cloudes

rethinking
how we
LEARN
BUILD &
PLAY
with Simulations

preface

Computer literacy and STEM (Science, Technology, Engineering, and Math) education are today at the forefront of educational efforts. They both have the potential of making people, young and old, participants of the ongoing technological revolution by opening the doors to entrepreneurship and well-remunerated jobs. Computer literacy is considered crucial as reflected by efforts such as those of code.org and scratch.org. The increased presence of social media, mobile applications, and mobile devices have made computing ubiquitous and computer programming a desirable and sought after skill that can be applied in any field where computers are used. STEM has been and will be the driving force behind ocean and space exploration, advanced manufacturing, robotics, biotechnology, and transportation to mention a few. STEM education is also considered crucial as reported by the National Science Foundation's National Science Board in 2007 and reflected by efforts at the national (http://www.ed.gov/stem) and state levels (washingtonstem.org).

While be the best way to learn programming is to write programs and to learn about STEM is to participate in science, technology, engineering, and math activities, getting enough exposure to those activities becomes a challenge.

One approach to getting that exposure is through modeling and simulation (M&S). M&S provides the means to capture a real or imaginary system in a computer and ask questions about that system. M&S helps develop the ability to 1) meaningfully simplify a complex problem; 2) capture the problem in a model; 3) describe the model in a computer language, 4) collect meaningful input data; 5) execute the model over time; 6) obtain and analyze results and 7) make inferences about a potential solution to the problem. Further, models and simulations expose users to mathematics, logic, probability and statistics along of developing problem solving and analytical skills.

cloudes User Manual v. 3.14
Copyright © 2014 Virginia Modeling, Analysis and Simulation Center
Testing Site: test1.cloudes.me
Human Dynamics Lab@VMASC/ODU
ISBN: 978-0-9909015-0-1

While M&S can be applied to areas such as physics, biology, oceanography, aeronautics and astrophysics, systems engineering, organizational management, and traffic engineering among others, our focus is on simulation for engineering and management.

cloudes focuses on understanding and addressing problems where queues are prevalent like manufacturing, banking and emergency systems. These types of simulations, called discrete-event simulations (DES), are some of the most studied and easy to learn. The focus on queues provides the ability to observe systems in reality, capture data about those systems, and establish a correspondence between model and system that facilitates the validation of models.

cloudes does so by providing a simulation environment where simulations can be designed, created, modified, executed, shared, and played with across computing platforms and operating systems by using just a web browser. **cloudes** is oriented to non-experts in general and to students and educators, in particular. It provides a simple interface with basic and intermediate features that would allow users to build complex simulations quickly and proficiently.

Our desire is that **cloudes** will help increase awareness in computer literacy and STEM education. Further, we hope that educators and students use **cloudes** to provide teaching and learning in the context of real and observable systems so concepts like theoretical distributions, simulation design, algorithms, and verification and validation make a bit more sense in their minds.

Thank you,

cloudes Team
Human Dynamics Lab @VMASC/ODU
1030 University Blvd.
Suffolk, VA 23435

our goal
is to make
SIMULATIONS
available to
ANYONE

table of CONTENTS

TABLE OF FIGURES

TABLE OF DOODLES

chapter 1

A BIT OF context

Figure 1.A Queue

LET'S START THIS CHAPTER WITH AN EXERCISE.

Go to http://vimeo.com/4633517 and watch the video.

Now, think for a moment and make a mental list of where you most commonly see lines/queues. Come up with at least **FIVE PLACES** (Hint: see Figure 1.A Queue) where queues generally happen. Go to the next page when ready.

EXAMPLES:

- School cafeteria
- Movies
- Coffee shop
- Grocery store
- DMV (department of motor vehicles)
- Restaurants

Did you guess some of the places on the list? What others did you think about? Please add them to the list!

Now write five reasons why you think queues form in these places? Think for one minute and write your answer:

____
____
____
____
____
____

Lines/queues are important to study because they might be a manifestation of inefficacy in a system. Queues can be caused by insufficient or unreliable resources, inefficient scheduling or a poor layout. Businesses and corporations study queues to improve productivity, customer experience and profitability.

EXERCISE 1A: Visiting your favorite fast food restaurant (part 1)

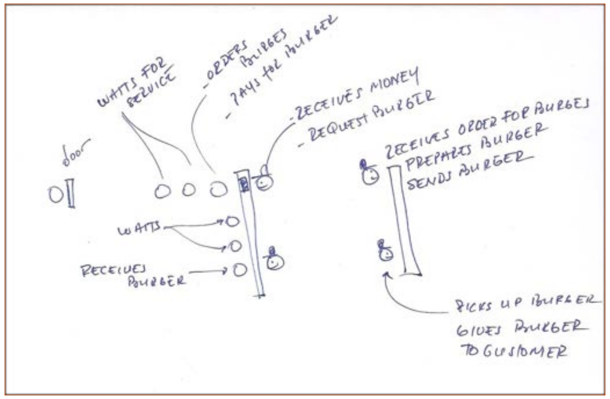
Imagine that you are hired by the manager of your favorite restaurant and given the task of figuring out how to reduce the length of queues and thereby help improve the customer's experience.

In the box below (Doodle 1), draw a diagram of the process of buying your favorite fast food meal. When done, go to the next page.

Doodle 1. Your Drawing

There is no "right" way to draw this. Any diagram that captures your observation will do.

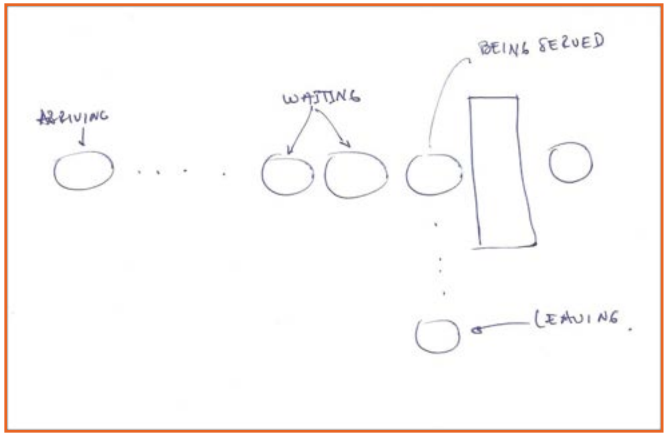
Doodle 2 shows a model (an informal conceptual model to keep up with simulation jargon) of a fast food buying process in a restaurant. A customer enters the store (door), then to a waiting area where they wait for service; if the cashier is free, order and pay for their food; after paying they move to a service area where they receive their order once it is ready. If the cashier is busy, they wait for their turn. Once an order is placed a cook prepares the meal and places it in a "meal line" where it is picked up by a server and given to the customer; once the customer has their order they exit the store.



Doodle 2. A Model of a Fast Food Restaurant

Many variations of this process are possible. We can have more cashiers, more cooks and no service area. Think of all the different configurations you might have encountered in the many places where you have encountered queues. Usually, they can be simplified to one or more servers helping one or more customers in a physical space. The process can be further simplified as a customer arriving at a place and if the sever is free, advancing to the server to receive a service before exiting the place. If, on the other hand, the server is busy (meaning another customer is already being served), the customer waits for their turn and for the server to be free in order to be served.

Doodle 3 shows a simpler model where customers arriving at a fast food restaurant waiting in line to be attended to, being served by a store employee and leaving the place.



Doodle 3. A Simpler Model of a Fast Food Restaurant

In Doodle 3, we consider that customers arrive at a certain interval (amount of time elapsed between two arrivals). We assumed that the transaction includes ordering, paying and receiving your meal and that each transaction takes some amount of time to be performed.

Now write three reasons why you think there may be queues in these places?
Think for one minute and write your answer:

____

____

____

____

____

After writing your answer, please move to the next page.

Any modeling process requires assumptions. Assumptions allow the modeler to simplify the modeling process by purposefully eliminating, combining, or modifying processes/components. Always start simple

EXERCISE 1B: Visiting to your favorite fast food restaurant (part 2)

Now that you have captured a doodle of the process of buying food in a fast food restaurant, you are asked to build a simulation. The simulation will be used for evaluating how many servers are needed in the restaurant in order to minimize the time customers spend waiting for service and their orders.

Figure 2 shows customers entering the store (arrivals), waiting in line if necessary (queuing), ordering (processing) with one of the three servers (resources) and then leaving (exiting).

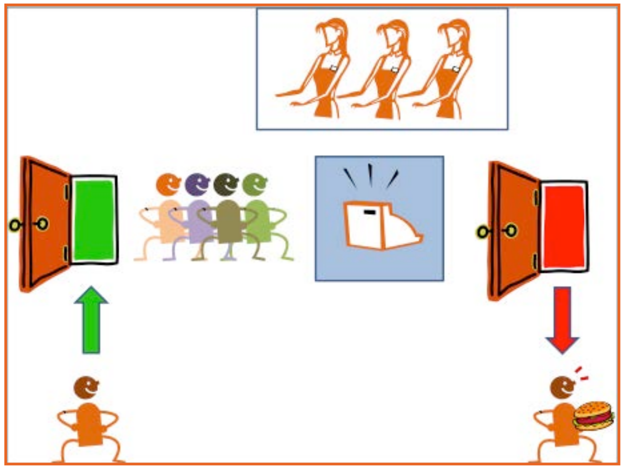


Figure 2. A Model of a Fast Food Restaurant

Figure 3 is the computational model (another name for simulation) that corresponds to the conceptual model captured in Figure 2. Figure 3 was built using cloudes, the tool that is the focus of this handbook.

Note: Doodle 2, Doodle 3 and Figure 2 are all conceptual models of the same process and very similar in that they all capture the system of interest but with different levels of detail. You can always rely on more elaborated means of capturing a conceptual model like frameworks or languages. Using a doodle at this point is sufficient to familiarize you with the practice of relying on a conceptual model before building a simulation.

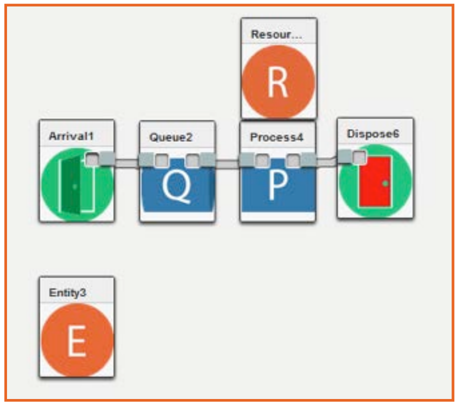


Figure 3. A Computational Model in cloudes of a Fast Food Restaurant

We will learn the basics for building this model in the next chapter.

summary

Modeling and Simulation is a great way to learn about the world that surrounds us. It also allows us to study things that do not exist yet or to which we do not have direct access. DES is a particular paradigm (approach) of Modeling and Simulation that focuses on capturing things that have identifiable events that occur at specific points in time. DES can be used to model things like movie theaters, DMV, and grocery stores. In practice, DES is used in resource intensive processes that involve queues such as manufacturing, business and healthcare.

In order to build Models and Simulations we can rely on artifacts called conceptual models. A simulation is a computational equivalent of a conceptual model that we can use to conduct experiments.

chapter 2

getting to KNOW cloudes

BEFORE WE START BUILDING SIMULATIONS, LET'S BRIEFLY TAKE A LOOK AT SOME IMPORTANT DETAILS ABOUT CLOUDES.

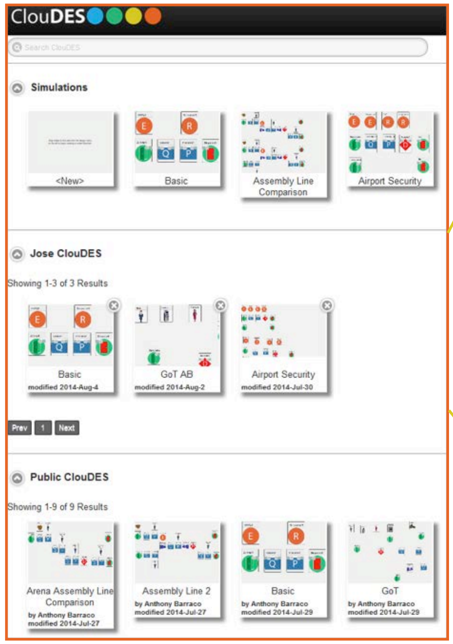
CREATING AN ACCOUNT

Creating an account is pretty simple. Just click **Sign Up**, and fill in the necessary information. A verification email will be automatically sent to your email account. After you verify the link between your email and the cloudes account, your account will be activated.
Note: Check your Junk Mail just in case the activation email gets misplaced.

EXPLORATION SECTION

Figure 4 shows the exploration section which has three components:

- Sample Simulations that are provided with cloudes as examples for building simulations;
- Personal cloudes (cloud-based simulations) which are identified by your name. These can be Private or Public; and
- Public cloudes which are simulations made available by all users.



Use this section to delete a simulation by clicking the (x) on the upper right corner of simulations' icons. Also, there is a label called Visibility in the personal simulations. This label will tell you if your simulation is public or private without opening a simulation and looking at the Environment Tab (explained later).

Figure 4. cloudes Explore Section

DESIGN SECTION

Figure 5 shows the design section. This section opens up after loading a simulation or creating a new one. On the left side, you have the name of the simulation on top and two major sub-sections: one with buttons (Control Box) and one with tabs.

The design section has a series of buttons that allows you to create, save, load and animate your simulation. In addition, you have an **Environment** tab, a **Design** tab, a **Replications** tab and a **Tools** tab.

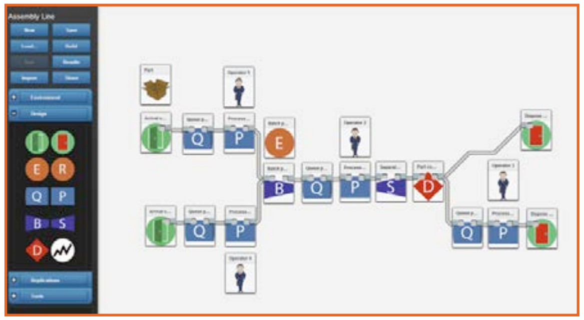
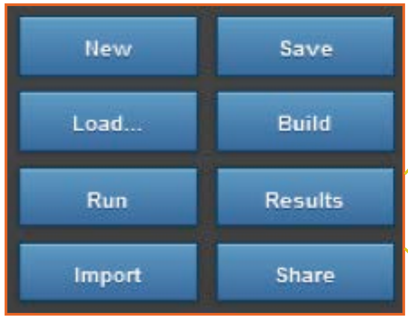


Figure 5. cloudes Design Section

CONTROL BOX

The control box is at the top left of the Design Area (Figure 6).



Build must occur for Run to be activated for new or modified simulations. For simulations that have already ran, you do not need to build again. Also, after you use Build, cloudes generates your simulation results as well.

Figure 6. Control Box

BUTTONS AND THEIR FUNCTIONS:

- **New**: Creates a new simulation
- **Save**: Saves the simulation. Make sure you save your work frequently as the system DOES NOT save automatically.

Note: The simulation will NOT be saved unless it has a name. Be sure to use the Environment Tab to name your simulation (next chapter)

- **Load**: This takes you to the Explore Section where you can load from samples, personal, or public simulations.
- **Build**: After you finish putting the simulation together (designing) press the Build button. This action sends the design to the server where the simulation is executed.

Note: Refer to the message box at the bottom left to see if the system found errors while building the simulation.

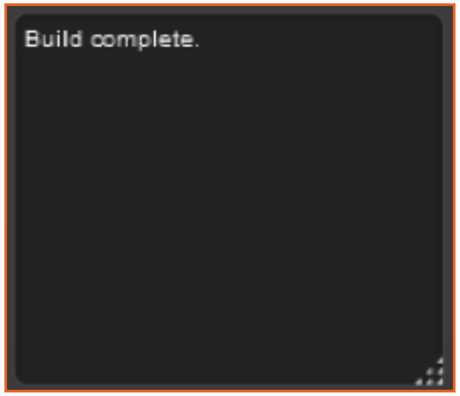
- **Run**: After building the simulation you can view animations using the Run button.
- **Results**: If you don't want to see animations, you can still have access to results and export them to your computer as an .xls file for further analysis.
- **Import**: Allows you to add a simulation to an already loaded simulation.
- **Share**: Allows you to share your simulation with others via email or social media.

Note: data collected through counters can be exported and imported for sharing purposes.

MESSAGE BOX

The message box (Figure 7) provides feedback on errors found while building the simulation. If errors are found, the program will tell you where the errors are. Address the errors and build again. You will get a "Build Complete" message if no errors are found.

You can find the message box at the bottom left of the Design Area.



Build complete.

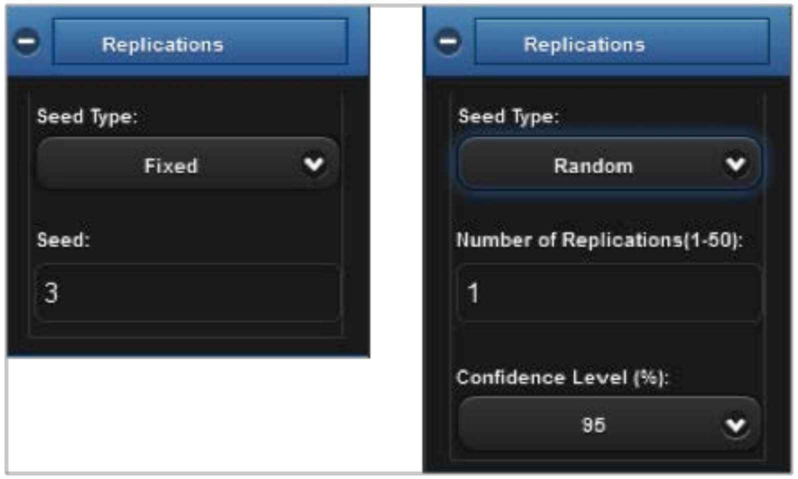
Figure 7. Message Box

REPLICATION TAB

Simulations rely on pseudorandom number generators to generate a sequence of numbers close to random. They are important to 1) replicate results and 2) inject variability in results akin to what happens in real life when repeating a process where values change. Figure 8 shows the fixed and random seed type.

The Fixed seed type allows users to replicate results of a particular experiment as long as the same seed is used. The Random seed type lets clouDES' random number generator create a new seed every time the simulation is run.

If the simulation has more than one replication (up to 50 currently), clouDES will provide the values per run and basic statistics of the variables considered. Note: if more than one (1) replication is needed, clouDES will allow you to export results for analysis in a spreadsheet. You will not be able to animate the simulation.

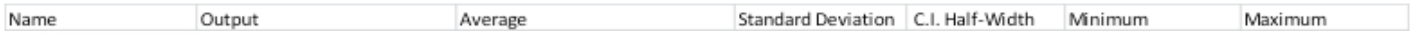


Replications
Seed Type: Fixed
Seed: 3

Replications
Seed Type: Random
Number of Replications(1-50): 1
Confidence Level (%): 95

Figure 8. Replication Tab

Figure 9 shows the header of the summary section (when exported in a file) after running a simulation more than once using the Random seed type.



Name	Output	Average	Standard Deviation	C.I. Half-Width	Minimum	Maximum

Figure 9. Header of Summary Results

For instance, if running a simulation 50 times with a 95% confidence level, you obtain that the variable Arrival (Name) has an average value of 477.76 (Output) with a standard deviation of 22.7 and a margin of error of 6.3 (C.I. Half-Width). This can be interpreted as: after 50 runs and at a 95% confidence level and a margin of error of 6.3 (95% of the time, the results will be within 6.3% of the mean), the estimated value of Arrival is 477 entities (total) with a standard deviation of 22.7.

summary

cloudes relies on the idea that if you are familiar with using the web, you are already familiar with **cloudes**. Processes like account creation and simulation sharing ease your way into the system. There are two major sections in **cloudes** : Exploration and Design. The former allows you to discover examples and simulations created by others while allowing you easy access to your personal simulations. The latter allows you to create, design, and ultimately play with your simulations so you can see the behavior of the system you modeled and/or analyze simulation results for planning or decision making purposes.

chapter 3

building a SIMULATION

THERE ARE TWO WAYS OF BUILDING A SIMULATION IN cloudes:

- BY USING AN EXISTING SIMULATION
- BY BUILDING IT FROM SCRATCH

Using an existing simulation enables you to modify one of the sample simulations, one of the public simulations or one that was shared with you. It is as simple as loading any of these simulations in the design area and modify it by adding or eliminating design elements and/or changing existing values to the ones you are interested in. Building a simulation from scratch means starting the modeling process with a conceptual model. However, some people do not use a conceptual model to build a simulation. While this may be true in some cases, we recommend starting with a conceptual model as other users may need the model to evaluate the simulation.

suggestion

To become familiar with cloudes , start with a simple simulation so that it can be altered easily. You can add more complexity to the simulation as you get more comfortable with cloudes.

Keep in mind that before you modify a simulation, create a conceptual model of it so that you can evaluate if the simulation can be altered in a manner that satisfies your requirements. Models and simulations are created to answer specific questions. As such, the question you are asking might not be the one asked by the original user.

STARTING WITH AN EXISTING SIMULATION

1) In cloudes Exploration Section select one of the sample simulations (Figure 10).

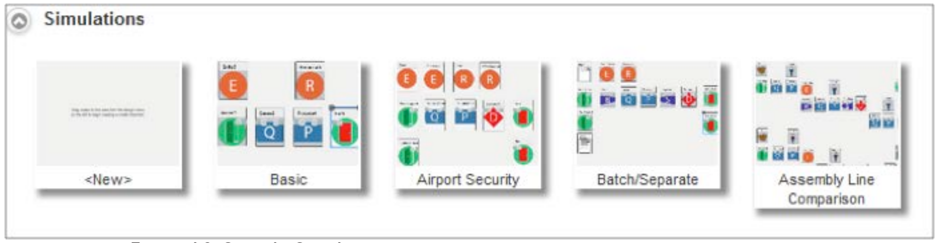


Figure 10. Sample Simulations

Start with something simple like the Basic simulation and expand it or change values to reflect your system of interest. These sample simulations showcase a progressive level of complexity or use. For instance, the Airport Security simulation is a bit more complex than the Basic. The Batch/ Separate provides an example of the use of those particular functions but it is not applied to a particular case. Lastly, the Assembly Line Comparison focuses on the ability of having two simulations running at the same time with the purpose of comparing their output based on variations of the simulations.

2) In cloudes Exploration Section (Figure 11) select one of the public simulations (Public cloudes).

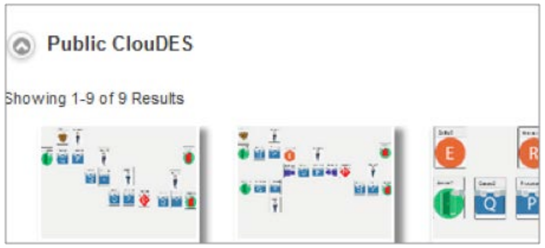


Figure 11. Public Simulations

As you move your mouse pointer over a simulation you will see a brief description of the simulation. If you want more information, open the simulation, go to Environment and click on Detailed Description. Keep in mind that this information is optional. However, we strongly suggest you fill this information with the following details:

- What is the purpose of the simulation?
- What input data is needed to run the simulation?
- What behaviors and outputs should we observe?

It is important to note that each of the simulation elements has a Description field. Make sure you obtain/ provide relevant information there as well.

STARTING FROM SCRATCH

Here you have two options:

Option 1 - Load Page: select New from the sample simulation list (Figure 12).

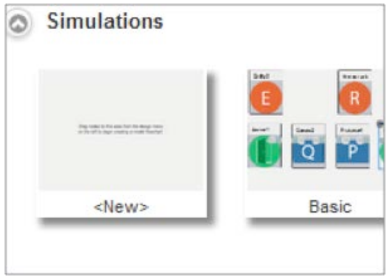


Figure 12. New Simulation in Explore Section

Option 2 - Design Page: Click on New (Figure 13).

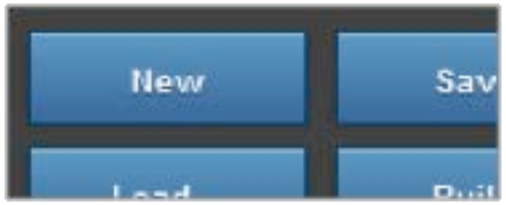


Figure 13. New Simulation in Design Section

In order to build more complex and/or sophisticated simulations, you need to become familiar with the components of cloudes. The following section will give you further insight on cloudes and provide details about its components.

ENVIRONMENT TAB

Let's say you selected the Basic Simulation as shown in Figure 14. After selecting the simulation, click to open the Environment Tab.

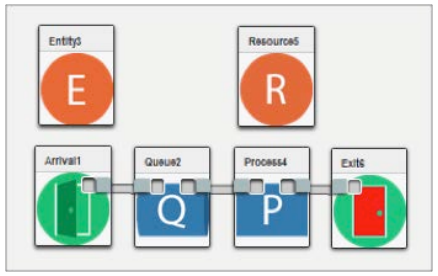
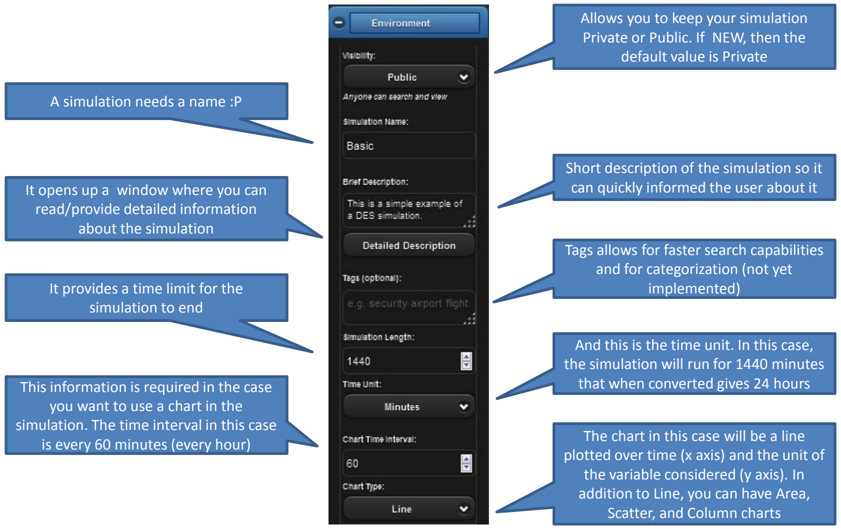


Figure 14. Basic Simulation

The Environment Tab provides the general characteristics of the simulation as shown in Figure 15.



Environment

Visibility: Public

Anyone can search and view

Simulation Name: Basic

Brief Description: This is a simple example of a DES simulation.

Detailed Description

Tags (optional): e.g. security airport flight

Simulation Length: 1440

Time Unit: Minutes

Chart Time Interval: 60

Chart Type: Line

A simulation needs a name :P

It opens up a window where you can read/provide detailed information about the simulation

It provides a time limit for the simulation to end

This information is required in the case you want to use a chart in the simulation. The time interval in this case is every 60 minutes (every hour)

Allows you to keep your simulation Private or Public. If NEW, then the default value is Private

Short description of the simulation so it can quickly informed the user about it

Tags allows for faster search capabilities and for categorization (not yet implemented)

And this is the time unit. In this case, the simulation will run for 1440 minutes that when converted gives 24 hours

The chart in this case will be a line plotted over time (x axis) and the unit of the variable considered (y axis). In addition to Line, you can have Area, Scatter, and Column charts

Figure 15. Environment Tab

DESIGN TAB

This tab allows you to build a simulation from scratch and/or add new elements to an existing simulation. Figure 16 shows the snapshot of this tab and a description of its blocks.

Drag, drop and connect these blocks to build a simulation. Entities and Resources do not need to be connected because they are called by other blocks like Arrival and Process respectively. Make sure that before attempting to build the simulation, you open each block and enter its Properties.

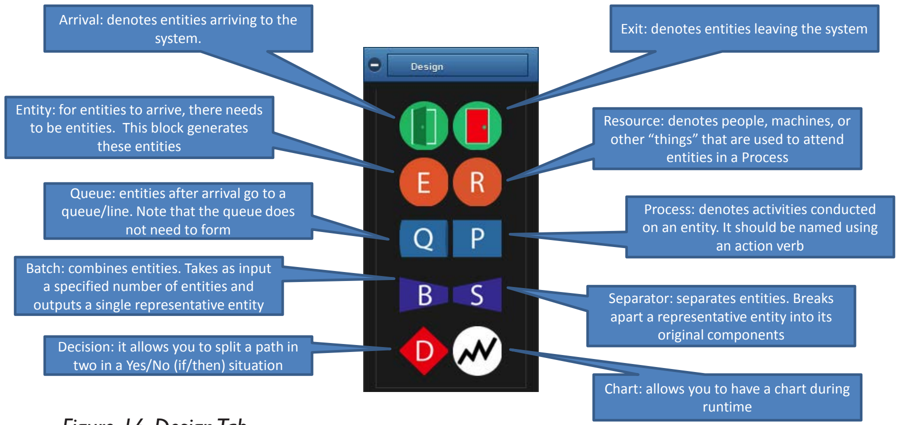
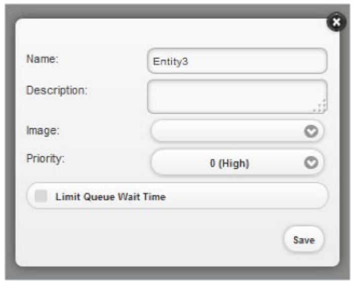

Design
Arrival: denotes entities arriving to the system.
Entity: for entities to arrive, there needs to be entities. This block generates these entities
Queue: entities after arrival go to a queue/line. Note that the queue does not need to form
Batch: combines entities. Takes as input a specified number of entities and outputs a single representative entity
Decision: it allows you to split a path in two in a Yes/No (if/then) situation
Exit: denotes entities leaving the system
Resource: denotes people, machines, or other "things" that are used to attend entities in a Process
Process: denotes activities conducted on an entity. It should be named using an action verb
Separator: separates entities. Breaks apart a representative entity into its original components
Chart: allows you to have a chart during runtime
E
R
Q
P
B
S
D

Figure 16. Design Tab

BLOCK PROPERTIES

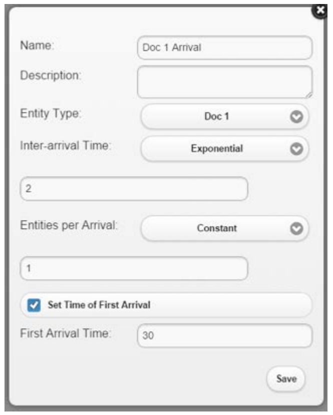
Each block has a set of Properties. There are two properties that all blocks have: Name and Description. They come with a default name that you can change and some allow you to change their icons. Description property allows you to provide/obtain information about that particular block. Figure 17 shows the Property window for Entity.



Name:	Entity3
Description:	
Image:	
Priority:	0 (High)
☐ Limit Queue Wait Time	

Figure 17. Property Window - Entity

Besides Name, Description and Image there are two other properties: Priority and Limit Queue Wait Time. The former determines where the entity is placed in the queue. The later allows entities to leave the system if they wait for too long. Figure 18 shows the **Property window for Arrival.**



Name:	Doc 1 Arrival
Description:	
Entity Type:	Doc 1
Inter-arrival Time:	Exponential
	2
Entities per Arrival:	Constant
	1
☑ Set Time of First Arrival	
First Arrival Time:	30

Figure 18. Property Window - Arrival

Besides Name and Description, the **Arrival Block** has the following properties:

- **ENTITY TYPE**: Entity associated with this Arrival
- **INTER-ARRIVAL TIME**: This parameter provides the probability distribution from which the time between entity arrivals is drawn. In other words, it captures the behavior of how often entities arrive to the system per unit of time. It has one or more accompanying fields for parameters of the probability distribution.
- **NUMERICAL VALUES** (values "2" and "1" in Figure 18): This is the parameter of the probability distribution like the mean or standard deviation (depends on distribution) for inter-arrival time and entities per arrival respectively.
- **ENTITIES PER ARRIVAL**: Entities can arrive more than one at the time. The default value is one entity at a time. It also has one or more accompanying fields for parameters of the probability distribution.
- **SET TIME OF FIRST ARRIVAL**: Allows you to define when the first entity (s) arrives.

Figure 19 shows the **Property Window for Queue**.

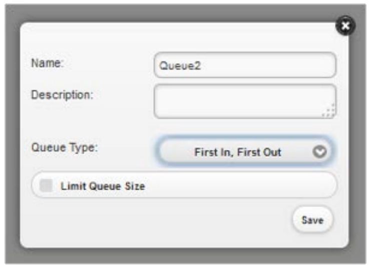
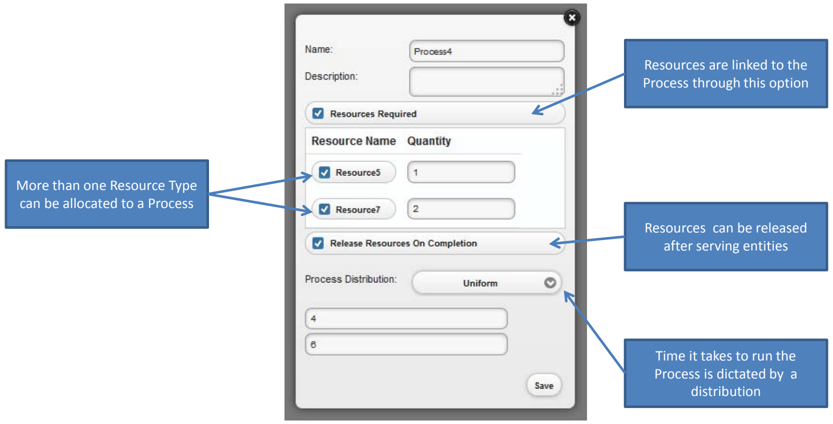

Name: Queue2
Description:
Queue Type: First In, First Out
Limit Queue Size

Figure 19. Property Window – Queue

In addition to name & description, the **Queue Block Properties** have the following parameters:

- **QUEUE TYPE**: Queue Type allows you to define the queue discipline (First In- First Out or Last In- Last Out).
- **LIMIT QUEUE SIZE**: This option determines the maximum number of entities in a queue.

Figure 20 shows the **Property Window for Resource**.


Name: Process4
Description:
Resources Required
Resource Name	Quantity
Resource5	1
Resource7	2

Release Resources On Completion
Process Distribution: Uniform
4
6
Save
Resources are linked to the Process through this option
More than one Resource Type can be allocated to a Process
Resources can be released after serving entities
Time it takes to run the Process is dictated by a distribution

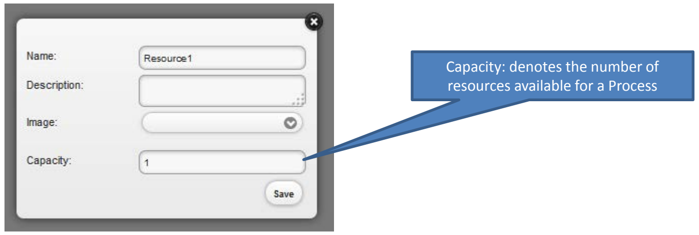
Figure 20. Property Window - Process

In order for a process to run, one or more resources must be available if resources are required. If resources are not required, the process will run immediately regardless of how many other processes are running. If resources are required, the user should select the "Resources Required" checkbox.

When resources are required, the user specifies which resources are required and in what quantity. An entity can only enter a process if a resource is available. If not, the entity must wait in the queue. If you want to make a resource available after it is done serving an entity select the "Release Resources On Completion" checkbox. The Process Distribution is the time it takes the process to complete. It has one or more accompanying fields for parameters of the probability distribution.

You can model a delay in a system by having a process without associated resources.

Figure 21 shows the **Property Window for Resources**.


Capacity: denotes the number of resources available for a Process

Name:	Resource1
Description:	
Image:	
Capacity:	1

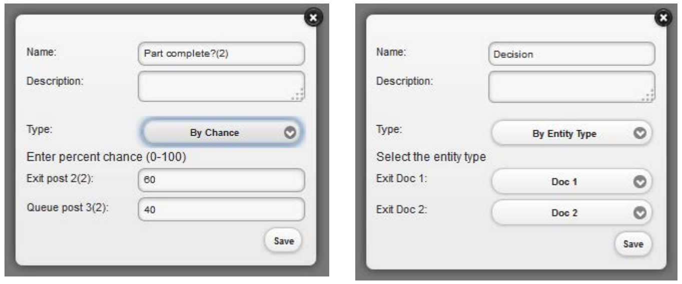
Save

Figure 21. Property Window - Resource

As shown in Figure 21, capacity captures the number of resources allocated to that process.

A Decision block allows you to select one of two paths depending on a condition. At this point, cloudes allows for two conditions: 1) based on chance and 2) based on entity type. For instance, based on chance, Figure 22 shows a setting for entities going through a Decision block. 60% of these entities will Exit (Exit post 2(2)), while 40% will go to another Queue (Queue post 3(2)).

Based on entity type, Figure 22 shows that entities named "Doc 1" will go to one exit (Exit Doc 1) while entities named "Doc 2" will go to another exit (Exit Doc 2).



Name:	Part complete?(2)
Description:	
Type:	By Chance
Enter percent chance (0-100)	
Exit post 2(2):	60
Queue post 3(2):	40

Save

Name:	Decision
Description:	
Type:	By Entity Type
Select the entity type	
Exit Doc 1:	Doc 1
Exit Doc 2:	Doc 2

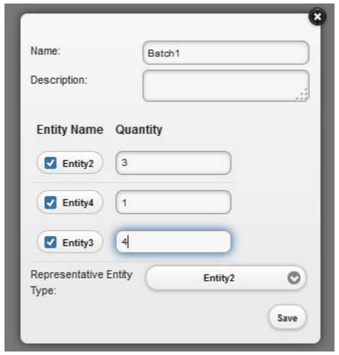
Save

Figure 22. Property Window - Decision

The Batch and Separate blocks allow you to process entities as a group and then separate them if needed. For instance, the assembly of parts into a product requires the Batch block but not the separate. On the other hand, the processing of parts in a group that later require an individual processing or a different grouping require both the Batch and Separate blocks.

The field named "Type" in the Decision block is not visible when the block is dragged into the Design Area. It must be connected to the blocks reflecting different paths to become visible. For instance, "Type" appeared in Figure 22 (left) when the Decision block was connected to an exit and to a queue.

Figure 23 shows both the **Batch and Separate Property Windows**.


Name: Batch1
Description:
Entity Name | Quantity
☑ Entity2 | 3
☑ Entity4 | 1
☑ Entity3 | 4
Representative Entity Type: Entity2

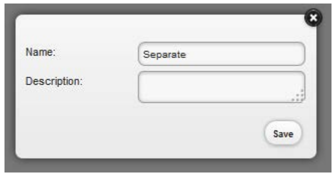

Name: Separate
Description:

Figure 23. Property Window - Batch & Separate

The Batch property window (left) shows two important aspects: 1) conforming entities and their respective amounts to create 2) a representative entity. In the example above, there are three conforming entities that when combined (3, 1, and 4) form the representative entity.

You can use an existing or create a new entity block in order to assign it to a representative entity. Computing-wise there is no difference between the two options. Modeling-wise there is a difference. Using a new Entity block suggests a different "thing" and not the modification of an existing one. For instance, a car is a new entity created by parts components and not a modified component like an engine (if using an existing Entity block).

The **Chart Block** (Figure 24) allows you to have one or more charts to monitor selected parameters during the simulation run.

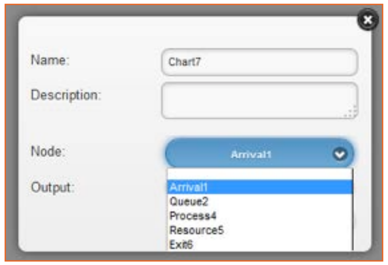

Name: Chart7
Description:
Node: Arrival1
Output:

Figure 24. Property Window - Chart

Figure 24 shows the parameters (Node) that can be graphed. It is important to note that parameters vary in the units they are measured. As such: Arrival/Exit are measured in terms of throughput (units/unit of time). Queue is measured in terms of average size. Process in terms of processing time and idle time, and Resource in terms of percent utilization.

Last, but not least, we have the **Exit Block** (Figure 25). This block marks the final point of the simulation. What entities do or where they go after this point is considered out of scope.

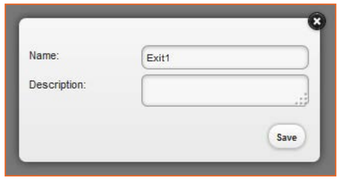

Name: Exit1
Description:

Figure 25. Property Window - Exit

RUNNING THE SIMULATION

Animating the simulation gives you a dynamic perspective of how people/goods arrive and are processed by the system. You will see how queues increase/decrease in size and the number of resources being released/used. Figure 26 shows the simulation animation control panel. This panel allows you to slow/speed up the simulation, pause it, and zoom in/out among other functions. You can also see the time units of the simulation on the top left corner.

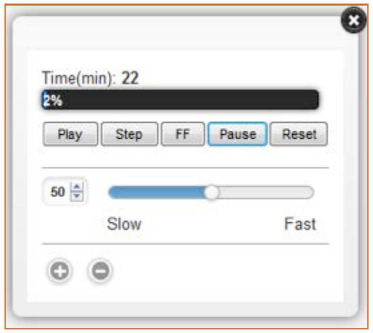


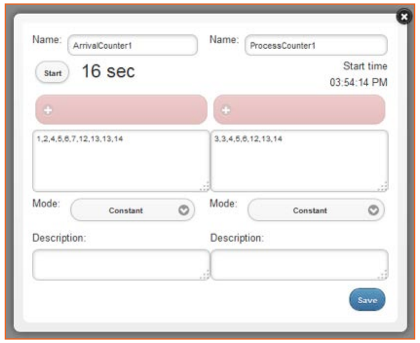
Figure 26. Simulation Control Panel

CAPTURING DATA USING COUNTERS

This is perhaps one of our favorite features of cloudes: the availability to capture inter-arrival and service time data and use it directly in your simulation.

Currently, if you want to collect and use data from a system, you need to use a chronograph or watch, and either capture time between arrivals or capture a time stamp and then calculate the difference between successive arrivals. Further, if you capture your data on a piece of paper, you still need to transcribe it to a spreadsheet and then find software that would allow you to calculate your empirical distribution from the data. If you use a computer to capture your data, you still need to find the software that calculates your empirical distribution from the data.

cloudes allows you to capture your data and use it immediately. You can have single or dual counters. Figure 27 shows a dual counter. A single counter has only one counting element. You can create and manage counters using the Tools Tab.



Name: ArrivalCounter1 Name: ProcessCounter1

Start 16 sec Start time 03:54:14 PM

1,2,4,5,6,7,12,13,13,14

3,3,4,5,6,12,13,14

Mode: Constant Mode: Constant

Description: Description:

Save

Figure 27. Dual Counter

First, the counter captures the time when you started counting. This is important so the user knows the time frame the data is useful for.

You have a Start and Save buttons that do what the names imply. After starting the counter you can Pause it and Save it. You capture the data by simply clicking on the Red button.

As you do so, the data is captured in the box in seconds. Inter-arrival times are then calculated in our servers.

notes on counters

1. SAVE YOUR COUNTERS EARLY AND OFTEN

2. Even though the data is captured in seconds, if your simulation is set up in minutes for instance, the system will do the conversion.

3. Now, you have three ways of using this data (Mode): Constant, Constant Repeated and Empirical Distribution. Constant means you will use the data as is. So, when the simulation stops using that data, there is no more activity. Constant Repeated allows you to repeat the existing data set for the simulation length. Lastly, Empirical Distribution allows you to calculate the distribution from the existing data set.

4. Simulation-build time could increase dramatically when using Constant Repeated and Empirical Distribution modes and a time unit other than seconds.

5. If you rename your counter, it will be saved with that name but the old one will be kept, so delete counters you don't need.

6. You can add more data to the counter. Go to manage and click on Edit and hit Start. This however, is not suggested unless you start collecting data at the time and day you stopped in order to keep the integrity of the data collected. This assumes conditions will remain the same. For instance, this would not apply to a school day vs. vacation day.

7. Currently, counters are associated to a user and not to a simulation. This may change because when simulations are shared and they have a counter, this is not visible to the recipient of the simulation.

8. As of publication time, we are further testing counters especially how to share them in order to facilitate group data collection. This may generate problems accessing existing counter data. If this is the case, please send us a note at note@ClouDES.me.

ummary

Creating a simulation does not have to be a complex process. It gets easier with practice and following basic guidelines. The basic premise is that one should start with a simple model and add details. Your knowledge of the system is more important to identify its key aspects, capture them and to add details when needed. Tools ought to be there to aid you in the modeling process, not to encumber it.

WHEN CREATING A SIMPLE SIMULATION FROM SCRATCH IN CLOUDES, WE SUGGEST:

- Create a conceptual model of the system of interest by drawing a doodle.
- Drag/drop Arrival and Exit first to capture the beginning and end of your simulation.
- Drag/drop Entity and Resource next. They do not need linking, but are needed by other blocks.
- Associate Entity with Arrival and populate Arrival.
- Drag/drop Queue and Process.
- Associate Resource with Process and populate Process.
- Connect Arrival, Queue, Process, and Exit.
- Build and Run (if no errors).
- View Results and play with the simulation values so you can create other scenarios (shorter inter-arrival times, more resources available, etc.
- Increase the complexity of the simulation by adding more blocks; associated them with entities and resources; populate them with values; and connect them with one another.
- Last but not least, **HAVE FUN**.
- Lastly, using data to capture the behavior of your system is important as it reflects what occurs in reality. Counters provide the means of capturing that behavior through data.

chapter 4

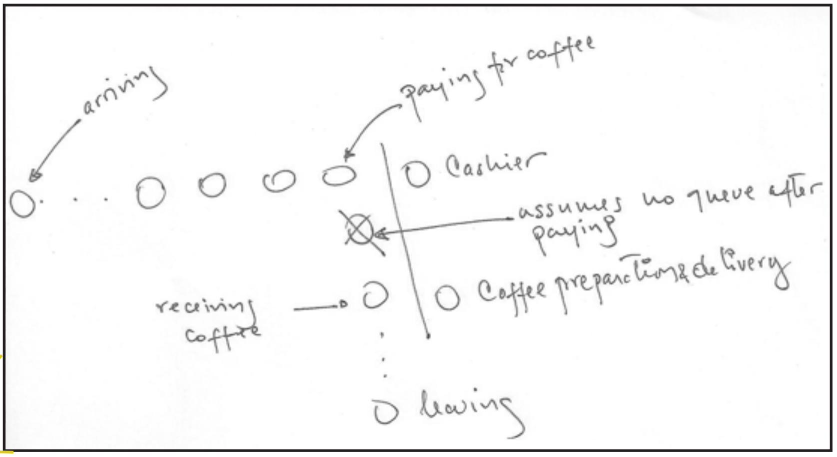
EXAMPLES AND exercises

It is important to make the model's assumptions explicit. Make sure you note them in the Environment Tab's Detailed Description field.

EXAMPLE 1: ONE CASHIER AND ONE "PREPARATOR."

The Basic simulation captures a system with one queue and one server. Let's expand this simulation to consider one cashier, that takes care of the transaction, and one server that prepares and delivers let's say coffee.

The conceptual model (informal) for the simulation is captured in Doodle 4.



Doodle 4. One Cashier and One Preparator System

The conceptual model shows that there is a queue, the person then pays for coffee, receives the coffee from someone else and then leaves the system. This model assumes that no line is formed between paying and receiving the coffee.

Figure 28 shows the simulation design in cloudes.

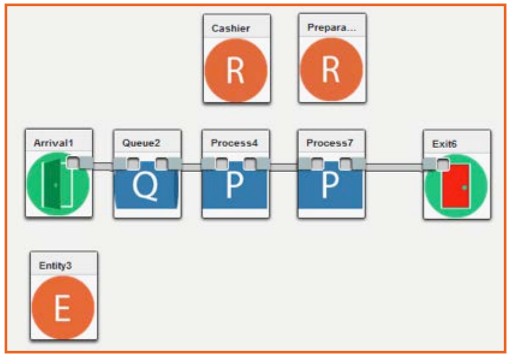


Figure 28. One Cashier and One Preparator Simulation Model

EXERCISE 1.

What were the changes to Basic? Write, or type your answer below in Doodle 5. (Hint: it requires several steps)

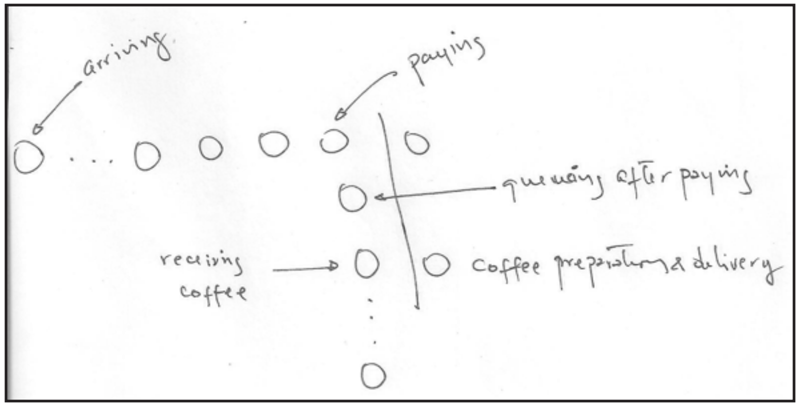
Doodle 5. Exercise box – Changes to Basic

This is the process:

1. Disconnect the existing Process from Exit
2. Add one more Resource and one more Process (remember to fill in their respective Properties. Resources' Capacity Property is one (1))
3. Associate the new Resource with the new Process
4. Connect the two processes in a sequence (associated with Cashier first) and then connect the second process with Exit
5. Build and Run

EXAMPLE 2: ONE CASHIER AND ONE "PREPARATOR" WITH AN "AFTER-PAYING" QUEUE

The conceptual model is for this simulation is captured in Doodle 6.



Doodle 6. One Cashier and One Preparator (version 2) Simulation Model

Just like Doodle 4, but we don't assume the lack of queue after paying and waiting for coffee.

EXERCISE 2.

What new element(s) is (are) needed to capture Doodle 6? Write your answer below in Doodle 7 (Hint: it is a short answer).

Doodle 7. Exercise Box – New Elements

The answer is pretty much straightforward: just add a Queue block between processes. Then, Build and Run.

What did you learn?

These two simple examples show us that you can always start simple and then add more detail to your simulations.

HOMEWORK 1: Take simulations from examples 1, 2 and 3 and 1) compare them while running. What can you see? 2) Compare simulation results. What can you say?

EXAMPLE 3: TWO CASHIERS AND TWO SERVERS.

This one is pretty simple. If we consider that there is a Queue after paying, the conceptual model would look like Doodle 6 but with two cashiers and two cooks.

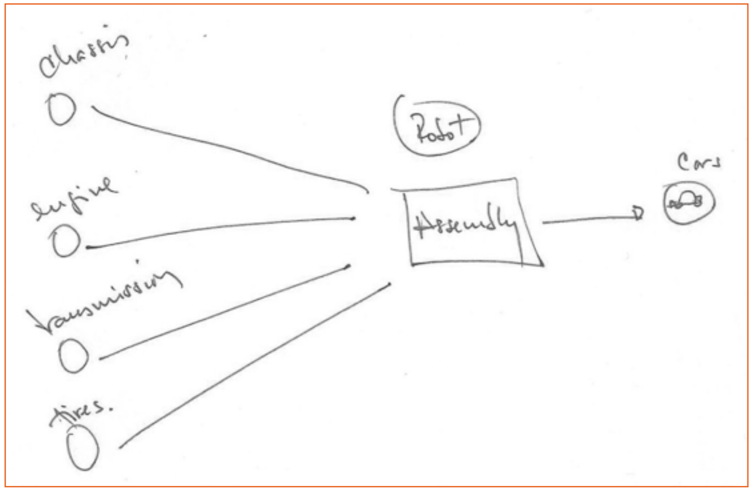
In this case, double click/tap on Resources and change capacity from one (1) to two (2). Then, Build and Run.

What did you learn?

This example shows you that you can change simulation values to explore different scenarios. In some cases you may need to do changes to the simulation (like in the example before), but in some cases all you need to do is change values in the parameters being considered.

EXAMPLE 4: LET'S BUILD A CAR

Let's assume that we are assembling cars from four parts: chassis (1), engine (1) , transmission (1) and tires (4). Let's also assume that one (1) robot is in charge of putting these parts together. Doodle 8 shows the conceptual model.



Doodle 8. Conceptual Model for A Robot-aided Car Assembly System

This doodle captures the system as four arrivals for each major component. These components then are assembled where one (1) chassis, (1) engine, (1) transmission and (4) tires are needed to make a car. The car then leaves the assembly to another process that is outside of scope for the moment.

EXERCISE 3.

Build the simulation in cloudes. Then compare it to that on next page. Use the doodle (Doodle 9) box below to help you with the conceptual model.

Doodle 9. Exercise Box

One potential simulation design is the one shown in Figure 29.

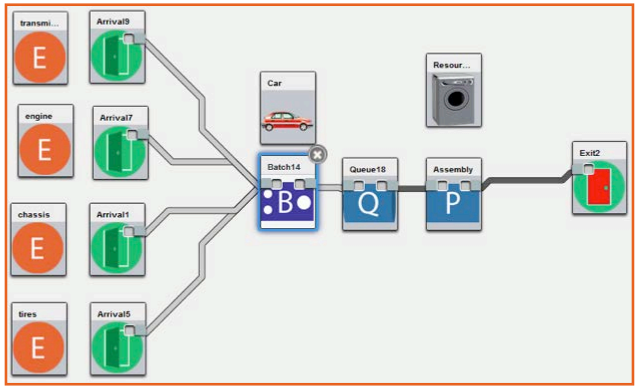
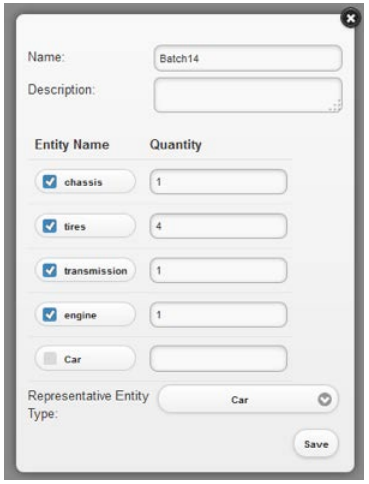


Figure 29. Simulation Model of for A Robot-aided Car Assembly System (v.1)

Figure 29 captures the model as:

- After part components arrive, there is no queue for receiving the parts
- Parts arrive to a place before assembly where groupings are organized
- These groupings go to the robot (looks like a washing machine in doodle 45) for Assembly.

The most important block to consider in this design is the Batch block. This block allows creating a representative entity called car out of 1 engine, 1 transmission, 1 chassis and 4 tires as shown in Figure 30.



Name: Batch14

Description:

Entity Name	Quantity
✓ chassis	1
✓ tires	4
✓ transmission	1
✓ engine	1
Car	

Representative Entity Type: Car

Save

Figure 30. Batch Properties for A Robot-aided Car Assembly System

Now, your simulation design might be different than that in Error! Reference source not found.. Is the design wrong? More likely, both designs are correct considering the assumptions made on each. All models and corresponding simulations are abstractions based on what we observe, what we know, and what we assume. Let's have a different design (Figure 31).

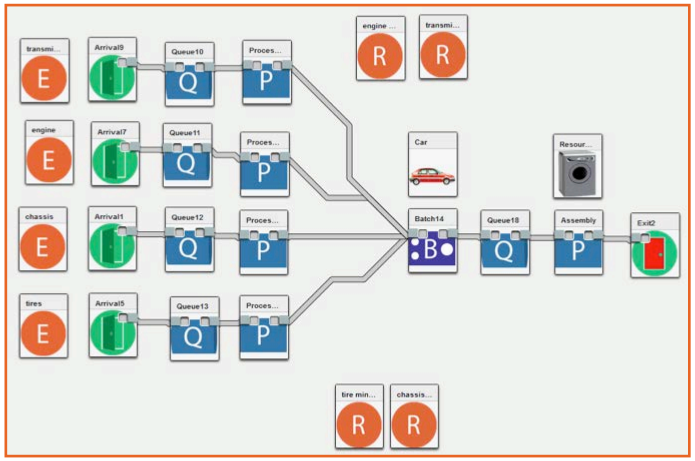


Figure 31. Simulation Model of for A Robot-aided Car Assembly System (v.2)

EXERCISE 4.

Please explain: 1) the difference between the designs from Figure 29 and Figure 31, 2) what are the design implications of Figure 29, 3) what are the design implications of Figure 31when compared to Figure 29, 4) build the simulation from Figure 31 in cloudes and 5) change parameters' values and see how the system behaves under different conditions.

____

____

____

____

____

What did you learn? This is a very comprehensive exercise. You must have realized the importance of how the model is captured as it can be captured differently even by the same person; you must have learned that each design has pros, cons and implications in terms of results and ease of design; lastly you must have learned that a simple design starts getting complex very quickly. While you may represent the system with a higher degree of closeness, it creates the downside that it is more difficult to study. So, there is always a tradeoff between complexity and how much you can explain with your models; the more complex the model, the more difficult to explain. At the same time, the more complex, the closer to reality you might be.

EXERCISE 5.

People go to an arena to see the show performed by Ms. Yellow, Mr. Green and Mrs. Purple (Figure 32).

- Three entry points
- In each entry point there is one person selling tickets
- There are 50,000 seats in the arena
- People sit arbitrarily

1. Build a simulation that captures this system
2. Add theoretical distributions and values arbitrarily
3. Capture the assumptions you make
4. Create two scenarios: 1) sell tickets in one hour and 2) sell tickets in two hours

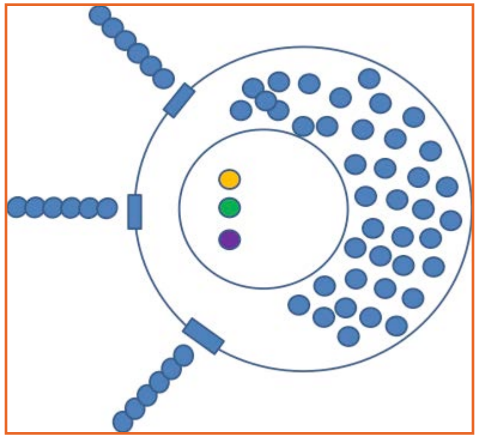


Figure 32. Ms. Yello, Mr. Green and Mrs. Purple Concert

EXERCISE 6.

Vehicles are arriving to a single toll booth that processes cars on both sides of the booth. However, only a single attendant works at the booth and handles vehicles on both sides, one at a time (Figure 33).

- Two separate lines form at the sides of the booth
- The average processing time is 22 to 150 seconds on either side of the booth
- The toll booth employee can only process one (1) vehicle at a time
- Due to a nearby on-ramp, there is a 65-35 split between which side of the booth is selected

1. Build a simulation that captures this system
2. Assume that vehicles on each line arrive with different inter arrival times
3. Assume that there is a delay of 8 to 46 seconds for a vehicle to pull up to the booth after the current vehicle has exited the booth.
4. Capture the assumptions you make
5. How many cars are served in an hour?
6. If Ramp is on the 35% side of the booth
7. If Ramp is on the 65% side of the booth

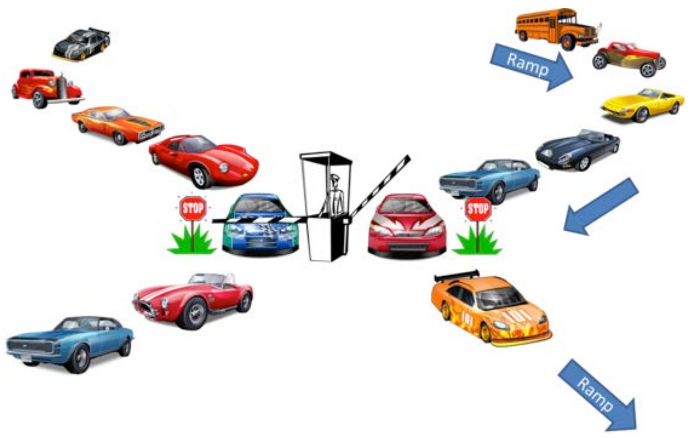


Figure 33. Toll Booth

EXERCISE 7.

People need to be evacuated from point A and taken to point B using buses of capacity of 100 people each (Figure 34).

- The time to travel between A and B ranges between 30 and 45 minutes.
- There are two loading points

1. Build a simulation that captures this situation
2. Assume there are enough buses and these are not reused
3. Capture the assumptions you make
4. How fast can you evacuate 100,000 people? Under what conditions?

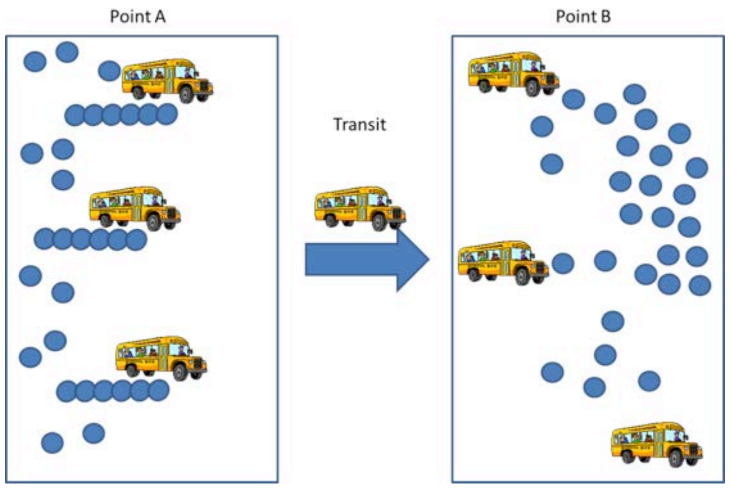


Figure 34. Evacuation from Point A to Point B

EXERCISE 8.

Passengers and flight crews arrive to an airport to take flights to City One and City Two. Passengers and crew are screened by TSA agents aided by X-Ray machines.

- There are six (6) agents and two (2) X-Ray machines. It takes two agents and one machine to screen a passenger.
- After this screening, passengers (only) can be randomly selected for further screening by one TSA agent (no machine).
- Crew and those passengers not randomly selected then move directly to their respective flights.
- A flight requires two crew members and 40 passengers to take off.

1. Draw a conceptual model and build a simulation that captures this system (Doodle 10).
2. Add theoretical distributions and values arbitrarily
3. Capture the assumptions you make
4. How many flights can depart in an eight (8) hour time span?

Doodle 10. Exercise Box – Airport Exercise

chapter 5

ABOUT cloudes

WHAT IS IT?

cloudes is a simulation environment that gives non-expert and experts alike the capability of designing and building discrete-event simulations (DES) using a browser. **cloudes**provides 1) learning in context and 2) modeling and simulating capabilities. Users learn about topics like queuing processes, simulation design and probability distribution in day-to-day scenarios like going to the movies, waiting at the airport, baking cookies or building a house. In addition, **cloudes** can be used by experts to study basic/ intermediate scenarios in areas like manufacturing, healthcare, business or transportation.

WHO IS IT FOR?

cloudes was developed with non-experts in mind; people that would like to know more about simulations but are afraid of its perceived complex nature. Middle and high school students are encouraged to use it in class or for exploring questions -that catch their curiosities. Teachers are encouraged to use **cloudes** for teaching STEM- related courses. College students, small business owners and professionals are also encouraged to use **cloudes** for basic/intermediate simulation design and building needs. No download is required.

WHY cloudes?

Simulations are thought to be for engineering/scientific activities only and applicable to complex scenarios. Simulations are part of our daily lives and as such we think they must be accessible to everyone willing to explore them. They help thinking about problems and more importantly find potential solutions to those problems. **cloudes** makes simulating easy by providing an intuitive environment for users of different training and education levels.

HOW DOES cloudes WORK?

cloudes is a simple to use, web-based, and multi-platform simulation environment for discrete-event simulation creation. You can design your simulations using a web browser (Windows, Mac, Linux, Android, iOS) and run your simulations in the cloud allowing you to run complex models without using powerful computers. No download needed. Just log in, use existing templates, use cloudes public simulations, or build your own.

INTER ARRIVAL/SERVICE (I/S) COUNTERS

Counters make it easy to capture data and use it in our environment. No need for chronographs and sheets of papers. Select a counter, collect your data, and use your data or the probability distribution of that data in arrivals or processes. cloudes has two counters: Single (one arrival or service) and Single I/S (one arrival and one service). The data is shown as it is captured but the difference between the data points (interarrival/service time) is calculated in the server.
Note: Currently (Sep. 11, 2014), the counter data is not shareable. It will be in near future update.

SIMULATION RESULTS

Simulation results are stored in the cloud or you can download them for further analysis. You can conduct unlimited runs and get summary results or conduct replications and export the full result.

WHAT IS UNDER THE HOOD (BONNET)?

Nowadays, taking an endeavor of this magnitude requires standing on the great work others have done. Here you will find a list and link to those efforts (both open source and commercial) at the moment of writing. For a description and link to their licenses, please go to: http://test1.cloudes.me/licenses

DESMO-J | http://desmoj.sourceforge.net/home.html
DESMO-J provides the simulation engine of ClouDES

Meemoo | http://meemoo.org/
Meemoo provides the capability for the simulation blocks to be connected.

jQuery | https://jquery.org/
jQuery provides JavaScript library for DOM manipulations, animations and AJAX handling (jQuery 1.9.1); user interface interactions, effects and widgets (jQuery UI 1.10.3) and HTML5-based user interface system designed for mobile devices (jQuery Mobile 1.10.3).

html2canvas | http://html2canvas.hertzen.com/
html2canvas Saves view of HTML elements as image

Backbone.js | http://backbonejs.org/
Backbone provides a JavaScript library for developing MVC-like applications.

Underscore.JS | http://underscorejs.org/
Underscore provides a JavaScript library with collection of helper functions for common tasks

Commons Math | http://commons.apache.org/proper/commons-math/
Commons Math provides the statistics library in order to calculate theoretical distributions.

Apache Tomcat | http://tomcat.apache.org/
Apache Tomcat provides the web server for Java applications.

PostgreSQL | http://www.postgresql.org/
PostgrSQL provides database management services.

Google Charts | https://developers.google.com/chart/
Google Charts provide a gallery of interactive charts and data tools.

Jaspersoft Studio | https://community.jaspersoft.com/project/jasperreports-library
Jaspersoft Studio provides the reporting engine.

SmartXLS | http://www.smartxls.com/indexj.htm
SmartXLS provides Excel spreadsheets generation.

TROUBLESHOOTING

The simpler the interface of a software the more complex the software tends to be inside. If cloudes is behaving erratically, it could be for a varied of reasons: the system has been updated, not properly terminated browser session, not properly terminated cloudes session, known bugs, and unknown (to us) bugs.

If you encounter erratic behavior, please send us a note to note@ClouDES.me or to mscube@odu.edu.

Please provide us with a screenshot and/or a description of the problem.

We address known bugs promptly and update the system ASAP. In the meantime, please try these and see if the issue is resolved:

1. CLOUDES IS BEHAVING ERRATICALLY:

a. Use the browser Reload option.
b. Clear your cache.

2. A SIMULATION IS BEHAVING ERRATICALLY:

a. Check that environment parameters are properly filled.
b. Check that blocks' properties are properly filled.
c. Create the simulation again from a sample or from scratch.

If the issue is resolved, delete the old simulation.

3. A SIMULATION IS NOT BUILDING:

a. Check the message box for error reporting
b. Make sure the simulation has a name.
c. Make sure that all needed block properties are filled.

4. A SIMULATION IS GENERATING ERRONEOUS RESULTS:

a. Check simulation parameters and assigned values.
b. Alter the design of the simulation if possible. This could be caused by an unknown bug that affects a block combination we are not aware of.

We will be reporting on system updates, new features and addressing known bugs in an upcoming blog.

Thank you for using cloudes. We hope that its goal of making simulations accessible to everyone is achieved every time it is useful to you.

NEW FEATURES

Please do send us recommendations for new futures. We will evaluate them, compare them to the ones we have planned, and assign them a priority. Your feedback is valuable to us.

cloudes is a platform for our research at VMASC. Insights gained from our research activities will be implemented in cloudes in areas like: Conceptual Modeling, Interoperation, Multi-paradigm modeling, Semi-automatic Verification, Web-based Simulation, and Symbiotic Simulations to mention some.

If you would like to contribute to cloudes by building new features, please let us know at note@ClouDES.me.

cloudes TEAM

cloudes is an initiative of the Virginia Modeling, Analysis and Simulation Center (VMASC), at Old Dominion University (ODU), to support and engage users like you in computer literacy, STEM activities and promote M&S entrepreneurship.

The cloudes VMASC team is (in alphabetical order):

- Anthony Barraco (Lead Developer)
- Saikou Y. Diallo, Ph.D.
- Ross J. Gore, Ph.D.
- Hamdi Kavak
- Christopher Lynch
- Jose J. Padilla, Ph.D. (Project Lead)

THE VIRGINIA MODELING, ANALYSIS AND SIMULATION CENTER

The Virginia Modeling, Analysis and Simulation Center is one of the world's leading research centers for computer modeling, simulation, and visualization. The mission of the Center is to conduct collaborative MS&V research and development, provide expertise to government agencies and industry, and to promote Old Dominion University, Hampton Roads and Virginia as a center of MS&V activities. Working with more than one hundred industry, government, and academic members, VMASC furthers the development and applications of modeling, simulation and visualization as enterprise decision-making tools to promote economic, business, and academic development.

OLD DOMINION UNIVERSITY

Old Dominion University is a state-assisted institution and one of only four Virginia schools in the Carnegie Research Universities (high research activity) category. The University offers a complete range of Modeling & Simulation degree options from Bachelor's to Ph.D.

cloudes

VMASC
OLD DOMINION UNIVERSITY

1030 University Boulevard
Suffolk, Virginia, USA

www.vmasc.odu.edu