
Time Window	30*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Calls Answered in MoreThan 20 sec (%)	(1-(vDayAnsweredLess20)/vDayAnswered)*100	CIRCLE	red	true	1	LINEAR
Abandonment Rate (%)	(vDayAbandoned/(vDayAnswered+vDayAbandoned))*100	NONE	blue	true	1	LINEAR

Time Plot: DayAbandonment

Name	Value
Data update	
Analysis Auto Update	false
Dataset Samples To Keep	200
Scale	
Time Window	30*day()
Time Window Units	MODEL_TIME_UNIT
Vertical Scale	AUTO

Plot Items:

Title	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Calls Per Agent Per Hour	(vDayAnswered/(vDayCCAgentsInPool*0.71*6.66))	NONE	red	true	1	LINEAR

Histogram: chart Call Arrival Distribution Through Day

Name	Value
Data update	
Analysis Auto Update	false
Appearance	
Bars Relative Width	0.8
Labels Vertical Position	DEFAULT

Histogram Data Items:

Title	Dataset	PDF Color	CDF Color	Mean Color	Width	Low % Color	High % Color
Call Arrival Time Distribution (hour)	dataSet	lightSlateBlue	darkKhaki	deepSkyBlue	1	coral	gold

Histogram: chart1 Agent Distribution Through Day

Name	Value
Data update	
Analysis Auto Update	false

Appearance	
Bars Relative Width	0.8

Histogram Data Items:

Title	Dataset	PDF Color	CDF Color	Mean Color	Width	Low % Color	High % Color
Agents Working Distribution (hour)	datSet1	lightSlateBlue	darkKhaki	deepSkyBlue	1	coral	gold

Histogram: chart3 Time In System

Name	Value
Data update	
Analysis Auto Update	false

Histogram Data Items:

Title	Dataset	PDF Color	CDF Color	Mean Color	Width	Low % Color	High % Color
Customer Time in System Distribution (min)	histWait1	lightSlateBlue	darkKhaki	deepSkyBlue	1	coral	gold

Histogram Data: dataSet

Name	Value
General	
Calculate CDF	false
Number Of Intervals	24
Show At Runtime	false
Show name	true
Values range	
Data range	false
Range Minimum	0
Range Maximum	24
Data update	
Analysis Auto Update	false

Histogram Data: dataSet1

Name	Value
General	
Calculate CDF	false
Number Of Intervals	24
Show At Runtime	false
Show name	true
Values range	
Data range	false
Range Minimum	0
Range Maximum	24
Data update	
Analysis Auto Update	false

Histogram Data: histWait1

Name	Value
General	
Calculate CDF	false
Number Of Intervals	50
Show At Runtime	true
Show name	true
Values range	
Data range	false
Range Minimum	0
Range Maximum	100
Data update	
Analysis Auto Update	true
Recurrence	1

Function: fChooseView

Name	Value
Function body	
Body	switch(chosenView){ case 0 : viewAreaGraphsYear.navigateTo(); break; case 1 : viewAreaGraphsMonth.navigateTo(); break; case 2 viewAreaGraphsDay.navigateTo(); break; case 3 : Origin.navigateTo(); break; } chooseView.setValue(chosenView); chooseView1.setValue(chosenView); chooseView2.setValue(chosenView); chooseView3.setValue(chosenView);

Radio Buttons: chooseView1

Name	Value
General	
Orientation	VERTICAL
String Values	Year Output, Month Output, Day Output, Model Working
Default Value	0
Link To	false
Public	true
Action	
Action	chosenView = chooseView1.getValue(); fChooseView();

Radio Buttons: chooseView2

Name	Value
General	
Orientation	VERTICAL
String Values	Year Output, Month Output, Day Output, Model Working
Default Value	3
Link To	false
Public	true
Action	
Action	chosenView = chooseView2.getValue(); fChooseView();

Radio Buttons: chooseView3

Name	Value
General	
Orientation	VERTICAL
String Values	Year Output, Month Output, Day Output, Model Working
Default Value	1
Link To	false
Public	true
Action	
Action	chosenView = chooseView3.getValue(); fChooseView();

Button: Set Time to Real Time

Name	Value
General	
Label Text	Real Time
Public	true
Action	
Action	getEngine().setRealTimeMode(true); getEngine().setRealTimeScale(1);

Button: Set time to 500 x speed

Name	Value
General	
Label Text	500x
Action	
Action	getEngine().setRealTimeMode(true); getEngine().setRealTimeScale(500);

Button: Set time to 50 x speed

Name	Value

General	
Label Text	50x
Public	true
Action	
Action	getEngine().setRealTimeMode(true); getEngine().setRealTimeScale(50);

Button: Set speed maximum

Name	Value
General	
Label Text	Maximum
Public	true
Action	
Action	getEngine().setRealTimeMode(false);

Button: Stop and Return to main screen

Name	Value
General	
Label Text	Stop and Return to Main Screen
Public	true
Action	
Action	getEngine().stop()