Table of Contents

[Exploring and implementing SimPy 2](#_Toc101080968)

[Project group # 223 - Members 2](#_Toc101080969)

[1. Abstract 3](#_Toc101080970)

[2. Background and use case 3](#_Toc101080971)

[Description of use case (Abused women’s shelter) 3](#_Toc101080972)

[Case types and processing of abuse victims 4](#_Toc101080973)

[3. Main findings 5](#_Toc101080974)

[SimPy building blocks – how it works 5](#_Toc101080975)

[Our code architecture 5](#_Toc101080976)

[User Guide 5](#_Toc101080977)

[Pros / Cons / Insights 7](#_Toc101080978)

[Comparison with Arena 7](#_Toc101080979)

[4. Output data from use case 7](#_Toc101080980)

[(a) Occupancy 7](#_Toc101080981)

[(b) Un-utilized capacity 8](#_Toc101080982)

[(c) Abuse victims turned away over time 9](#_Toc101080983)

[(d) Abuse victims sent to hotel over time 10](#_Toc101080984)

[Conclusions 11](#_Toc101080985)

[References 12](#_Toc101080986)

# Exploring and implementing SimPy

# Project group # 223 - Members

* Omer Ansari
* Harshini Prabakaran
* Sreeram Jayabalan

# Abstract

1. SimPy is a simulation package based on the popular Python language.
2. Our team is attempting to understand SimPy, it’s strengths, weaknesses and apply it in a real work simulation use case.
3. Our intent is to share our learnings of this package through opinions, recommendations, a user guide and impressions on any weaknesses.
4. Our expectation is that, In addition to being a project deliverable, this structured approach to learning will help reinforce our understanding of SimPy so we may use it more effectively.

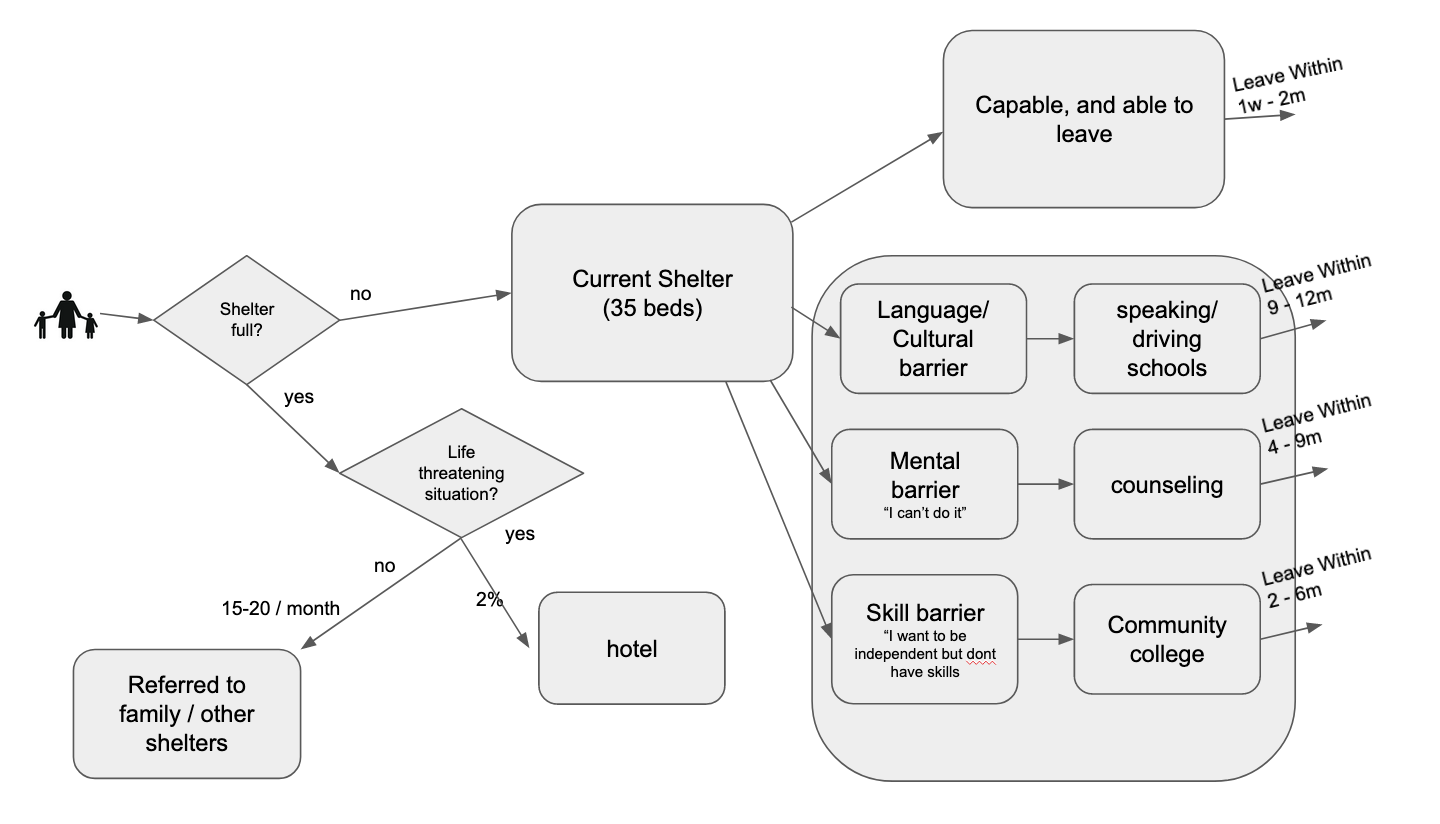
# Background and use case

## Description of use case (Abused women’s shelter)

We have chosen to combine (a) Language oriented problem and (b) Application oriented problem from the rubric to learn a new simulation package develop and deliver a solution to a real-life problem for our project. For the learning the language part of it, we have explored the SimPy open source Python package and learnt how to use it. We have chosen to apply simulation using Simpy, to benefit a non-profit women's shelter. The non-profit women's foundation we volunteer with provides services for abused women in North Texas. One of the services is to provide a safe shelter for abused women and their families. The foundation is now at capacity with their shelter. It is actively seeking to apply for a loan to create another shelter. The cost of the new shelter is directly proportional to the number of beds. Therefore, it has to determine, and explain in the business case, the optimum number of beds needed to cater for the growing needs. This would form the basis of the funding request from the shelter to expand and create the required additional capacity.

Our endeavor is to use simulation to show that at the current rate of intake, the existing shelter (and beds) are insufficient to cater to the demand. Simulation would provide a sense of how many additional beds will be needed. We interviewed the CEO for the shelter and reviewed existing metrics to understand distribution of various events and associated parameters. This model then attempts to simulate the current scenario and simulate the distribution of families this organization has to turn away because of current capacity constraints.

## Case types and processing of abuse victims



There are essentially four different ways to process the incoming abuse victims, they are described below:

The shelter does not force any of its client out ever, so the outflow is completely variable and needs to be modeled as well. The clients successfully leave the shelter in 4 scenarios:

1. **CASE 1: (CAPABLE) Capable and quickly back on their feet:** These women have occupational skills, know the culture, the English language, and are mobile. They just need a short respite at the shelter until they collect themselves and they are back on their feet quickly.
   * Stay at shelter: These women generally leave the shelter between 1 week to 2 months.
2. **CASE 2: (CULTURAL) Language and cultural barrier:** Some women may not even know how to speak proper english, or read or write english. They may also not know how to drive. In these circumstances, the organization provides opportunities for driving and other basic language training to get these women to start becoming self-sufficient.
   * Stay at shelter: These women and their families may have the longest stay in the shelter, ranging from 9-12 months.
3. **CASE 3: (MENTAL) Mental barrier:** Some women are not self-confident to take on providing for their families, as they may have never done it, or not done it for a long time. For them, this organization provides counseling and coaching services to build up their confidence. Along with such services, other vocational training opportunities are also provided to these women.
   * Stay at shelter: These women are not starting from the cohort in Group 2, but still have a considerable journey ahead of them to become self-confident and then self-sufficient, They can end up staying at the shelter for 4-6 months but for some, the stay can also a long tail approach 9months.
4. **CASE 4: (SKILLS) Skills barrier**:These women want to be on their own, and have the motivation and desire. However they may not have the required skillset to provide for their families. The women's shelter provides funds for vocational education and options associated with the community colleges to help such women. (this may always not be a viable option depending on the circumstances of the client)
   * Stay at shelter: These women are typically back on their feet independently between 2-6 months.

# Main findings

## SimPy building blocks – how it works

SimPy is a common and open-source simulation library in Python. It provides several useful features and we have learnt the basics and used it to implement a discrete event simulation scenario.

SimPy uses the concept of event generators (which essentially store a “FEL” or Future events list that can be called in chronological or any pre-determined sequence using a simple yield statement).

All “generators” are written as processes. These processes perform event orchestration – e.g. Customer arrivals, servicing a customer etc. At a very high level processes are what execute the events created by the “event generators”.

## Our code architecture

* Various events have been modeled as their own methods.
* A “main” program calls the methods in sequence to mimic the workflow specific to the arrival, processing and sending-off of abuse victims.
* Each of the methods fire in sequence as specified in the main method, to process the abuse victim and their families.
* All metrics generated as byproduct of the simulation is stored in dataframes/lists, to be used for visualization and user analysis.

## User Guide

* This user guide section talks through how our application can be re-used for similar situations. We have used a variable based approach to ensure this code can be re-used easily for similar use cases. We reviewed samples and interviewed the CEO of women’s shelter to understand the distributions that various events follow and realistic parameters they would take.
* We have included our impression of the SimPy package and its pros/cons under a separate section. However this user guide is not a user guide to the SimPy package or a tutorial, as that information is easily available online from many reliable sources.
* Consequently, this application would only apply in a social-service situation where events and distributions can be reasonably expected to be similar.
* We have parameterized all the parameter values, so as anyone tries to re-use our application, they can update the parameters to values applicable to their specific use-case.
* The first step is to validate the three key events, and if the distribution/logic depicted below apply to your use case:

|  |  |  |  |
| --- | --- | --- | --- |
| Event | Description | Distribution | Notes |
| Arrival of abuse victims | This models the arrival of abuse victims | Poisson | This is a non homogenous Poisson process and varies depending on month of the year. |
| Processing of abuse victims | This describes how long victims and their families stay at the shelter. | Static number based on rules | Since the stay is strictly based on case type, there is no variability and hence no distributin here. |
| Taking leave of abuse victims | This describes what happens to victims and families at the end of the stay | Rule based static number | At the end of the stay period victims and their families have to leave due to limited capacity at the shelter. |

* All our code is in one consolidated Jupyter notebook. Once the simulation runs, the results are automatically fed into various metrics reports.
* Before you start the simulation run, please remember to update the following parameter values, depending on the specifics of your use case:

|  |  |  |  |
| --- | --- | --- | --- |
| Event | Description | Distribution | Notes |
| Arrival of abuse victims | This models the arrival of abuse victims | Poisson | This is a non homogenous Poisson process and varies depending on month of the year. |

* You will have to execute the reporting cells in the notebook to view the various metrics that visualize the output parameters.

## Pros / Cons / Insights

The visualization capabilities of SimPy are severely lacking and users will have to build their own.

Through the use of generator functions SimPy processes generate events and interact with other processes. Several interacting processes can mimic complex super processes like supply-chain and distribution, healthcare servicing, electronic retail store sales and distribution amongst others. We used SimPy only in the context of discrete event simulation and our insights are restricted to this space.

## Comparison with Arena

The biggest difference with arena is the lack of “configurability” in SimPy. Naturally, since this is a package and not a full-fledged application all classes modeled in SimPy are specific to the chosen distribution.

Arena on the other hand allows users to change distributions on an event with a few clicks.

SimPy however gives user a lot more flexibility by controlling the behavior of the model. Example, you are restricted to pick only the distributions that Arena displays as options in the drop down list. With SimPy you can chose to implement any distribution you want that can be coded up.

# Output data from use case

## Occupancy

* This metric measure how busy the shelter is over time with occupancy expressed in person-nights over the 12 months.
* The bar graph shows the actual values from the first run. We subsequently did three runs of our model and the second line graph shows the mean occupancy values across the three runs over the 12 months as a solid line. The dotted lines show the actual run values.

Background pattern

Description automatically generated

Chart, line chart

Description automatically generated

## Un-utilized capacity

* This metric measure how much surplus capacity the shelter carries, it’s a mirror opposite of the occupancy metric above.
* The bar graph shows the actual values from the first run. We subsequently did three runs of our model and the second line graph shows the mean surplus capacity values across the three runs over the 12 months as a solid line. The dotted lines show the actual run values.
* It’s important to note here that because of the unpredictable arrivals and the differing duration of stays based on the case, there will always be some unused beds on some days due to the timing difference. We see this play out clearly in the simulation modeling, and at the small numbers, this manifestation does not imply the shelter is under capacity.

Graphical user interface, text, application

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

## Abuse victims turned away over time

* This metric measures how many abuse victims were denied shelter because of capacity issues.
* The bar graph shows the actual values from the first run. The second line graph shows the mean values (of abuse victims turned away) across the three runs over the 12 months as a solid line. The dotted lines show the actual run values.

Background pattern

Description automatically generated

Chart, line chart

Description automatically generated

## Abuse victims sent to hotel over time

* This metric measures how many abuse victims were sent to a hotle since the shelter is out of capacity.
* The bar graph shows the actual values from the first run. The second line graph shows the mean values (of abuse victims sent to hotels) across the three runs over the 12 months as a solid line. The dotted lines show the actual run values.

Background pattern

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

## Conclusions

The simulation shows there is a consistent and large number of abuse victims being turned away due to shelter’s over capacity situation. This can be used as a business case to also backup what is happening in reality, to re-affirm that the over-capacity is not a one time phenomenon, but a reality that will continue given the nature of the processes. This can make a strong business case to seek funding to expand the capacity of the shelter.

# References

* <https://simpy.readthedocs.io/en/latest/simpy_intro/> - Team SimPy
* <https://simpy.readthedocs.io/en/latest/> - Team SimPy
* <https://www.youtube.com/watch?v=0osGrraoCX0> – Kevin Conrad
* <https://www.youtube.com/watch?v=NypbxgytScM> – Real Python
* <https://www.lincs.fr/events/simpy-package-tutorial/> - Thomas Tournaire (Nokia Bell Labs France)