**IT Service Desk Simulation Model**

**Valparaiso University**

**Pavan Kumar Battula**

**Introduction:**

The IT Service Desk is a simulation model of agent and customer interaction for resolving Information Technology related problems. This model is designed for the purpose of predicting the future service requests for a certain number of years by using historic data of it and optimizing the resolution time to resolve each ‘problem’ or ‘request’.

This system is an agent-based modeling. It contains the entities of representatives, customers, and patch (like an office’s service desk). It has global variables and can be accessed anywhere in the system. The breeds of this system are agents and users. Each breed and other entities have their own variables that can be accessed with the breed or entity’s plural input-value anywhere in the system. It has two buttons that one is to set up the system like initializing variables and creating turtles and the other is to run the system to execute the functions. The objective of the system is achieved by performing various functions in a certain order of execution. The results of required metrics are plotted in the form of graph, so it is easy to understand and observe the required outputs and unexpected outcomes. The results are stored into an excel sheet file and collected data can be used for predicting future service request dynamics and various other dynamics of system for optimization of it. This system can be further expanded to automated operations like chat bots for optimization.

Finally, this system is evaluated with various input variables in different scenarios or methods to observe the dynamics of different future service request volumes, average resolution times, and total resolution times. This system can be interesting when working with different variables values on the interface like playing a video game.

**Existing Models and Research**

Bober, P. (2014). Simulation for IT Service Desk Improvement. Quality, Innovation, Prosperity, 18, 47-58.

Bartsch, C., Mevius, M., & Oberweis, A. (2010). Simulation Environment for IT Service Support Processes: Supporting Service Providers in Estimating Service Levels for Incident Management. 2010 Second International Conference on Information, Process, and Knowledge Management, 23-31.

Bosu, M.F., Abuaiadah, D., Khanna, P., Nepia, S., & Palmer, D. (2019). Evaluation of IT Service Desk: A Case Study.

Orta, E., Ruiz, M., Hurtado, N., & Gawn, D. (2014). Decision-making in IT service management: a simulation based approach. Decis. Support Syst., 66, 36-51.

From the Bober, P. (2014). Simulation for IT Service Desk Improvement, learned insights of such things as raising requests, prioritizing them, discrete execution of procedures, and optimization techniques. Incident response techniques are well illustrated in Bartsch, C., Mevius, M., & Oberweis, A. (2010). and it helped a great deal in the development of request-processing procedure in my model. Evaluation techniques are well explained and reflected in the Bosu, M.F., Abuaiadah, D., Khanna, P., Nepia, S., & Palmer, D. (2019). source. The decision-making techniques are taken with reference from the Orta, E., Ruiz, M., Hurtado, N., & Gawn, D. (2014).

**The Model**

Generate-issue

Agent-behavior

Breeds

GO

Request-assistance



User-behavior

Move-users

Global Variables

Assign-tasks

Record historical data

Agents-own variables



Process-tasks

Predict future-request-volume

Visualize and report



Customer-own variables

Calculate slope

Logistic function

update-performance-metrics



setup

**Diagram 1: structure and flow order of execution of model**

The model is structured in flexible and required order of execution to achieve the targeted outcomes of the model. The left side of the model diagram 1 is setup and different variables of the model, the middle part of the diagram are the main functions or procedures of it, and the right side of the diagram are the sub-procedures and their sub-procedures. The arrow lines indicate how the order of execution is performed in the model. Various mathematical operations are performed such as probability, mean and simple arithmetic operations in multiple procedures to manipulate the resolution time, total resolution time, and future-request-volume. This structure is helpful to create further sub-procedures to optimize with little modifications to the main procedures.

**Analysis:**

To implement the structure where I have followed the criteria of ODD (overview, design, details) for my analysis. I have explained about each section in the below.

**Overview section:**

1.Purpose and pattern: the purpose of this model is to implement a simulation model for optimizing the system performance and predict the count of future service requests. The patterns are like assigning tasks randomly, finding the key performance metrics such as average resolution time and total resolution time. The execution of procedures such as moving customers and assigning tasks, processing tasks, and updating performance metrics and storing them into a file for sharing and tracking data for future predictions or usages.

2. identifying entities, state variables and scales: breeds and patches, global variables, and other local variables. The scales of the breeds.

3.process overview and scheduling: Models execution starts with moving users, then as follows to generate -issues, assign tasks, process tasks, update-performance metrics. The scheduling is done according to the required order of execution.

**Design Concepts:**

The basic principles of this model are adhered to establish relationship between the agent and customers for resolving the issues, randomness in assigning tasks and resolution times for issues, introducing stochasticity elements to simulate real world-variability. Additionally, integrate a feedback loop by continuously updating and visualizing the performance metrics.

This model exhibits emergent properties through the interplay of agents and customers such as patterns like varying agent workload and average resolution times. It emerges from decentralized decision making because of the stochastic interactions between agents and customers. This model has sensing manifestations such as task assignment, agent availability, issue status.

Prediction is implemented for this model as it must answer one of the objectives of its. As predictions can be calculated with probabilities, I used randomness in the measurement of metrics. One of the big questions that the count of future service requests is answered through by estimating various metrics like average resolution time and total resolution times and current service requests. Stochasticity is used in this model to generate randomness in the assignment of tasks and average resolution times. This is essential for real-world variability.

Interaction in the model through several layers, first, user randomly generates issues, without a fixed pattern. And agents then interact with these issues on the criteria of user wait times, dynamic allocation of tasks. The interaction intensifies as the agent resolves the issues, affects the global variables like average resolution time and total resolution times. Collectives are represented by the groups of breeds like agents and users. The behaviors and interactions within each collectively impact the overall system dynamics.

**Details:**

Initialization is the process of setting default values to variables such as global variables to zero and other variables, creating breeds and setting up the whole model environment on the patch. Agents and customers are positioned on the patch, set to specific attributes of their own.

Input Variables are set through the help of the sliders which are flexible for the interface user to change them and test the system. Number of agents, number of customers, prediction years, mean and standard deviation value.

The sub-models are part of the main procedures, and they are given on the right of in fig1. They perform their individual tasks. Modifications of the sub-procedures can be done without disturbance to the main procedures. I created various sub models which are independent and executed in the function calling order to the results successfully. The model has sub models such as user behavior, agent behavior, issue-generate, request- assistance, logistic function, calculate slope, and predict future requests.

**Methods**

The methods that I used in this model to manipulate the results of this model, to observe the results clearly and accurately,

**Method of optimization by changing variable values**: The variables like number of agents can varied to optimize the average resolution time and total resolution times. I would like to show the results with various cases or scenarios of optimization method.

**Case1:** Input values: Number of agents = 10, number of customers = 201

I am attaching figure1 of interface of model and its execution by setting the above input values as a part of illustrating the optimization of model in the next page.

This is what the interface of model looks like. The input sliders, output monitors, patch, setup and go buttons, and plots of different results.

A screenshot of a computer

Description automatically generated

**Fig 1: Interface of model with execution of case1 input values**

In fig1. You can see the slider of number of agents is set to 10 and number of customers is set to 201. When we execute the model with those values, the average resolution time (ART) is 5.348 and the total resolution time (TRT) is 1075 per 201 issues.

A graph showing a graph

Description automatically generated with medium confidence A graph of a graph

Description automatically generated with medium confidence

**Fig2: Average Resolution Time** **Fig3: Total Resolution Time and Future-req-vol** the above figures are graphical representation of average resolution time and total resolution time and future-requests volume.

**Case2:** By increasing the number of agents while the number of customers is constant, we can optimize the average resolution time and total resolution time that means the time taken to resolve a problem can be significantly decreased. It is clearly illustrated in the figure4 below.

A screenshot of a computer

Description automatically generated

**Fig4: Increment in number of agents while number of customers constant**

You can see that the value of number of agents is set to 40 while number of customers is 201 (constant same as in fig1). The results, ART is **1.606**, TRT is **323**, and future request volume (FRV) is increased to 58 (fig1) to 295 (fig4) for the span of same 5 prediction years (constant), it means that more problems can be solved within the given time of 1075 of TRT (fig1).

A screen shot of a white screen

Description automatically generated A graph on a white board

Description automatically generated

**Fig5: Decrease in Avg Resolution Time** **Fig6: Decrease in TRT and Increase in FRV**

The more agents can resolve an issue within less time than it takes for less agents. This is one of the ways to optimize the IT service desk’s performance. There are several other ways to optimize its performance while maintaining the cost is low. There are methods like automizing the system with an AI chat bot and self-serving of customers websites or apps to optimize performance.

**2.What if analysis method**:The processing time for each problem could change depending on the complexity of it. So, I used random normalization of resolution time for each problem with the inputs of mean and standard deviation. The default values are set as mean 60 and standard deviation 10 to the random normal function. We can set any values within their ranges, so the mean range of values are 10 to 100 and standard deviation range 1 to 50. We can use this method to see variations in the results of all metrics for each run or rerun.

**Results**

The anticipated results of this model are successfully achieved with the model I designed and implemented. The big questions such as prediction of the count of future service requests and optimization of service desk performance are accurately answered.

A screenshot of a computer

Description automatically generated

**Fig:7 Result of optimized Average Resolution Time**

The results here are the total resolution time response and future service request count response.

A graph of a graph

Description automatically generated

**Fig8: Count of Future Requests Count and Result of Optimized Total Resolution Time**

**Discussion**

The big questions are the prediction of number of future service requests and optimization of the performance of system.

The figure8 shows the number of future service requests on the plot which is **295** for a span of **5-year** prediction. While the number of agents is 40 and number of customers is 201, mean is 60 and standard deviation is 10. The total resolution time is **323** and average resolution time is **1.606.**

This model is designed to predict the number of future service requests for the range of 1 to 25 years.The model is run with both 10 and 20 prediction years to find their respective number of future service requests for this discussion. I attach the plots and images below, please check them out. You can experiment with any values of your choice to predict the count.

A graph on a white board

Description automatically generated

**Fig9: Prediction of Future Service Requests for 10 years**

The above figure 9 shows the number of future service requests on the plot which is **590** for a span of **10-yea**r prediction. While the number of agents is 40 and number of customers is 201, mean is 60 and standard deviation is 10. The total resolution time is **335** and average resolution time is **1.666.**

Notice how the average resolution and total resolution time for both 5 and 10 years is almost same but the requests of 10 years are increased as expected. This shows that the total time taken for resolving the requests of 10 year will be almost equal to the time taken for resolving the requests of 5-year. This is a clear illustration of optimization of the system.

If I run for the 20 years with the same variables it is as shown in below figure 10,

A graph with a red line

Description automatically generated

**Fig10: Prediction of Future Service Requests for 20 years**

The above figure 10 shows the number of future service requests on the plot which is **1184** for a span of **20-yea**r prediction. While the number of agents is 40 and number of customers is 201, mean is 60 and standard deviation is 10. The total resolution time is **321** and average resolution time is **1.59.** This discussion clearly explained the answers to the big questions.

**Limitations:**

Hiring more agents for optimization mayn’t be budget friendly.

Automation such as AI chat bots or self-servicing portals are not implemented.

**Extensions:**

This model can be extended for automation like AI bots.

Self-servicing portals for customers with interface login credentials can be implemented.

Economics and cost related sub procedures and plots can be created in this model.

More attributes for breeds and patches can be extended based on the choice of the developer.

**Conclusion:** This model is developed to answer the following questions,

What are the service requests count for 5, 10 and 20 years? How to optimize IT service desk Performance? This model clearly answered to the questions with real world environments.

References

https://blogs.valpo.edu/sstreuber/

Bober, P. (2014). Simulation for IT Service Desk Improvement. Quality, Innovation, Prosperity, 18, 47-58.

Bartsch, C., Mevius, M., & Oberweis, A. (2010). Simulation Environment for IT Service Support Processes: Supporting Service Providers in Estimating Service Levels for Incident Management. 2010 Second International Conference on Information, Process, and Knowledge Management, 23-31.

Bosu, M.F., Abuaiadah, D., Khanna, P., Nepia, S., & Palmer, D. (2019). Evaluation of IT Service Desk: A Case Study.

Orta, E., Ruiz, M., Hurtado, N., & Gawn, D. (2014). Decision-making in IT service management: a simulation based approach. Decis. Support Syst., 66, 36-51.

Siti-Nabiha, A., Thum, W. Y., & Sardana, G. D. (2012). A case study of service desk's performance measurement system.*International Journal of Commerce and Management, 22*(2), 103-118. <https://doi.org/10.1108/10569211211239412>.

Zgodavova, K., Kisela, M., & Sutoova, A. (2016). Intelligent approaches for an organisation's management system change.*TQM Journal, 28*(5), 760-773. <https://ezproxy.valpo.edu/login?url=https://www.proquest.com/scholarly-journals/intelligent-approaches-organisations-management/docview/1835572584/se-2>.

Malley, M. G. (2017). *Proactive IT Incident Prevention: Using Data Analytics to Reduce Service Interruptions*(Order No. 10257804). Available from ProQuest One Business Dissertations. (1873892659). <https://ezproxy.valpo.edu/login?url=https://www.proquest.com/dissertations-theses/proactive-incident-prevention-using-data/docview/1873892659/se-2>.

Carrie, L. M., Grooms, J. C., & King, H. (2018). To Infinity and Beyond—Gamifying IT Service‐Desk Training: A Case Study.*Performance Improvement Quarterly, 31*(3), 249-268. <https://doi.org/10.1002/piq.21263>.

Special Thanks! To Professor Sonja Streuber



**About Author:**

**Pavan Kumar Battula.** is a graduate student of MSc in Analytics and Modeling at Valparaiso University, Valparaiso, Indiana, 46383**. Email:** [pavankumar.battula@valpo.edu](mailto:pavankumar.battula@valpo.edu).

# References

Battula, P. K. (2023, December). IT Service Desk Simulation Model. *Simulation Modeling, IT service Desk model, Valparaiso university, 8*, 1-8. Retrieved from https://docs.google.com/document/d/1r3XIAeoDQF91xVVK6WFEvukD13iagL7c/edit?usp=sharing&ouid=110503208051189214832&rtpof=true&sd=true