



Airport Landside Modeling (Computer Applications and Modeling)

Dr. Antonio A. Trani

**Professor of Civil and Environmental Engineering
Virginia Polytechnic Institute and State University**

CEE 4674 - Airport Planning and Design



Material Presented in this Section

- Brief description of airport terminal simulation languages
 - Advantages and weaknesses
 - Basic constructs
 - Example of VPI Airport Terminal Simulation Library (EXTEND)
 - Example APM simulation model (APMSIM) developed at Virginia Tech
- Future directions and impacts



Purpose of the Discussion

- Until now you understand the basics of probability, continuous, and discrete event simulation and modeling using generic mathematical packages (such as Matlab) or spreadsheets
- Dedicated simulation languages keep track of many of the book keeping activities required in the simulation model
 - you can concentrate in the model with minimal implementation burden
 - productivity and model development are enhanced
- Simulation tools provide rich GUI constructs to enhance understanding from a decision maker viewpoint



Simulation Languages

Discrete Event Simulation Languages

SIMULA, STELLA, SIMSCRIPT II.5, MODSIM,
SLAM III, GPSS-H/GPSS-PC, Arena/SIMAN, EXTEND

Continuous Simulation Languages

ACSL, Simulink, STELLA, SIMSCRIPT II.5, MODSIM,
SLAM III, GPSS-H/GPSS-PC, Arena/SIMAN, EXTEND

- Many of these languages provide a hybrid modeling capability (i.e., discrete and continuous paradigms in the same model)



A Typical Simulation Language

- While all languages provide very similar constructs to build models, some have more complex Graphic User Interfaces (GUI) than others
- The basic building blocks of a simulation model are maintained.
- Building blocks of discrete simulation languages:
 - Entities, Attributes, Resources, Queues, Accumulators, and Events
- Building blocks of continuous simulation languages:
 - Integrator (reservoir, tank), generator, delay structures, function blocks, rate variable blocks, etc.

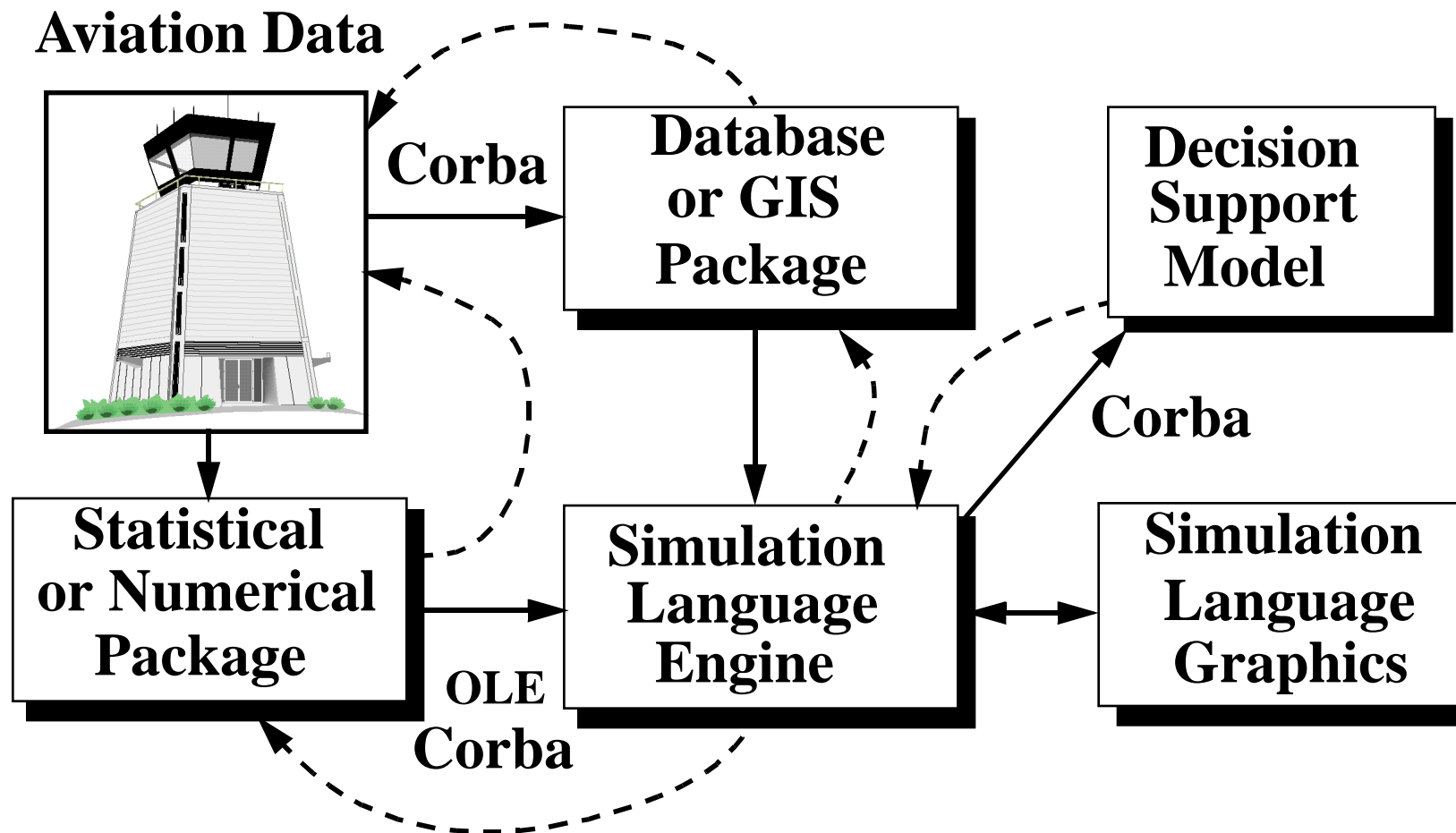


A Typical Simulation Language (cont.)

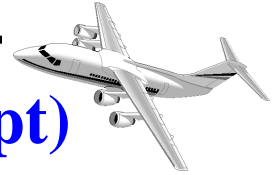
- Modern (3rd generation simulation languages) provide rich GUI interfaces to construct and prototype models (EXTEND, Stella, and Arena are good examples of these)
- Object-oriented data abstraction is the norm
- All simulation languages have connectivity capabilities with other packages (i.e., statistical, spreadsheets, word processors, programming)
 - ODBC/Corba support for database connectivity
 - OLE support for Windows programming compatibility
 - Statistical package support for regression and input/output analysis



Example of a Simulation Language Use and Connectivity



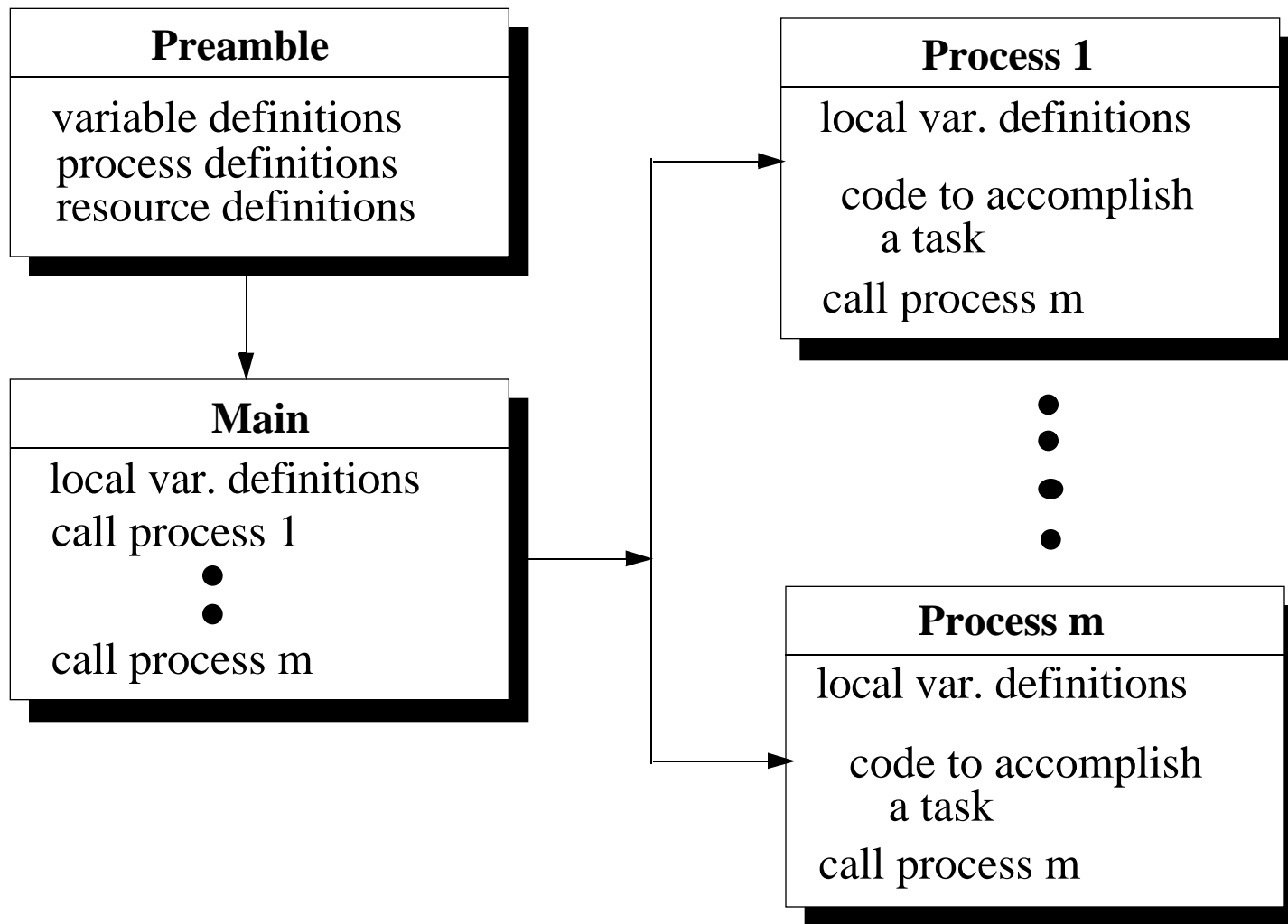
Sample 2nd Generation Sim. Language (Simscrip)



- Developed and marketed by CACI (US)
- Provides English-type constructs
- C-type model construction
 - **Preamble file** (contains global variables, resource definitions, event types, tally definitions, etc.)
 - **Main file** (first routine to be executed: calls others in the program)
- C-language portability (platform independence)
- Heavily used in military and **aviation** modeling in the 1970's and 1980's in US
- Stable compiler (few modifications in the past five years)



Typical Organization of a Simscript II Model



Sample 2nd Generation Sim. Language (Simscrip)

- Graphic capabilities were added in the 1980's (Simgraphics)
- One of the first simulation languages used in distributed simulation mode (i.e., running processes in different computers)
- SIMMOD - the airspace and airfield simulation model developed by the FAA was developed in Simscrip II.5 in the late 1970's
- Today, **MODSIM** is replacing Simscrip II.5 and adding full Object-Oriented Programming paradigm capabilities (RAMS - another airspace simulation model by Eurocontrol was developed in using MODSIM)

Sample 3rd Generation Sim. Language (EXTEND)



- Developed by Bod Diamond (1987) and distributed by Imagine That, Inc.
- Provides a rich GUI to develop models using blocks
- Access to C-like block code (Modl language)
 - **Blocks are classified in terms of libraries**
 - Drag-and-connect approach
 - Hierarchy blocks (multiple blocks working towards a common goal)
- Some portability (WIN and Mac versions available)
- Used in manufacturing, transportation and control engineering

Sample 3rd Generation Sim. Language (EXTEND)



Extend Libraries

Generic Library: has blocks that perform basic functions as math, decision handling and input/output

Discrete-Event Library: contains blocks for creating discrete-event simulation models (including activity delays, resources, processes, etc.)

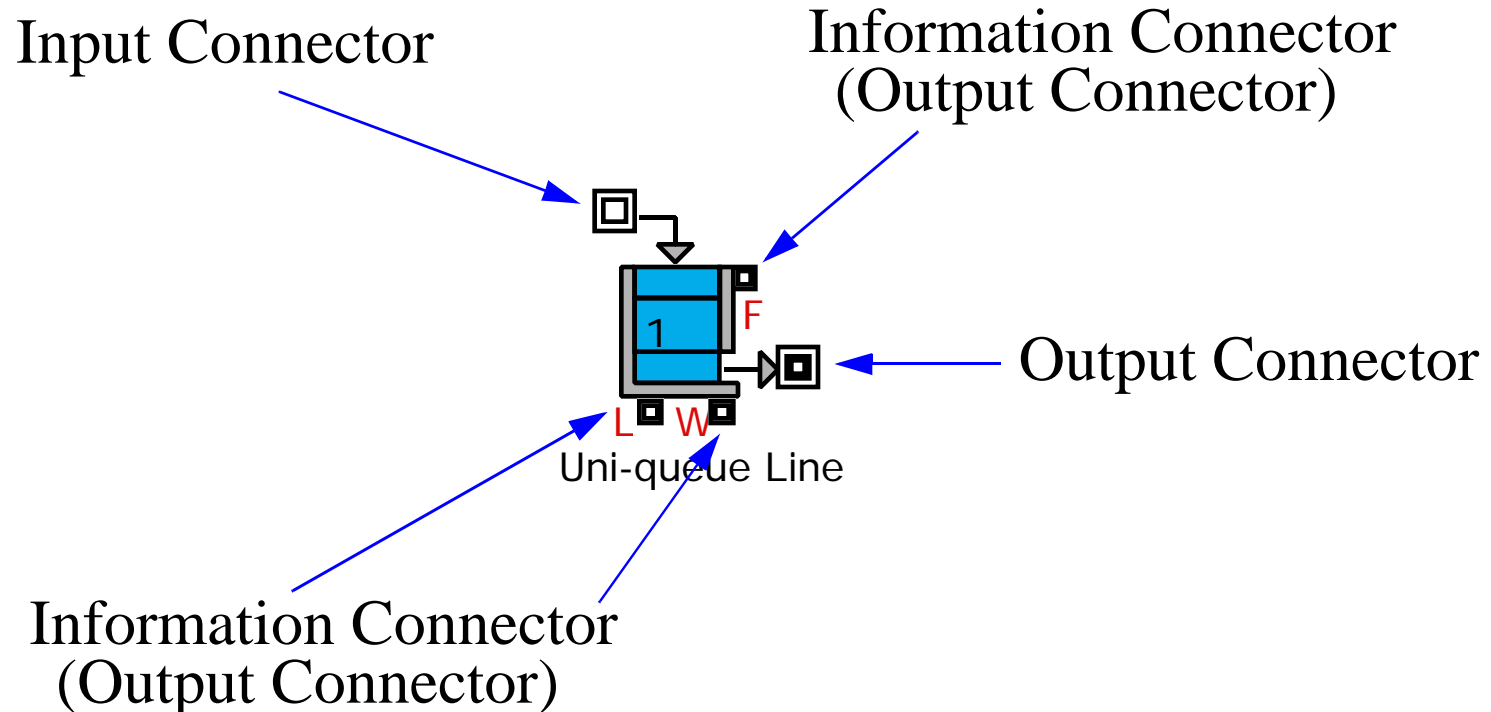
Plotter Library: contains blocks for plotting the results in any type of simulation model

Animation Library: holds blocks that are used for animating hierarchical blocks



Taxonomy of a Block

Blocks are complex structures in Extend that allow quick modeling of complex processes. Shown below is a typical block showing its components.





Taxonomy of a Block (cont.)

Each block in Extend is **extensible** and **modifiable** to the source code level (Mod1 - the language used in Extend - blocks is an extension of the C language).

```
procedure getArrays()
{
    if (rCount != sysGlobalint2)
    {
        if(useString)
            getPassedArray(sysGlobal5, itemArrayA);
        if (costing)
            getPassedArray(sysGlobal9, itemArrayC);
        rCount = sysGlobalint2;
    }
}
```

Example of Security Check Point Queue in Extend



This example replicates the seaport example discussed in handout 7 (queuing models). The idea is to compare the simulation results with those obtained with the analytic queueing model.

The purpose of the example is to show you how simple a multiple server problem is constructed and solved using various simulation packages.

Example : Level of Service at Security Checkpoints



The airport shown in the next figures has two security checkpoints for all passengers boarding aircraft. Each security check point has two x-ray machines. A survey reveals that on the average a passenger takes 45 seconds to go through the system (negative exponential distribution service time).

The **arrival rate** is known to be random (this equates to a Poisson distribution) with a mean arrival rate of one passenger every 25 seconds.

In the design year (2010) the demand for services is expected to grow by 60% compared to that today.

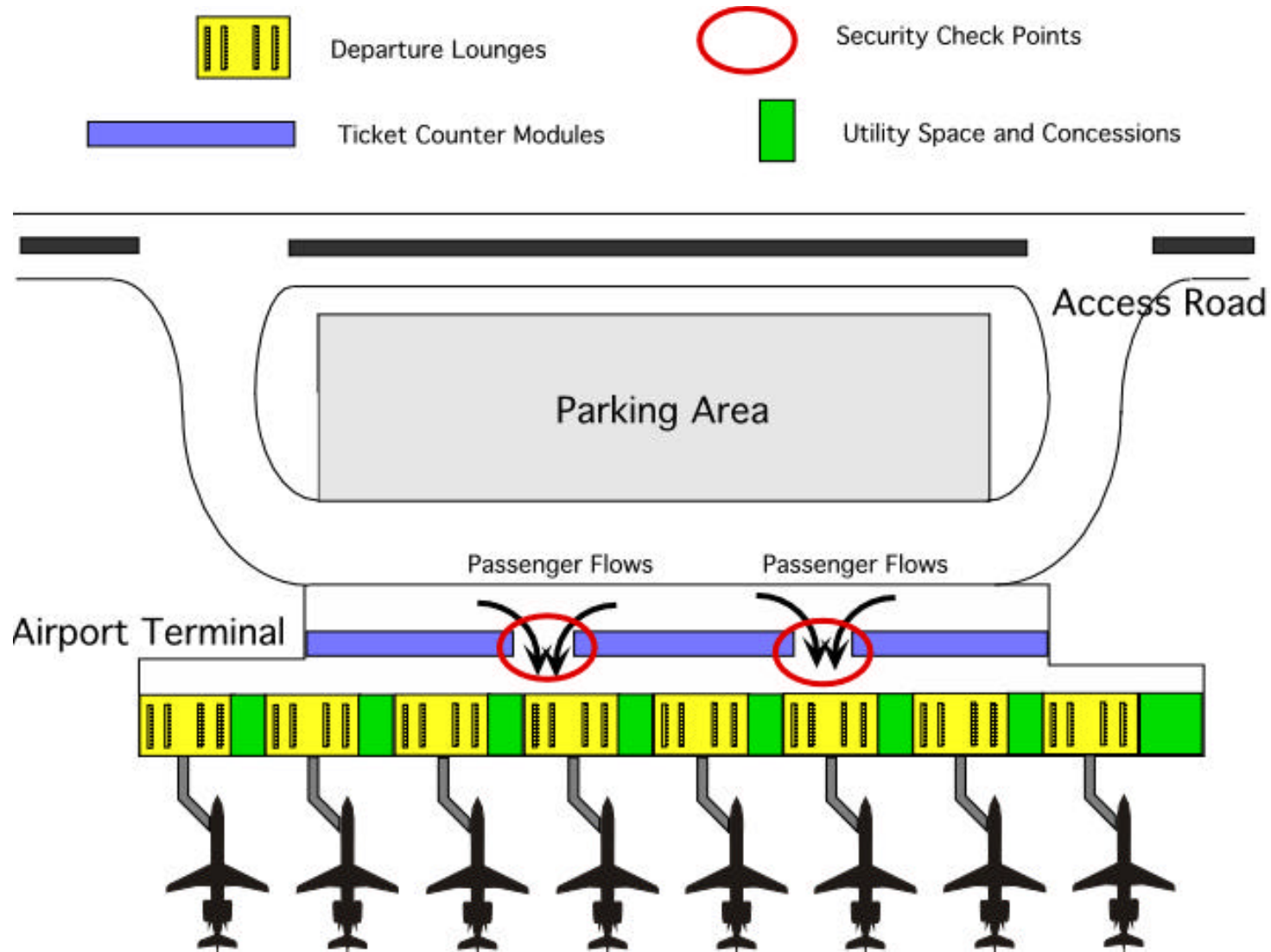


Relevant Operational Questions

- a) What is the current utilization of the queueing system (i.e., two x-ray machines)?
- b) What should be the number of x-ray machines for the design year of this terminal (year 2010) if the maximum tolerable **waiting time in the queue** is 2 minutes?
- c) What is the expected number of passengers at the checkpoint area on a typical day in the design year (year 2010)?
- d) What is the new utilization of the future facility?
- e) What is the probability that more than 4 passengers wait for service in the design year?

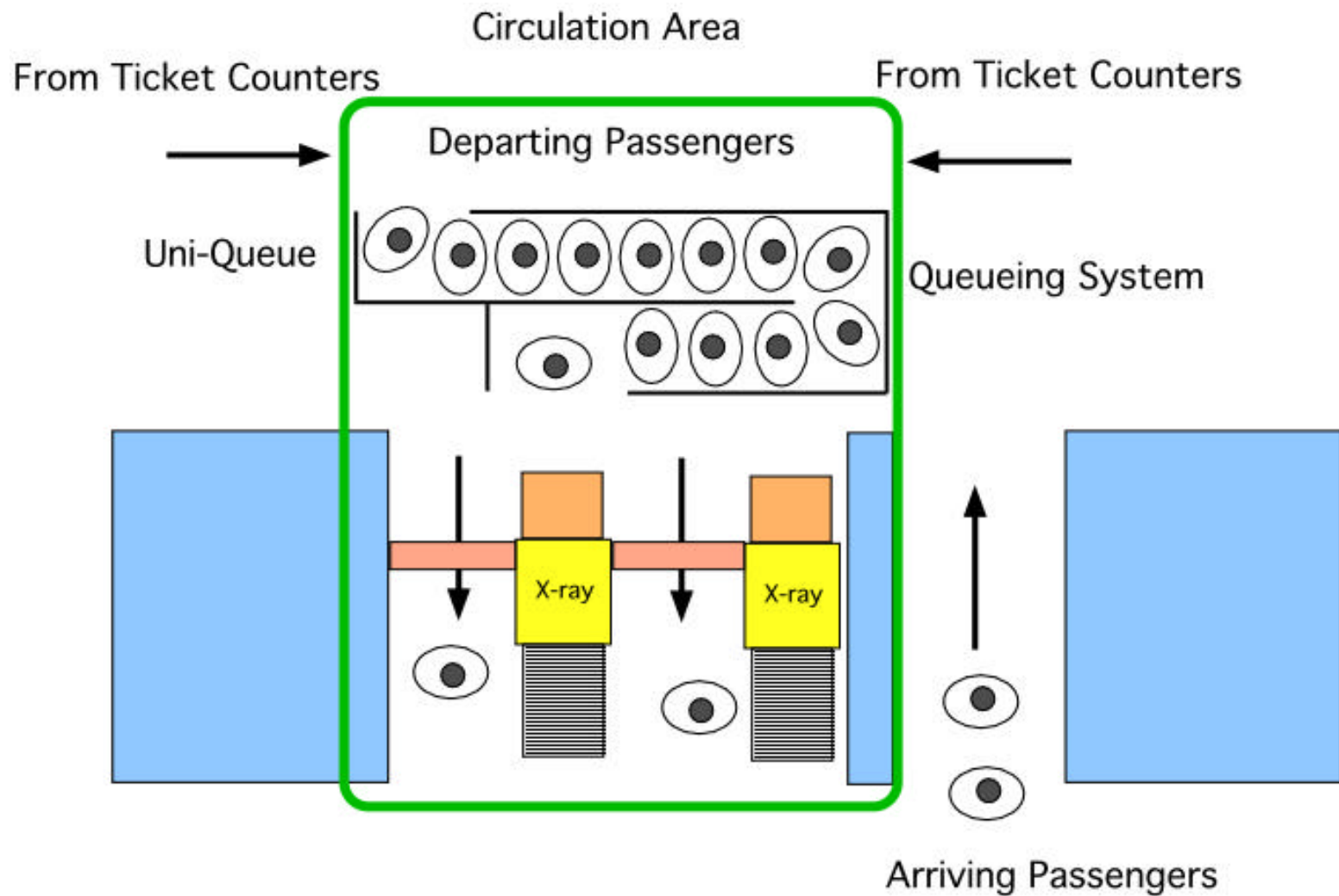


Airport Terminal Layout





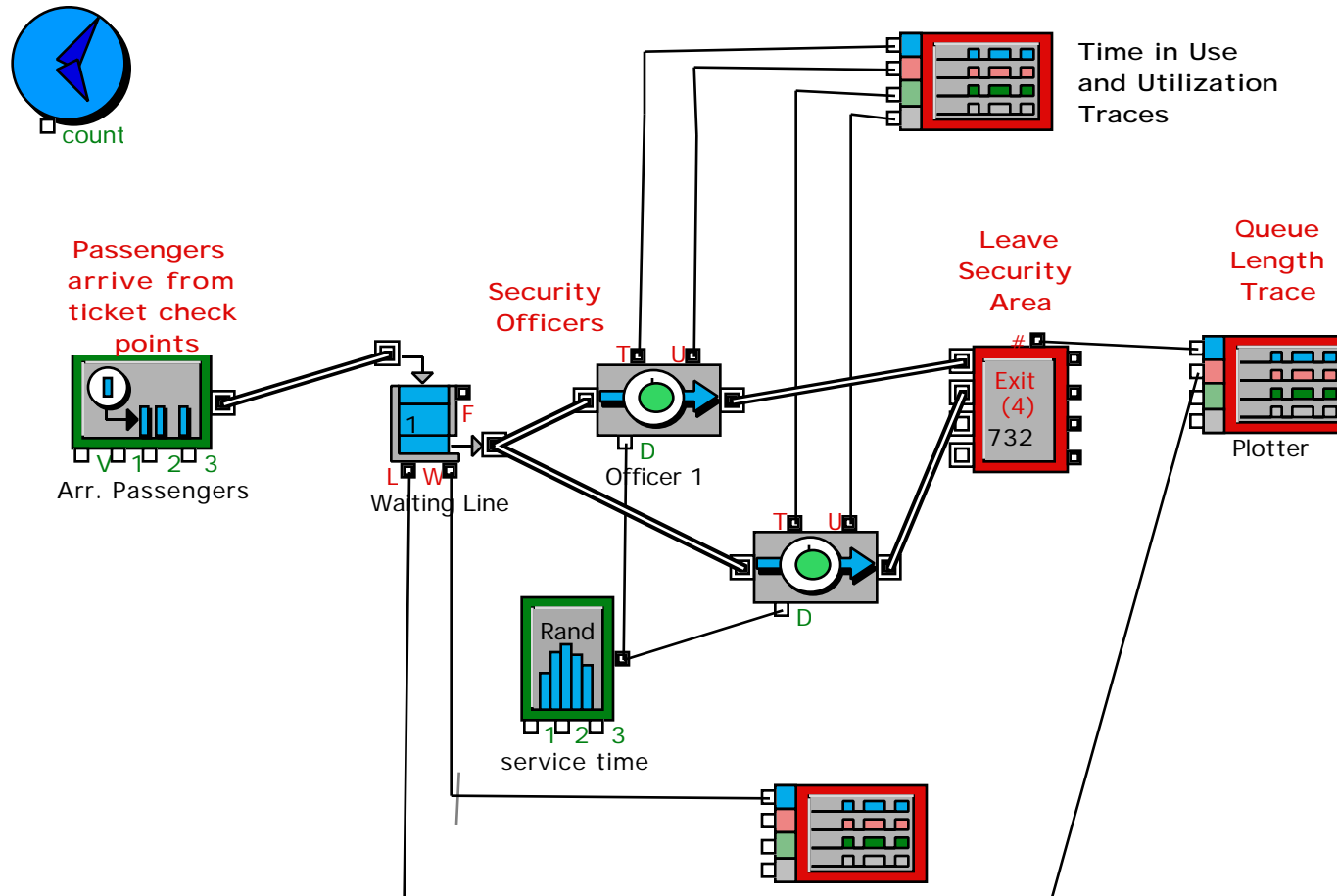
Security Check Point Layout





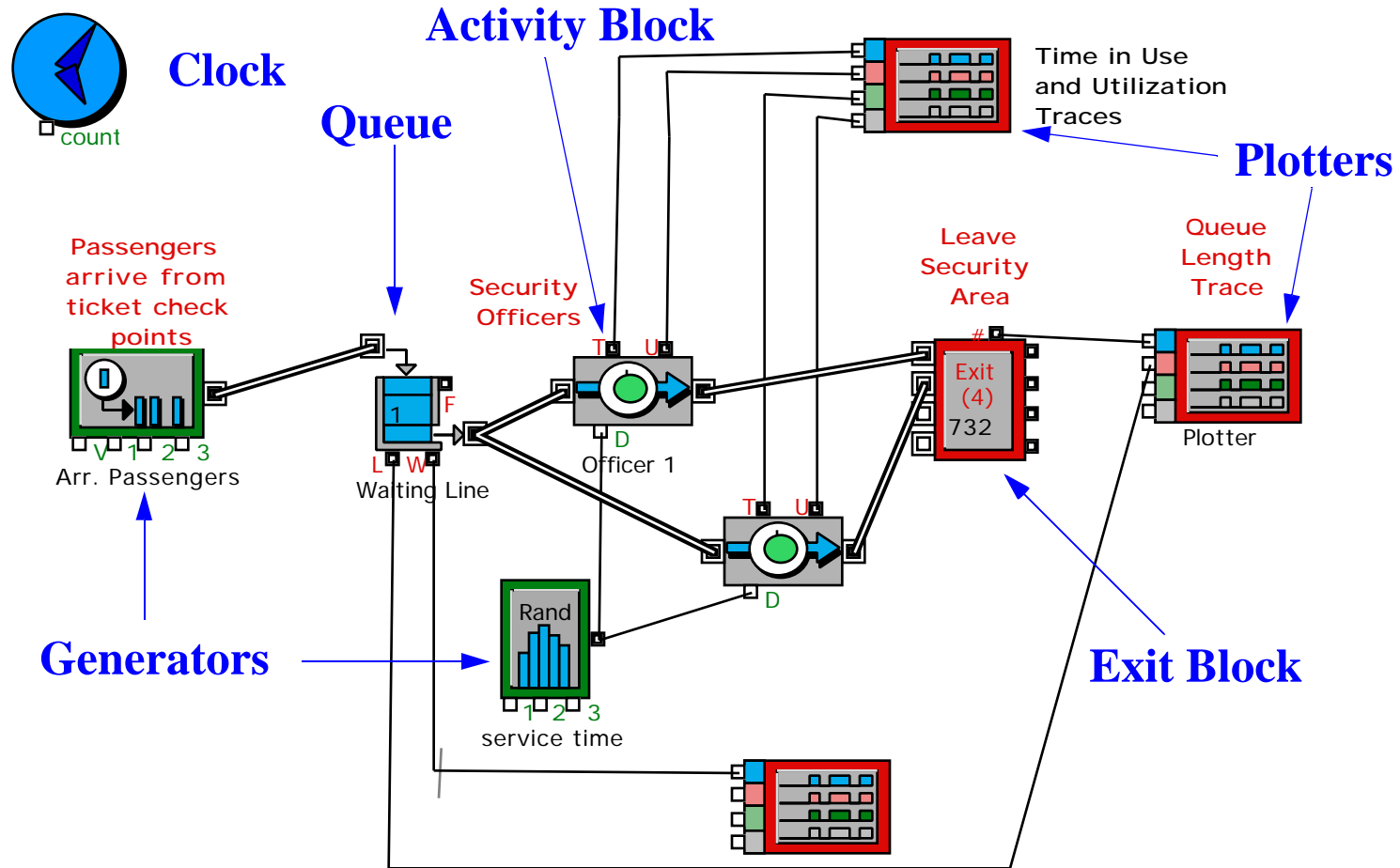
Security Check Point Model (Baseline)

Representation of the system modeled in Extend (s=2)



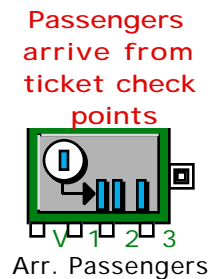


Extend Model with Block Labels

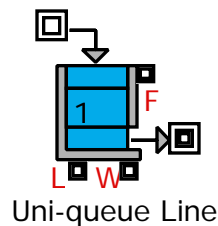




Explanation of Extend Blocks Used



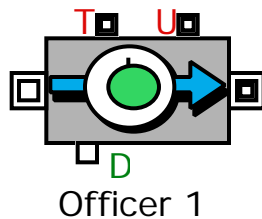
Discrete Block Generator - generates discrete entities in conjunction with discrete event models. This block has 18 different Probability Density Functions (PDF) to choose from, including an empirical distribution format to enter real data without a PDF fit.



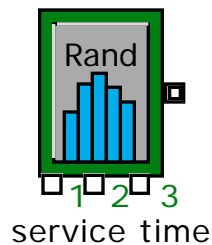
Queue FIFO - this structure keeps track of physical queues where the first entity arriving is the first one to be processed. Parameters L and W represent the queue length and waiting time, respectively. Output F is a binary function that takes values of 1 when the queue is full (zero otherwise)



Explanation of Extend Blocks Used



Activity Delay - holds an entity for a specified amount of time. Input parameter is D , the delay inside the activity block. Output parameters are T and U which represent the time in use and the utilization of the block, respectively.

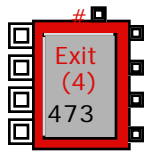


Random Number - generates a random number according to a pre-specified distribution. Input parameters are 1, 2, and 3 which represent 3 input arguments used to define user-defined distributions. The single output parameter is a random number. This block is typically used in conjunction with activity delays.



Explanation of Extend Blocks Used

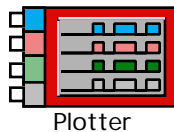
Leave
Security
Area



simulation.

Exit Block (4) - destroys up to 4 items from further consideration in the simulation. The total number of entities absorbed by the block are reported. The output # connector reports the number of entities exiting the

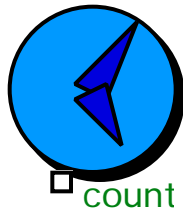
Queue
Length
Trace



Plotter, Discrete Event - This block plots up to 4 parameters in the same figure. Note four input parameters in the left hand side of the block.



Explanation of Extend Blocks Used



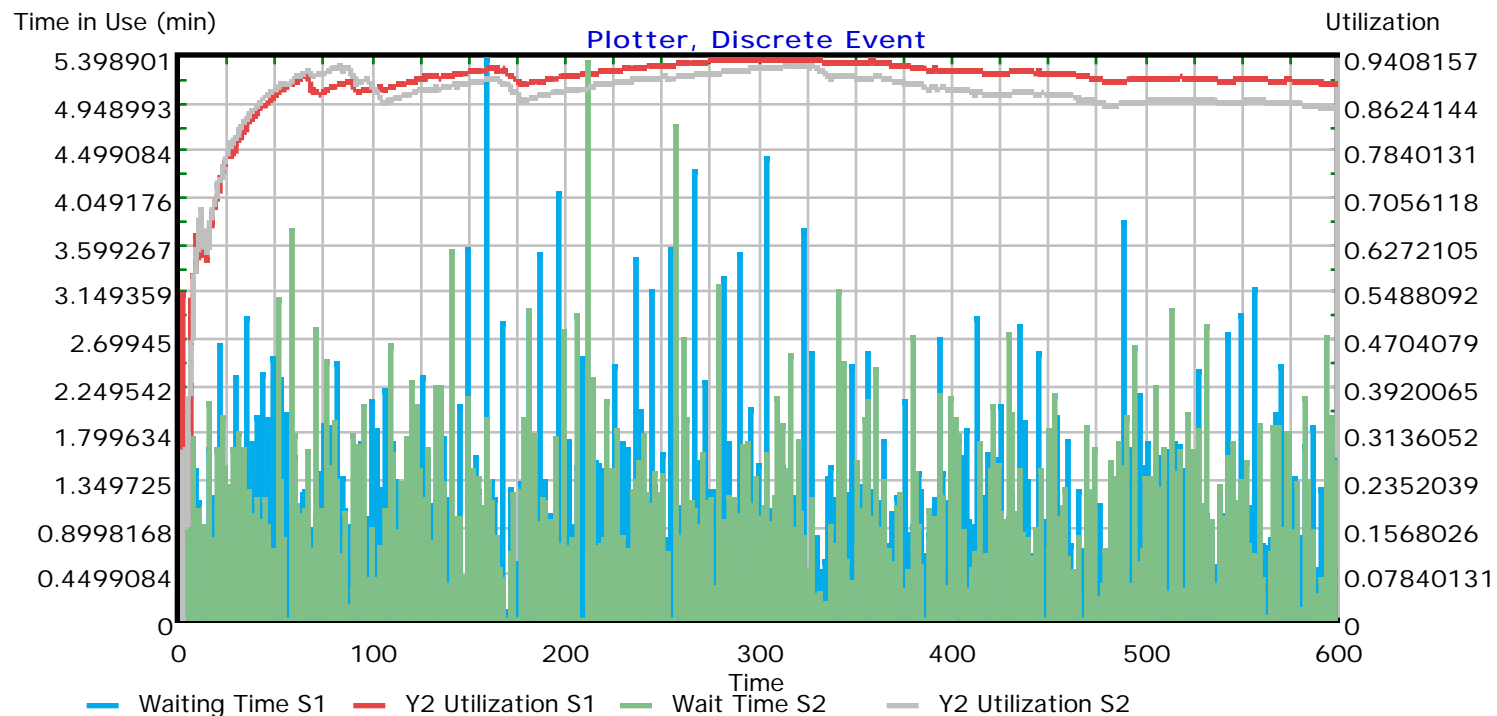
Executive Block - controls the duration of the simulation. This can be done either by count (i.e., a prespecified number of units or by time duration).

Note: this block has to be present in all discrete simulation models in Extend. Its location must be at the left most position in the simulation model (a necessary convention in Extend).



Sample Results for Two-Server System (Baseline Conditions)

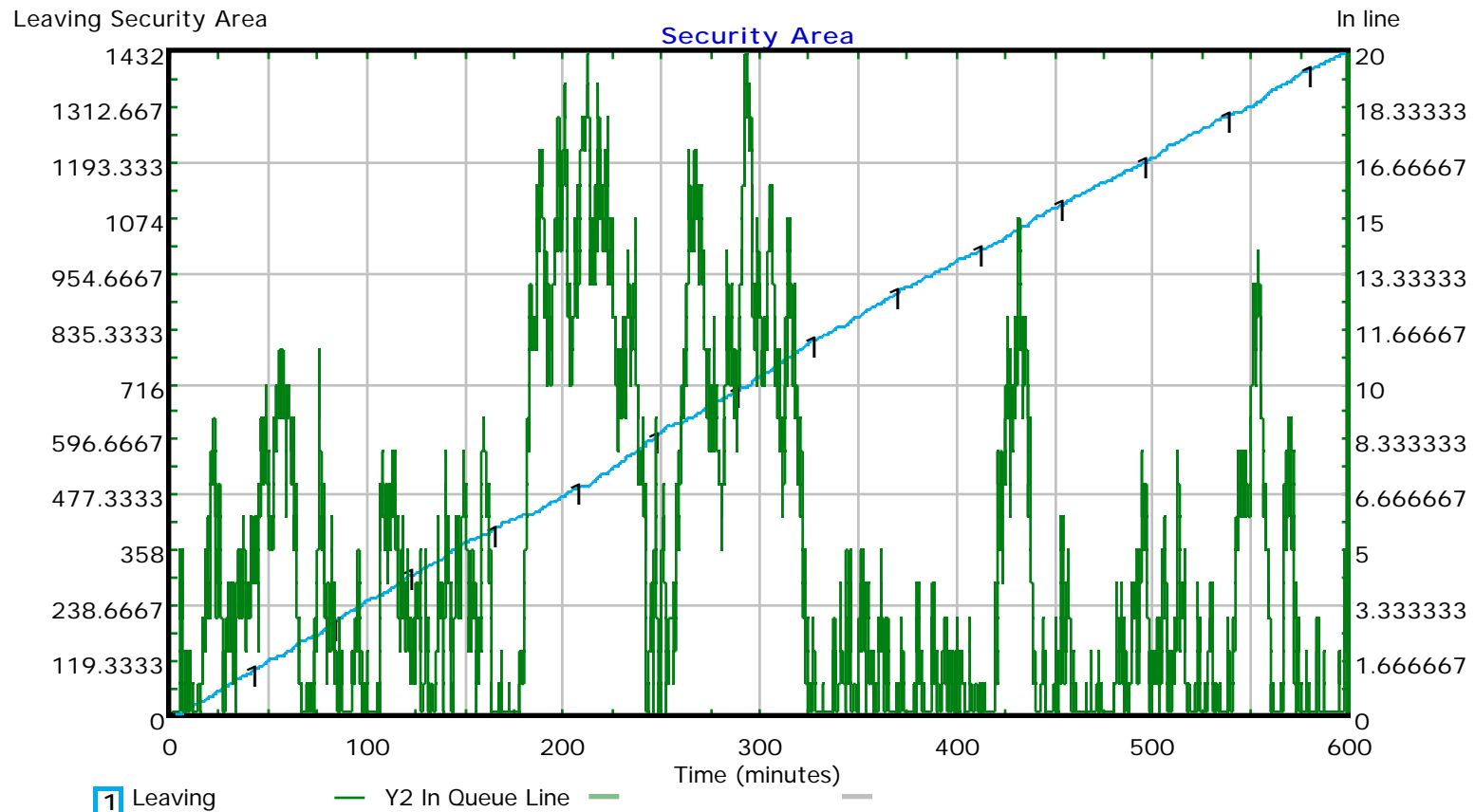
The results below show the utilization and waiting time instances for the security checkpoint area





Uni-Queue Length Trace

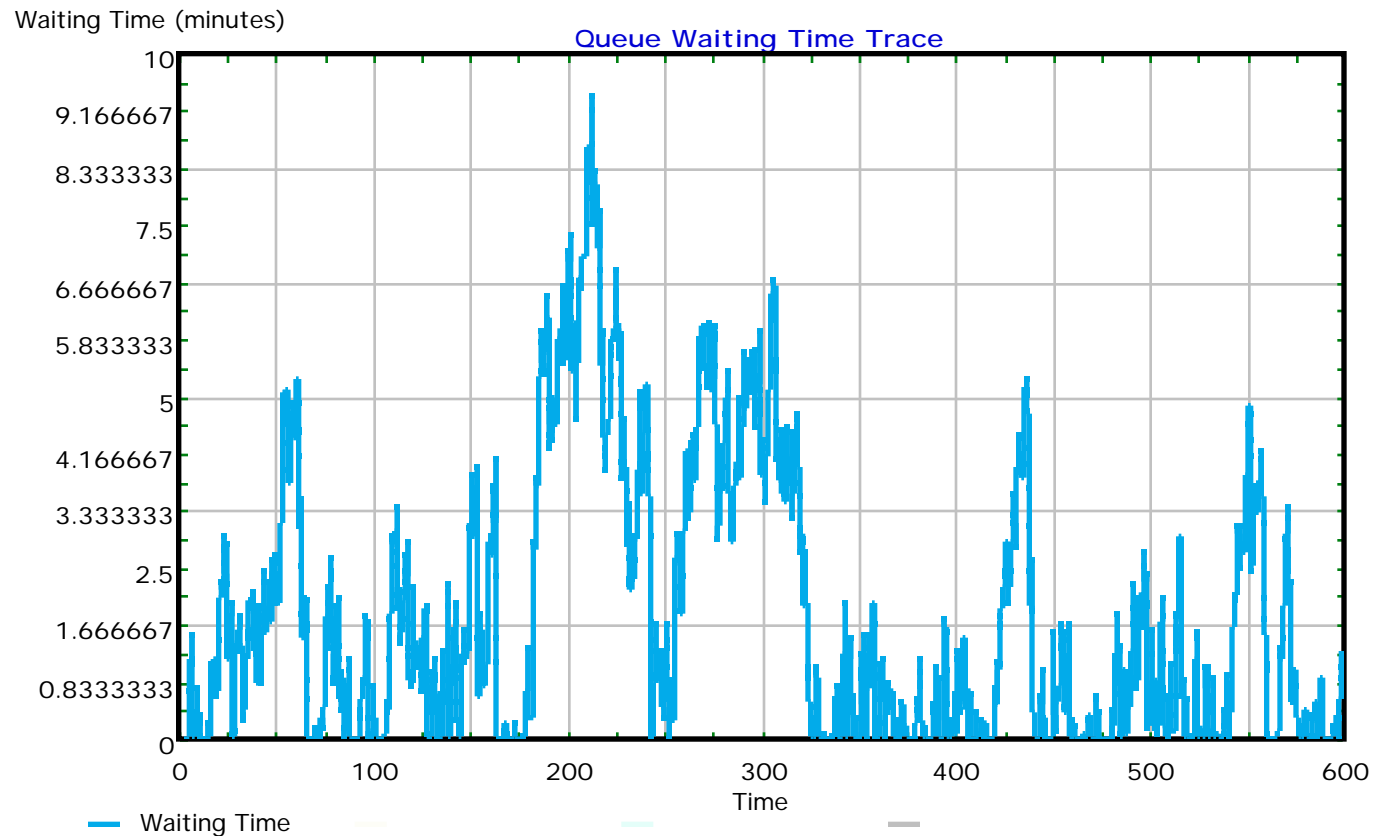
The following diagram depicts the uni-queue length





Uni-Queue Waiting Time Trace

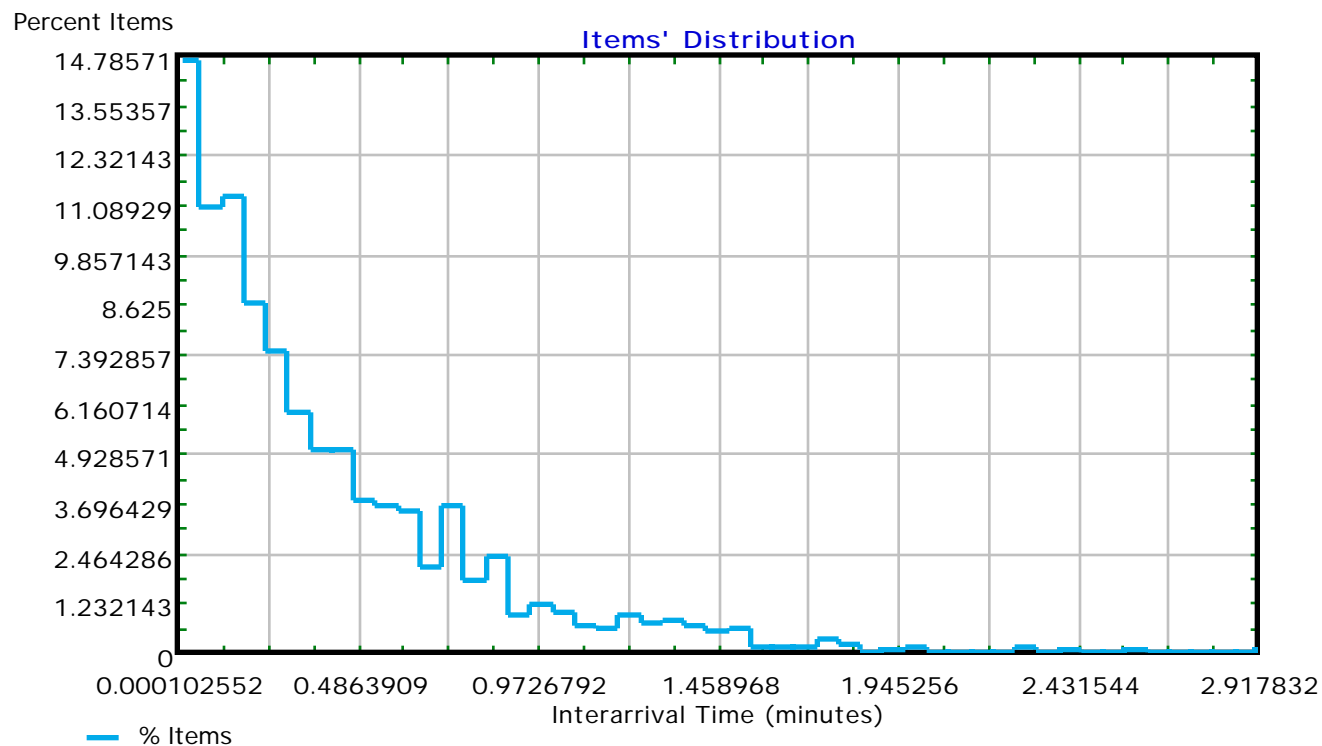
The following diagram depicts the waiting times vs time





Distribution of Interarrival Times

The following plot of interarrival times was generated using Extend in the security check point model





Discussion of Results (Baseline Year)

The following table shows typical results for the baseline year and compares them with those of the analytic model

Table 1. Results for Security Check Point System.

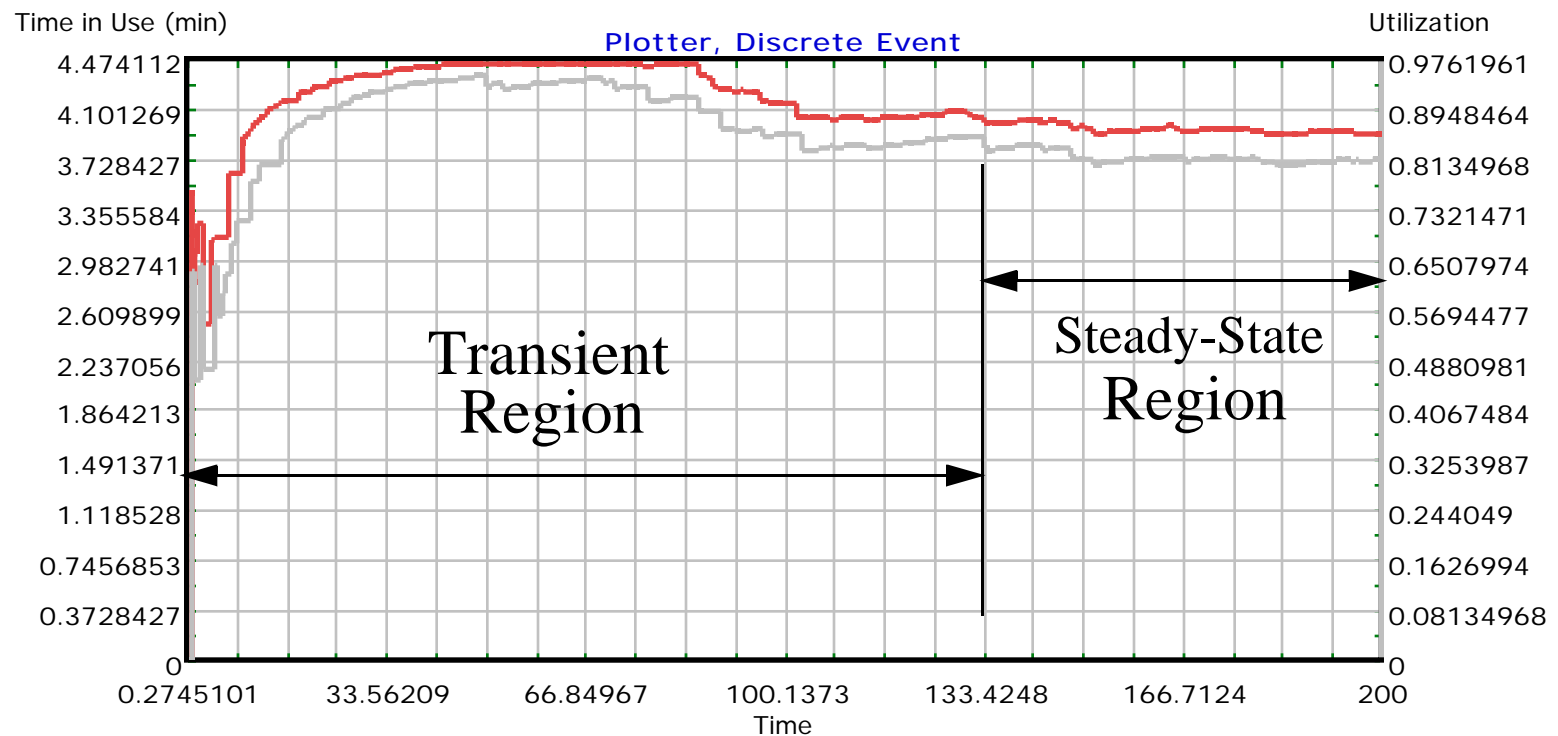
Parameter	Analytic Queueing Model ($s = 2$)	Simulation Model (2 servers)
Utilization ()	0.900	0.916 / 0.886 ^a
Expected Queue Length (L_q) - per	7.60	6.96
Exp. Waiting Time in Queue (W_q) - min	3.20	2.85

a. Two values are reported because Extend keeps independent statistics for each server.



Simulation Length and Stability of Results

Simulation results require careful interpretation to guarantee stable solutions as exemplified below





Part (a) Baseline Utilization Results

- From the previous graphs is evident that the utilization of each server in the baseline year is around 90%
- This result is consistent with that obtained with the analytic queueing model (i.e., multi-server and infinite source)

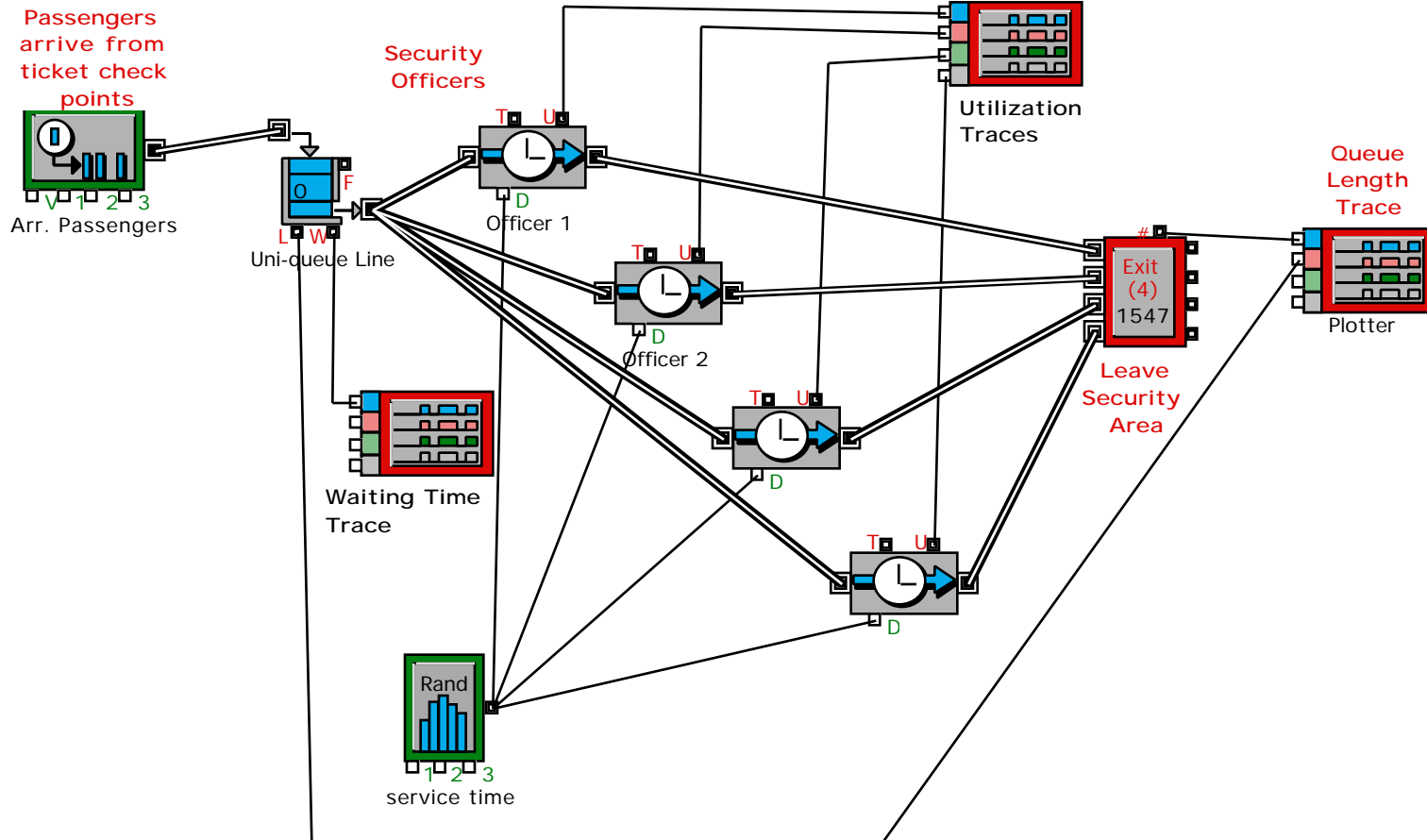


Part (b) Horizon Year Operation

- Recalling from the analytic model that 4 x-ray machines were needed to provide levels of service below 2 minutes (in the queue)
- The corresponding Extend model and results are illustrated in the following pages
- Clearly the results of the simulation model are in close agreement with those of the analytic model presented in the queueing models section
- As a matter of simulation practice you should always check the results of your simulation models with close form solutions before developing large and more complex models



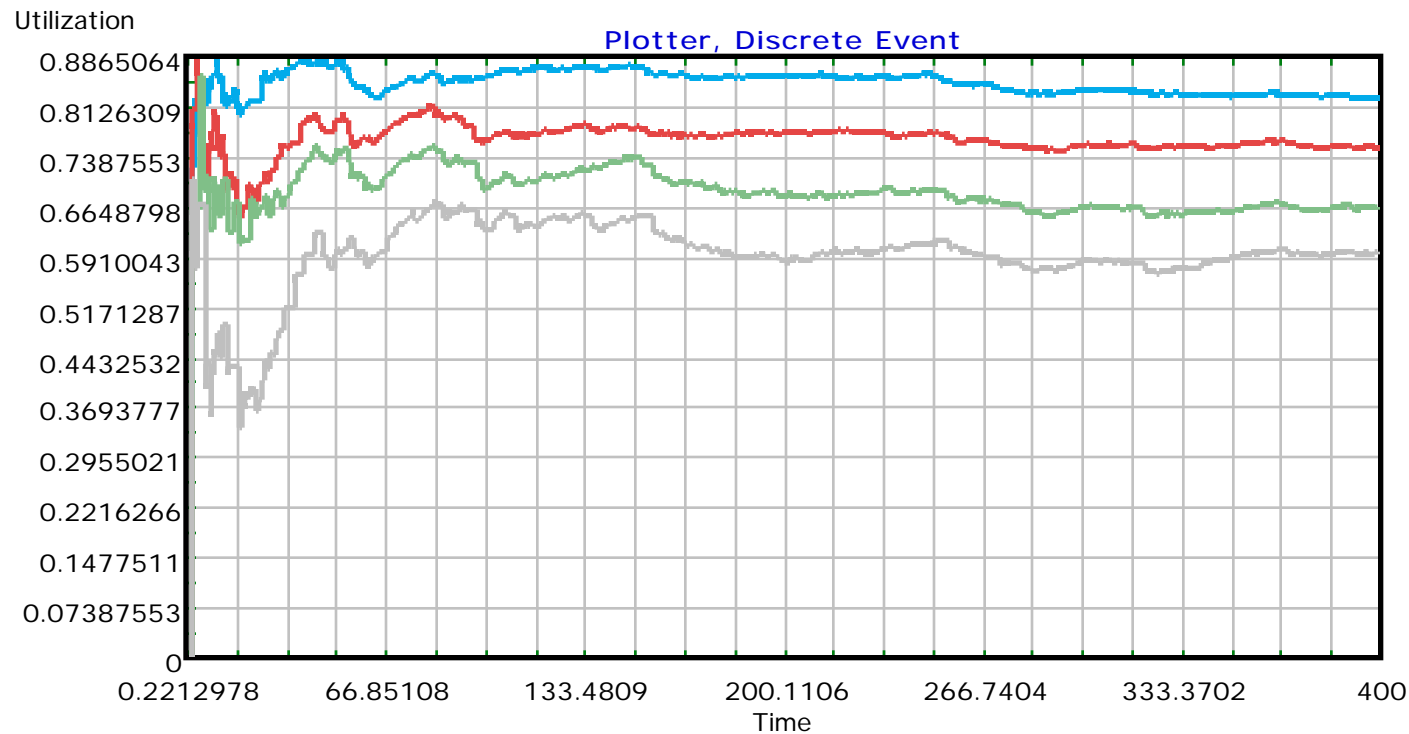
Horizon Year Extend Model (4 servers)





Horizon Year Model Utilization

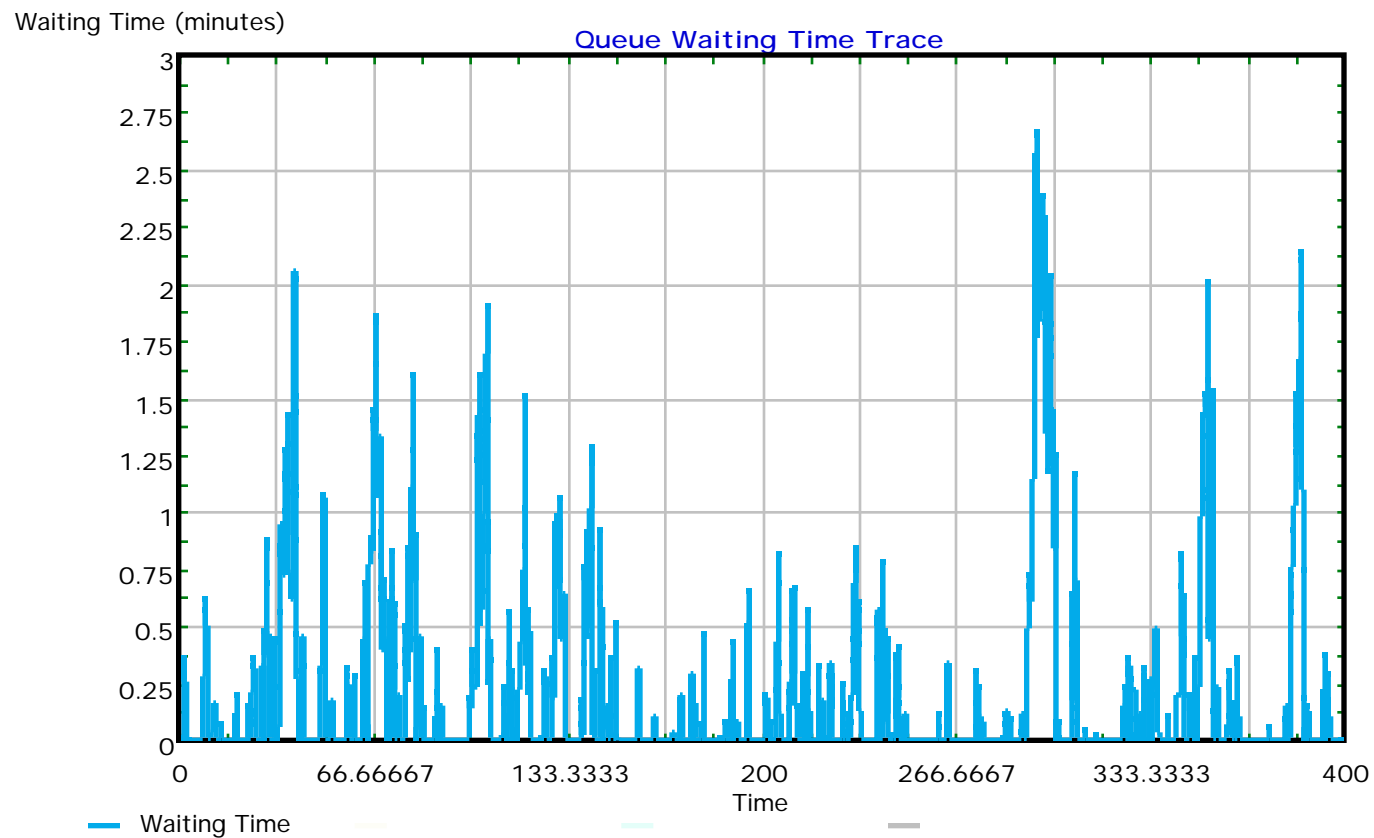
The average utilization is reduced with 4 servers and 60% increase in the demand functions (400 minute simulation)





Horizon Year Results (W_q)

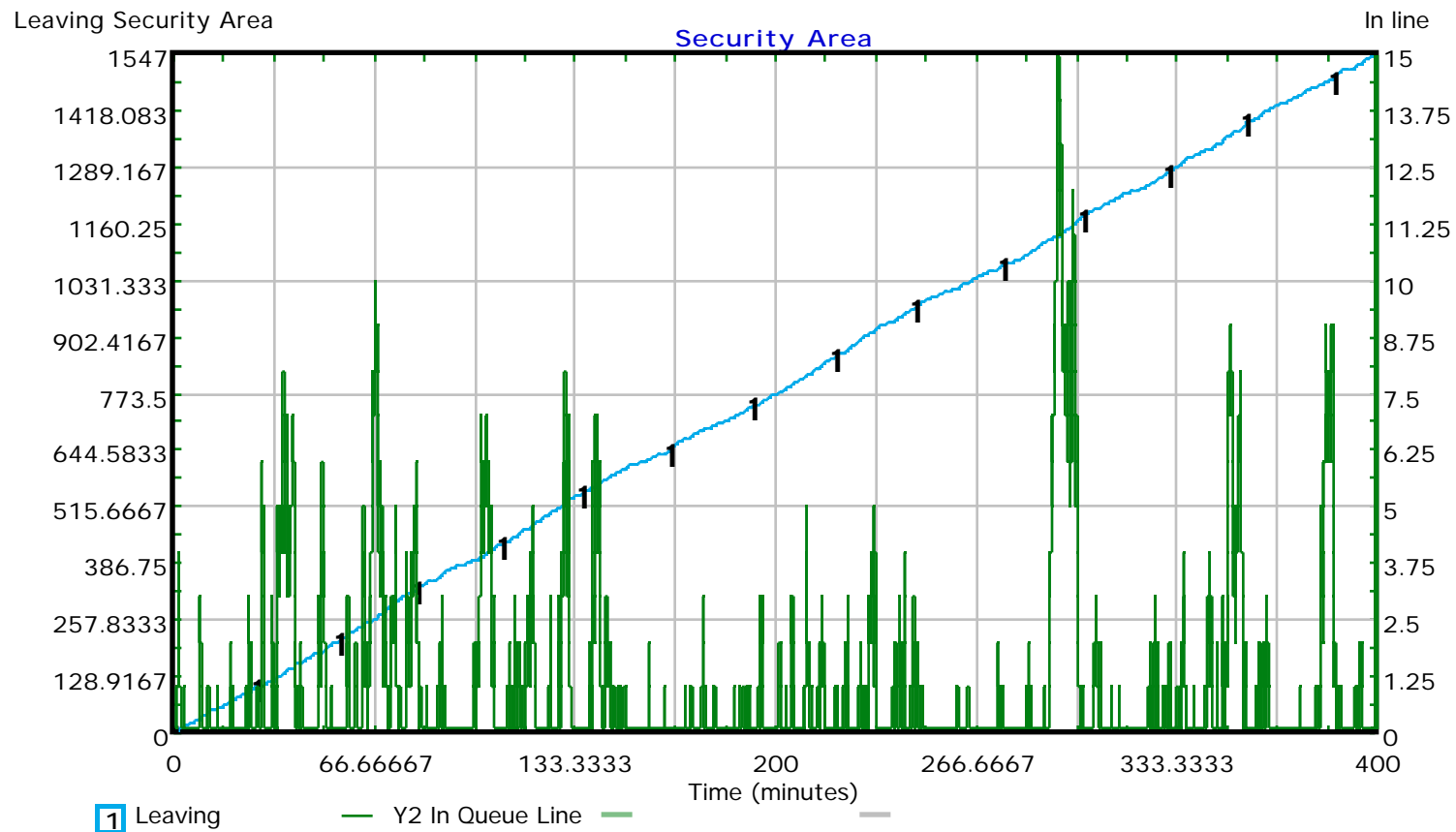
The following plot shows the behavior of the waiting time function during a 400 minute simulation





Horizon Year Queueing Characteristics

Four servers are capable of coping with 60% increase in demand (400 minute simulation)





Discussion of Results (Horizon Year)

The following table shows typical results for the horizon year and compares them with those of the analytic model

Table 2. Results for Security Check Point System.

Parameter	Analytic Queueing Model (4 servers)	Horizon Year (4 servers)
Utilization ()	0.72	0.82 / 0.75 / 0.66 / 0.60 ^a average = 0.71
Expected Queue Length (L_q) - per	1.18	0.97
Exp. Waiting Time in Queue (W_q) - min	0.30	0.26

a. Four values are reported because Extend keeps independent statistics for each server.



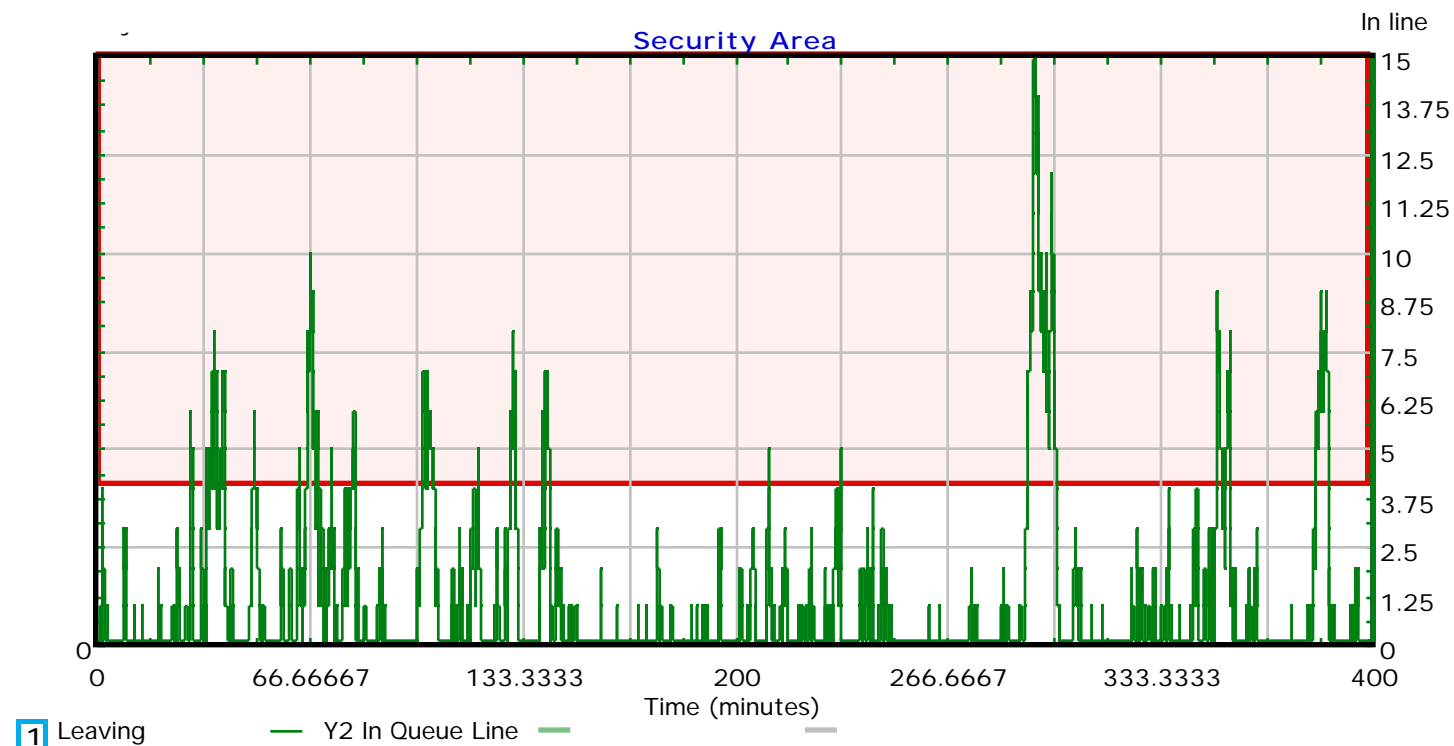
Answers to Parts (b) - (d)

- b) The number of x-ray machines should be 4 to provide a level of service below 2 minutes
- c) The the expected number of passengers at the checkpoint area on typical day in the design year should be around 0.97 persons (quite small for this type of facility but the result of the 2-min waiting time limit)
- d) The utilization of the future facility should be around 71% (average of four servers)



Answer to Part (e)

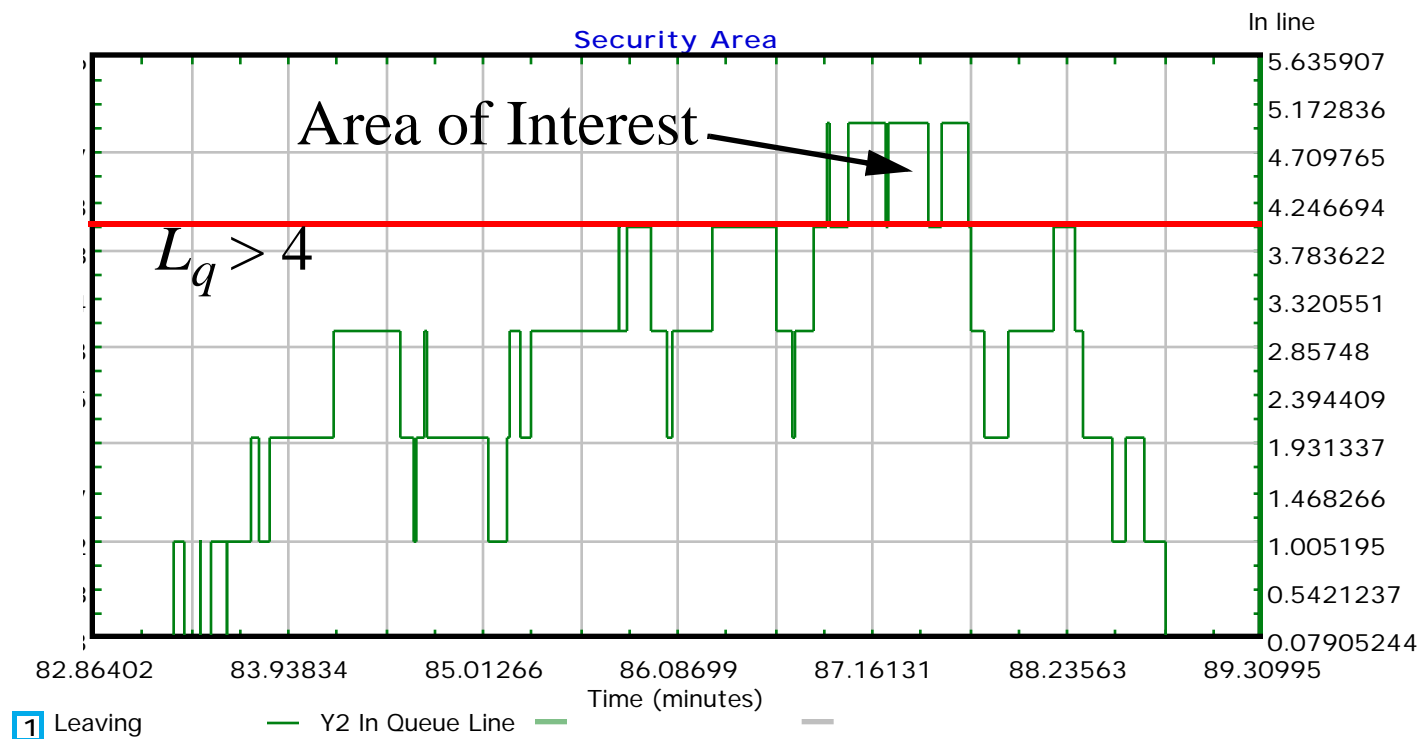
The Queueing Length (L_q) versus time plot provides insight on this. The red area corresponds to $L_q > 4$.





Answer to Part (e) - continuation

Isolating a small region of the queue length function we can see more clearly the function to be integrated





Answer to Part (e) - continuation

- The diagram is useful to guesstimate the probability of $L_q > 4$. However if we need an accurate answer we should extract the numerical values of L_q from the Extend output
- An equivalent way to execute this analysis is to define a new busy function in Extend such that,

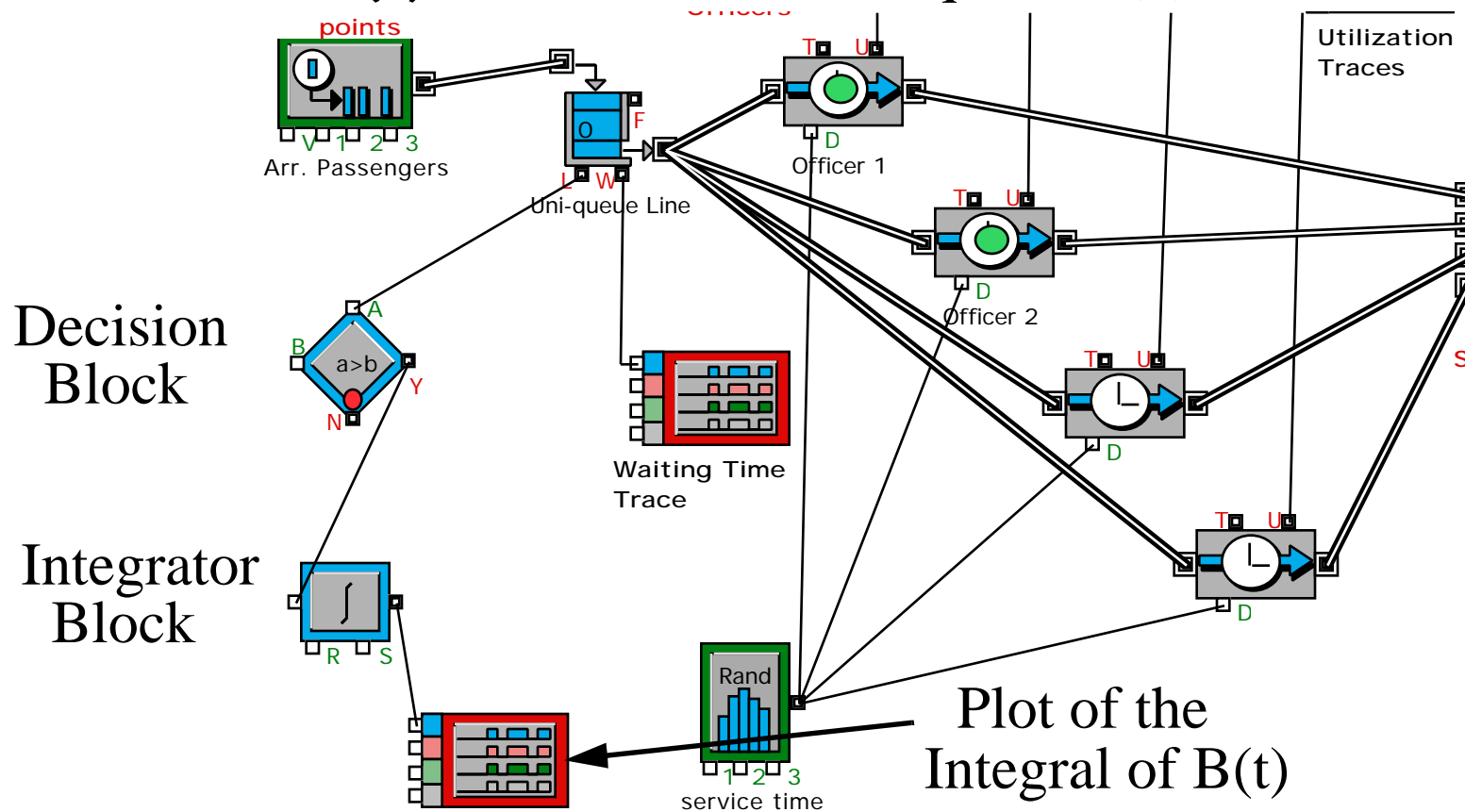
$$B(t) = \begin{cases} 1 & \text{if system if } L_q > 4 \\ 0 & \text{if system if } L_q \leq 4 \end{cases} \quad (1)$$

then integrate this function over time to obtain the number the percent of instances where $L_q > 4$.



Computation of $L_q > 4$ in Extend

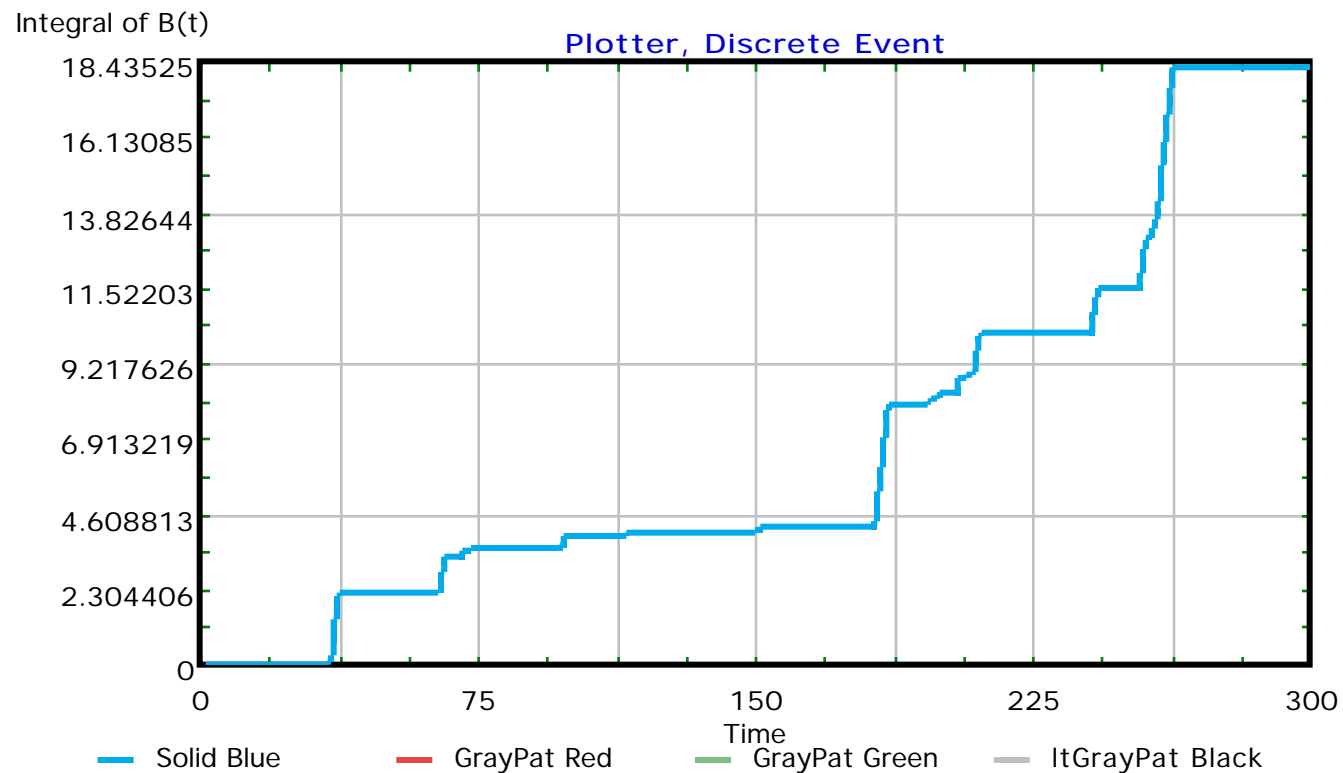
A decision block and an integrator compute the new busy function, $B(t)$, defined before in Equation (1)





Graphical Result of the Integral of $B(t)$

The result below shows the integral of $B(t)$ over time





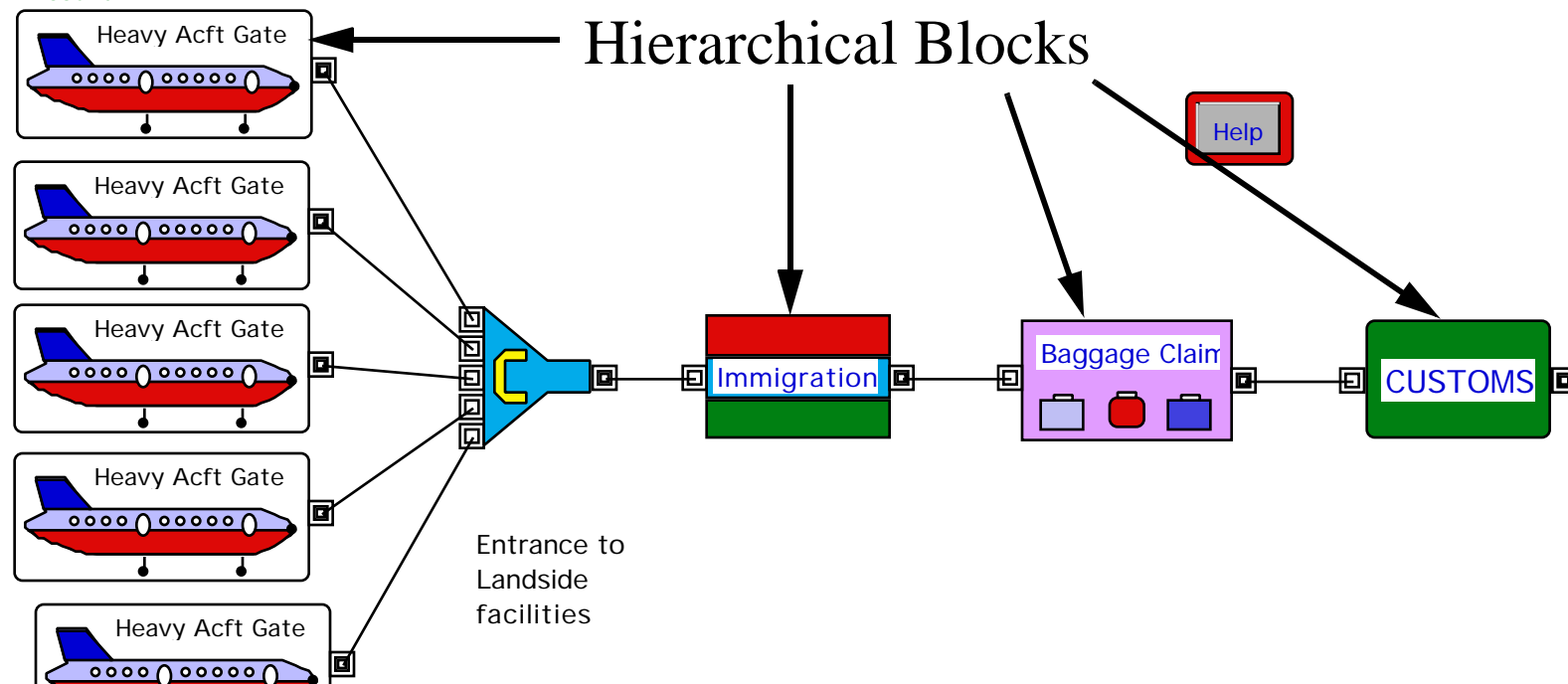
Avoiding Clutter with Hierarchical Blocks

- Hierarchical blocks are complex structures that contain a series of related Extend blocks
- Their main purpose is to avoid clutter in a complex model and, at the same time, provide a quick way to abstract complex behavior with minimum effort
- In simple terms, hierarchical blocks provide a way to create templates of behavior through the combination of multiple Extend blocks
- An example of a hierarchical block from a prototype landside simulation model developed at Virginia Tech is shown on the next page



Hierarchical Blocks of a Simple Airport Terminal Model (ALPS - Airport Landside Performance System)

These are some of the hierarchical blocks available in ALPS (an Extend library developed by Kulkarni and Trani, 1994)



The Importance of Attributes in Simulation

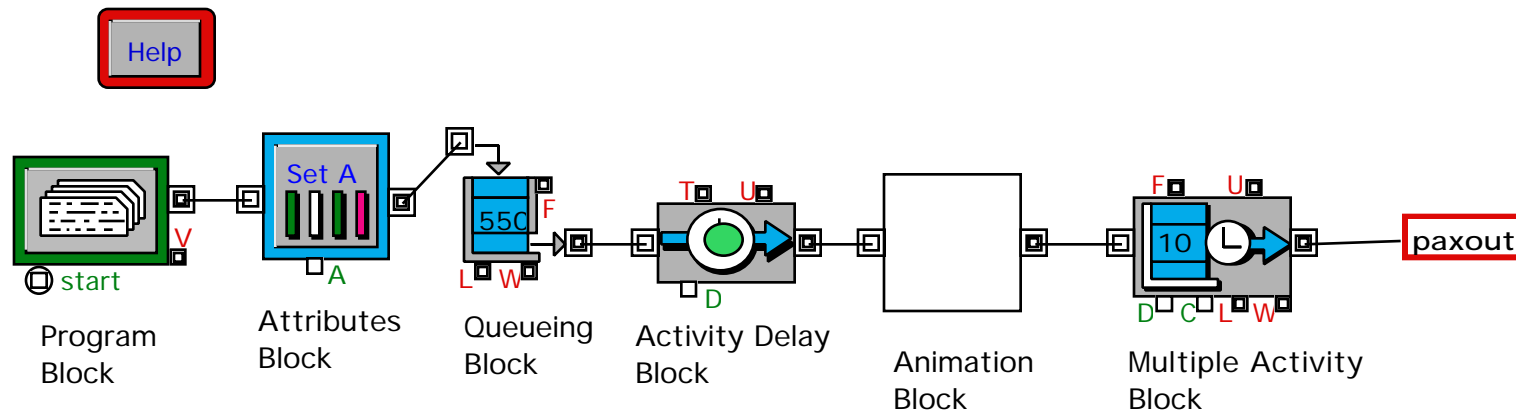


- **Attributes:** characteristics of entities used to differentiate behavior in the simulation process (i.e., transfer vs terminating passengers)
- Attributes move with the entity and thus can be “retrieved” in simulation blocks to perform various processes based on the attribute of an entity at time t
- Practically all simulation languages support attributes
- For example, suppose we are simulating domestic passenger baggage claim procedures. Some passengers carry **carry-on luggage**, others carry **full baggage** that needs to be retrieved from a direct feed baggage system.



Attributes (Illustration)

- If the percentages of passengers carrying full bags is known a random number generator and an attribute block can be used to model both types of passengers
- In this example an attributes block is used to assign either carry-on or full bags to each passenger



APM Simulation Model (Lin and Trani, 1995)



Purpose of research model

- To model individual passenger and APM vehicle movement at airports
- To determine the sensitivity of system performance
- To estimate the APM vehicle energy consumption
- To examine the flexibility of an APM system

Generically we call this model APMSIM

APM Requirements Analysis



Level of Service Analysis

APM Demand Analysis

Capacity Analysis

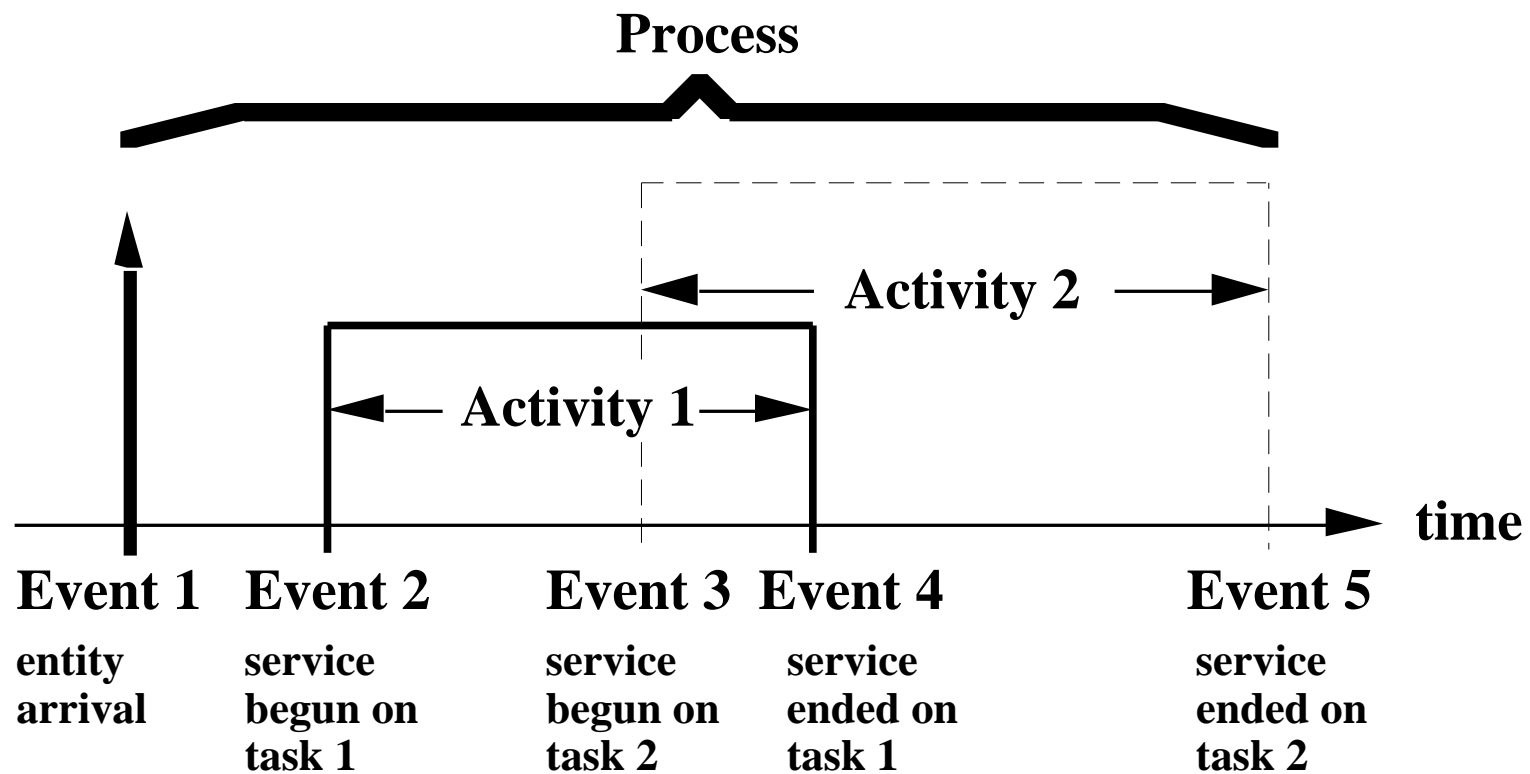
Flow Analysis

Energy Consumption Analysis



Methodology

Developed a hybrid simulation model in Extend™



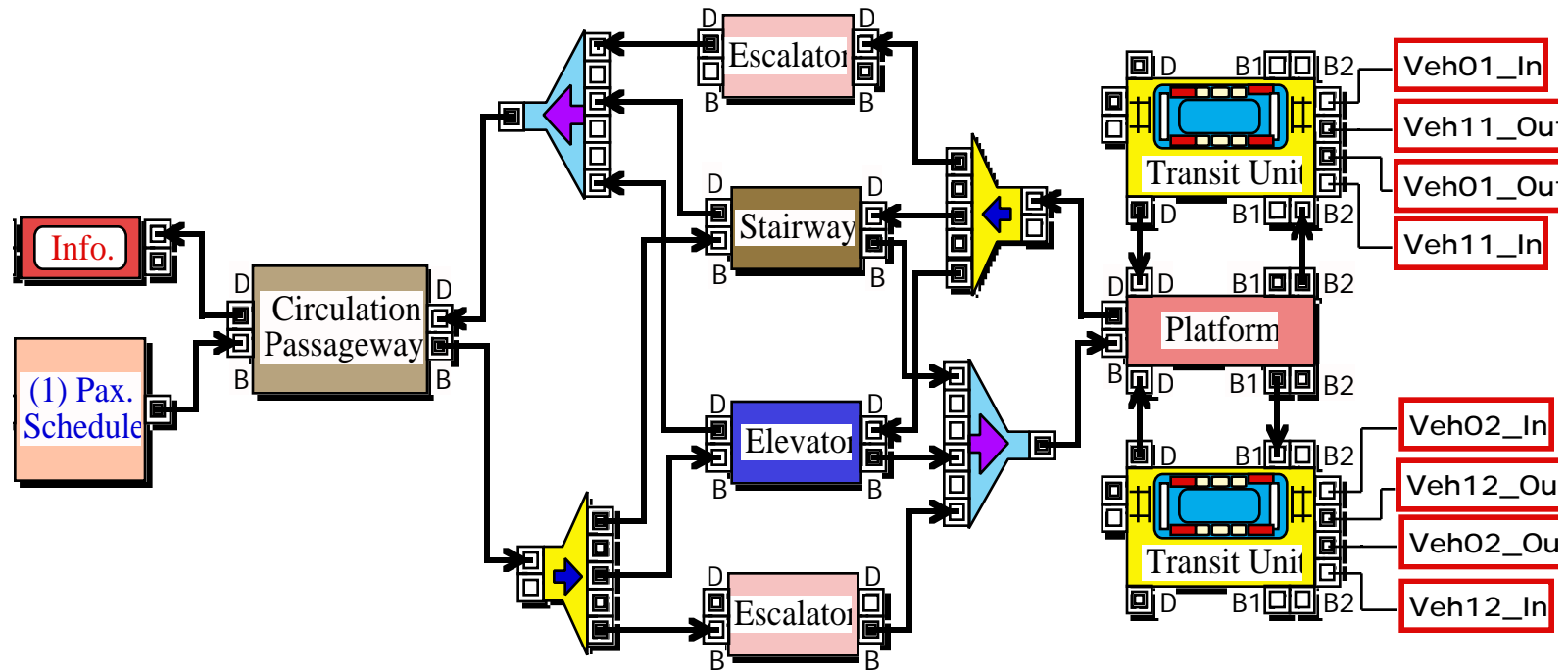
APM Simulation Model Description



- APM Simulation Model
 - APM station model
 - APM guideway model
 - Energy consumption model
- Simulation Model Logic
 - Passenger flows
 - Vehicular flows



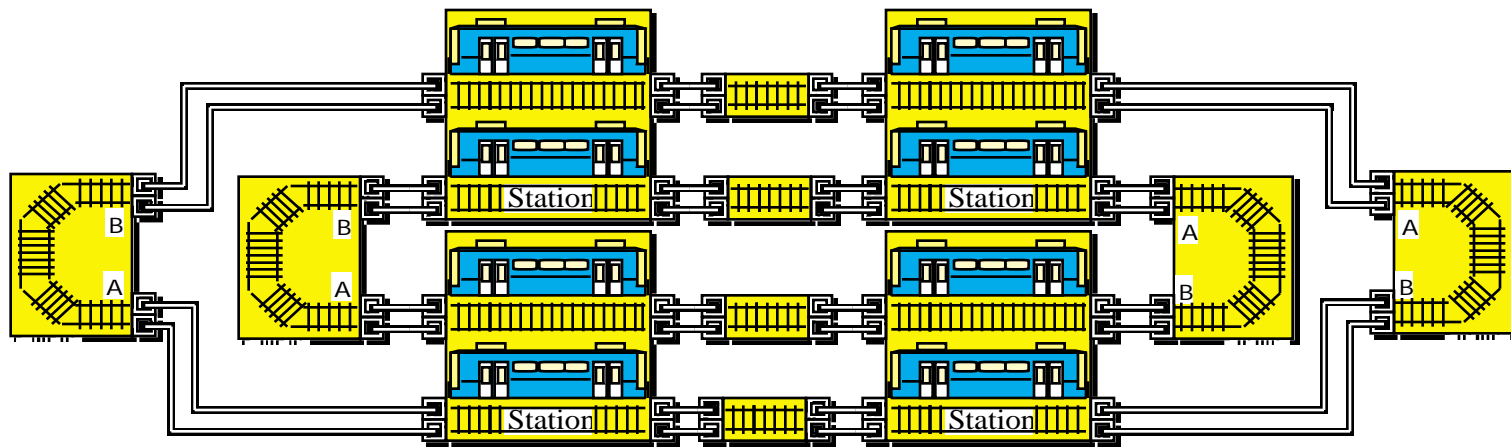
Sample Schematic of Station Model



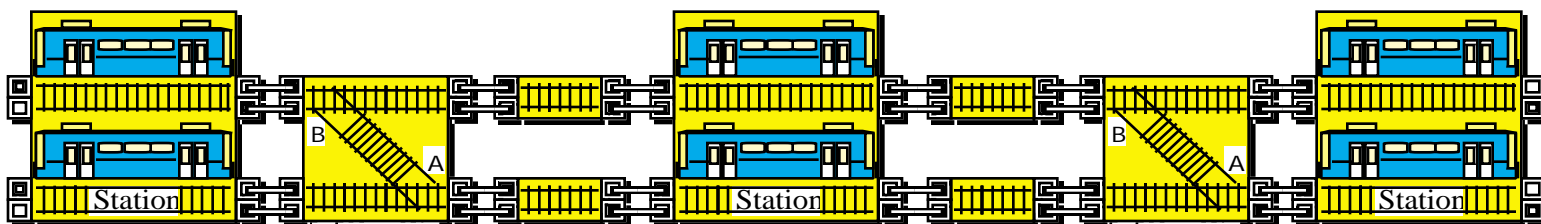
→ Boarding Passenger Flow Direction
← Deboarding Passenger Flow Direction



Sample APM Guideway Model



Double-Lane Loop



Pinched Loop with Turnbacks



APM Energy Consumption Model

The purpose of this submodel is to estimate the following:

- Speed
- Acceleration
- Travel Time
- Travel Distance
- Power Requirements
- Energy Consumption
- LOS (Level of Service)
- Occupancy and Load Factor



Energy Modeling

- David's Equation, Power required and energy consumed
- Integrate these relationships over time (use Extend's RK-4 integration algorithms)

$$R_{(a+r)} = K_0 + \frac{K_1}{w} + B(V) + \frac{CAV^2}{wn}$$

$$P = \frac{T(V)}{\eta}$$

$$E = 3.6 \times 10^{-6} \int_0^t P dt$$



Typical Station Results

- Travel time for an individual passenger
- Average waiting time for a facility or a TU
- Total number of passengers arriving at or leaving a station or a facility
- Queue length at a facility
- LOS of a facility in terms of area per passenger



Typical Guideway Results

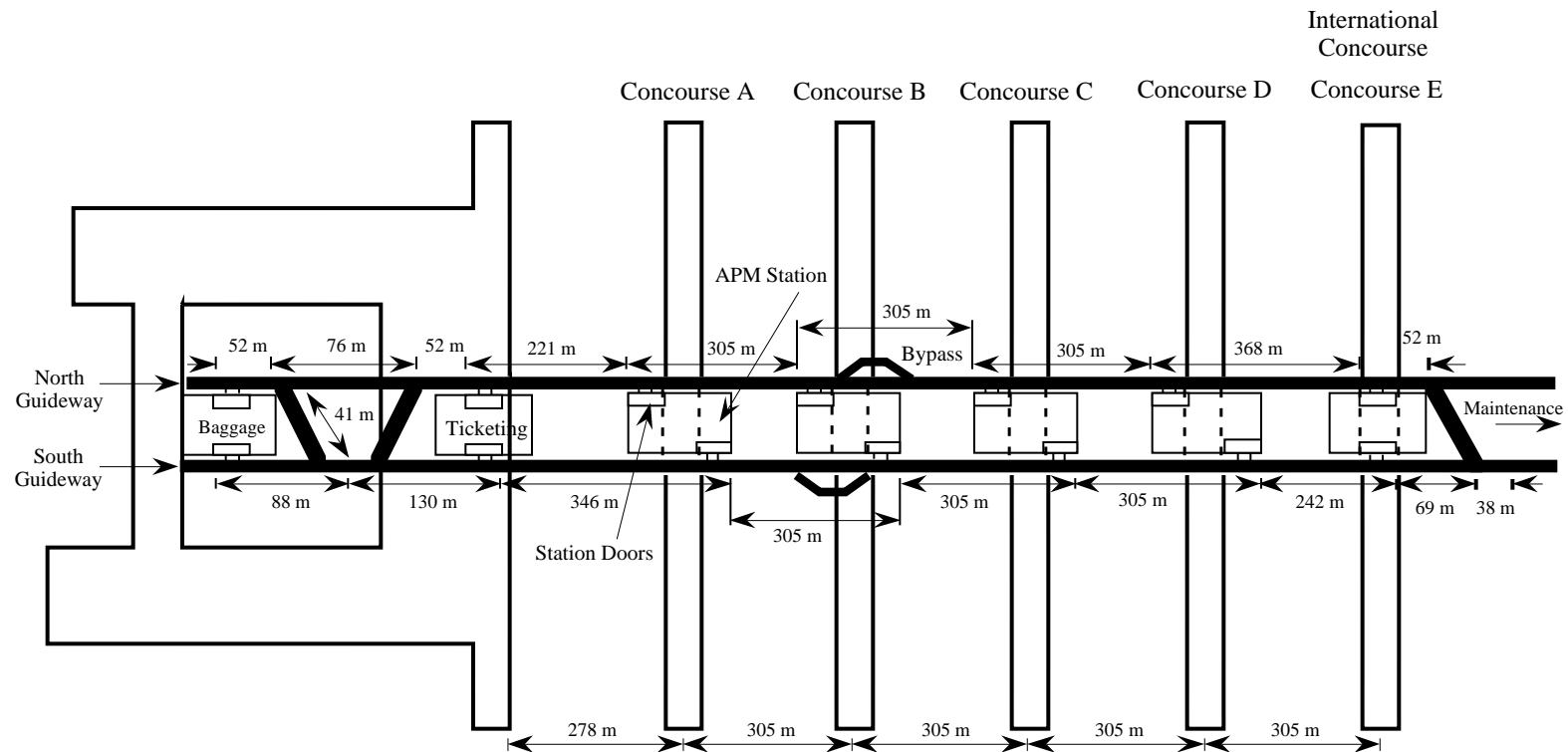
For a single guideway section or for the complete guideway network the model estimates:

- Travel time of an individual TU or passenger
- Occupancy and load factors of a TU
- TU speed, acceleration, deceleration, and travel distance
- TU power requirements and energy consumption
- Number of TUs in a guideway or a system
- LOS in a vehicle



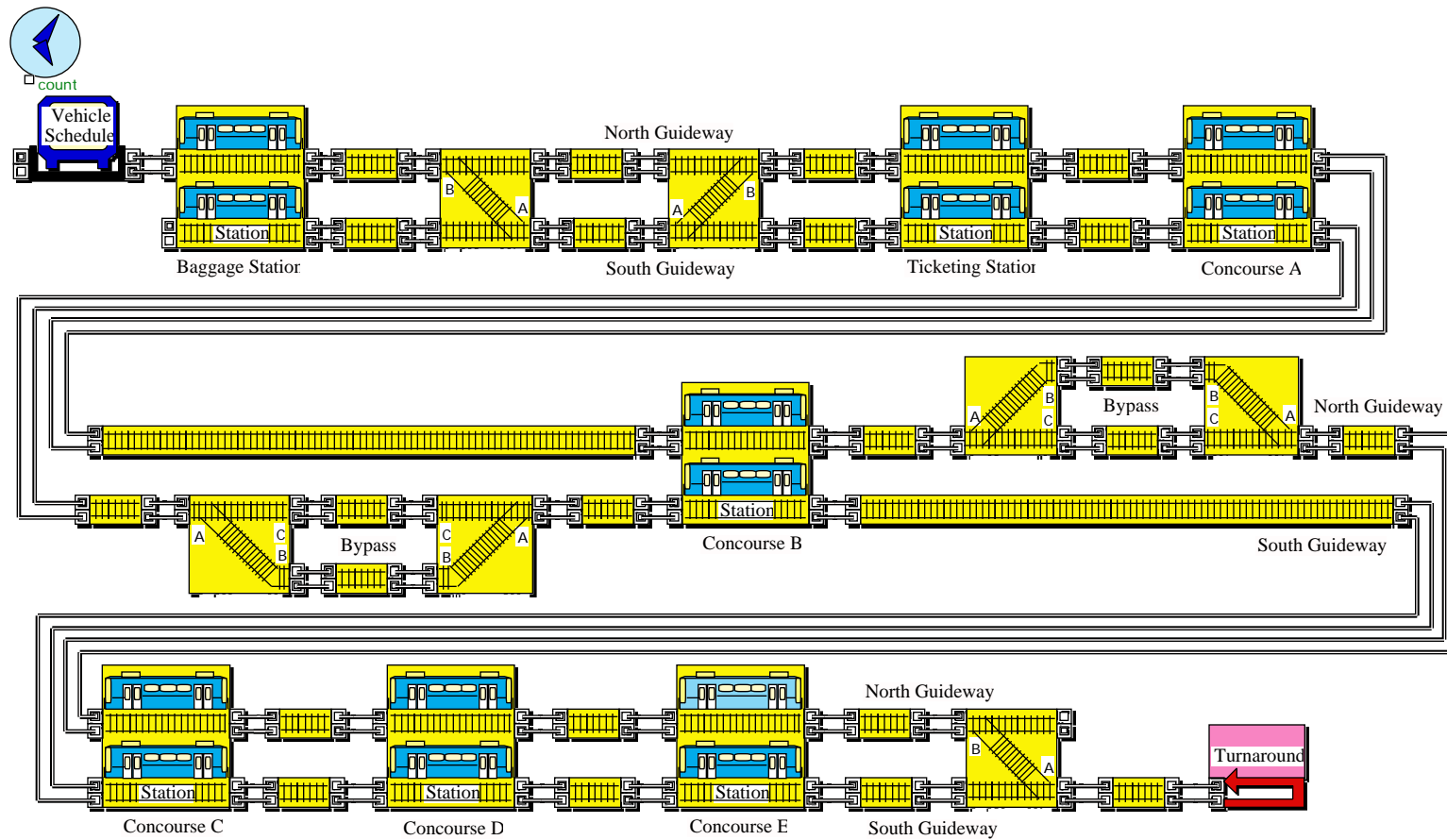
Sample Application of the APMSIM Model

We modeled Atlanta Hartsfield Intl. Airport in the US





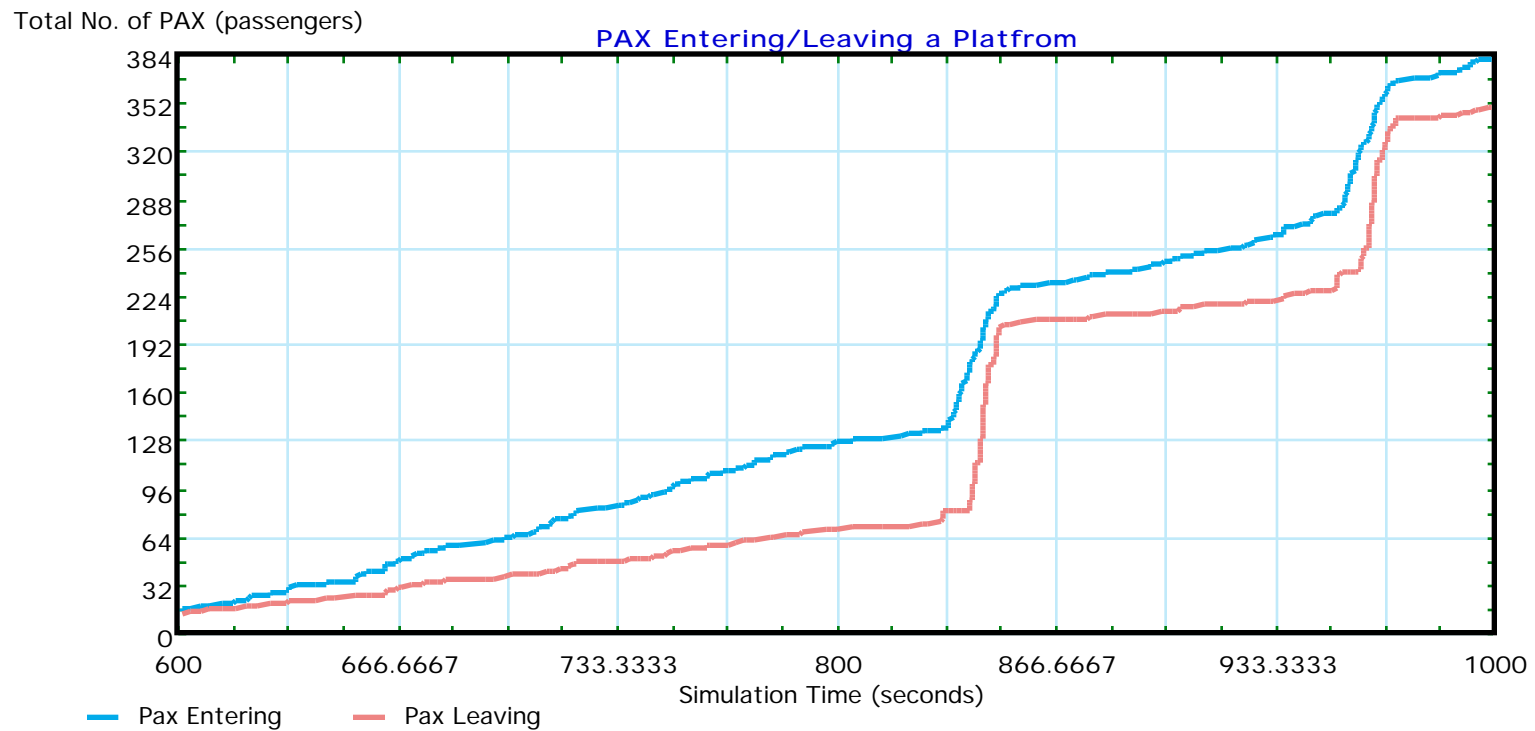
Atlanta APMSIM Guideway Model





Atlanta APMSIM Station Results

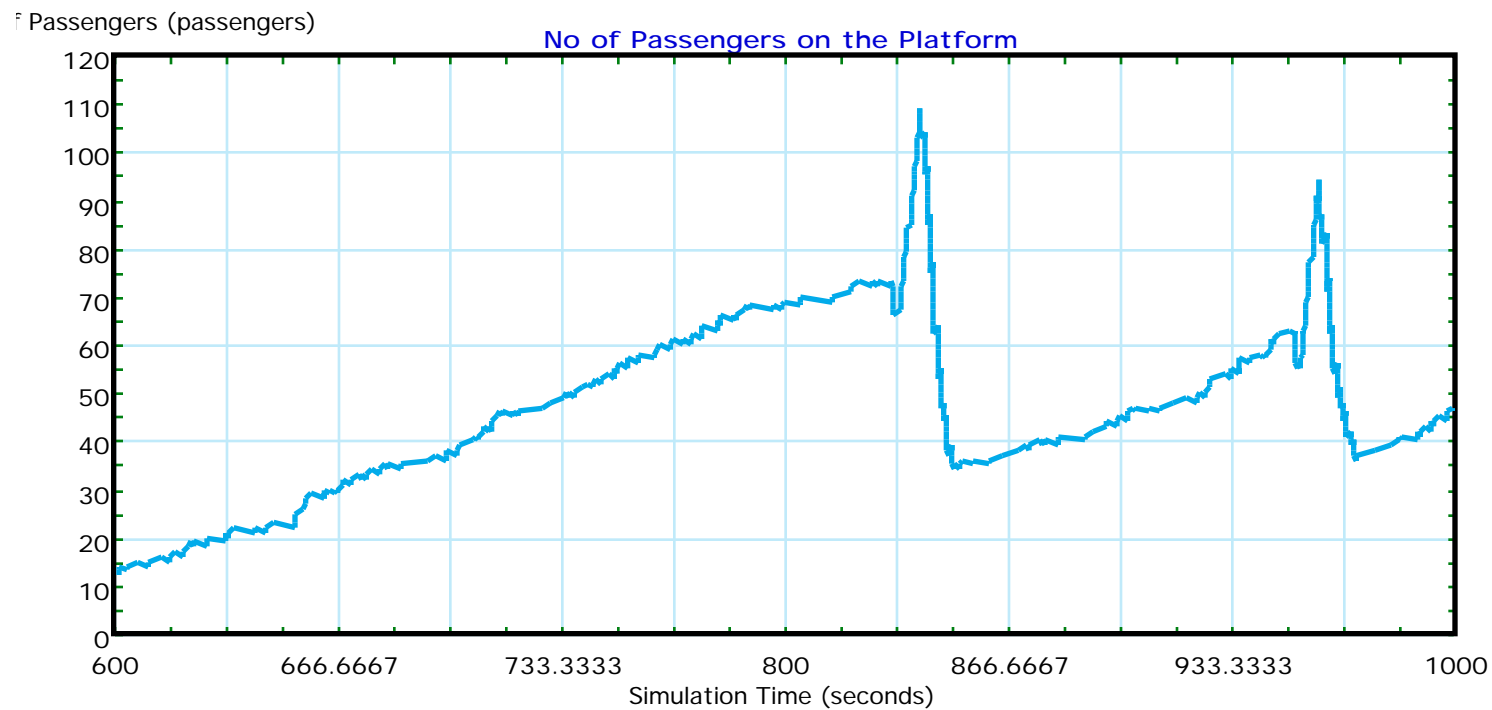
Passengers entering and leaving a station





ATL APMSIM Simulation Results

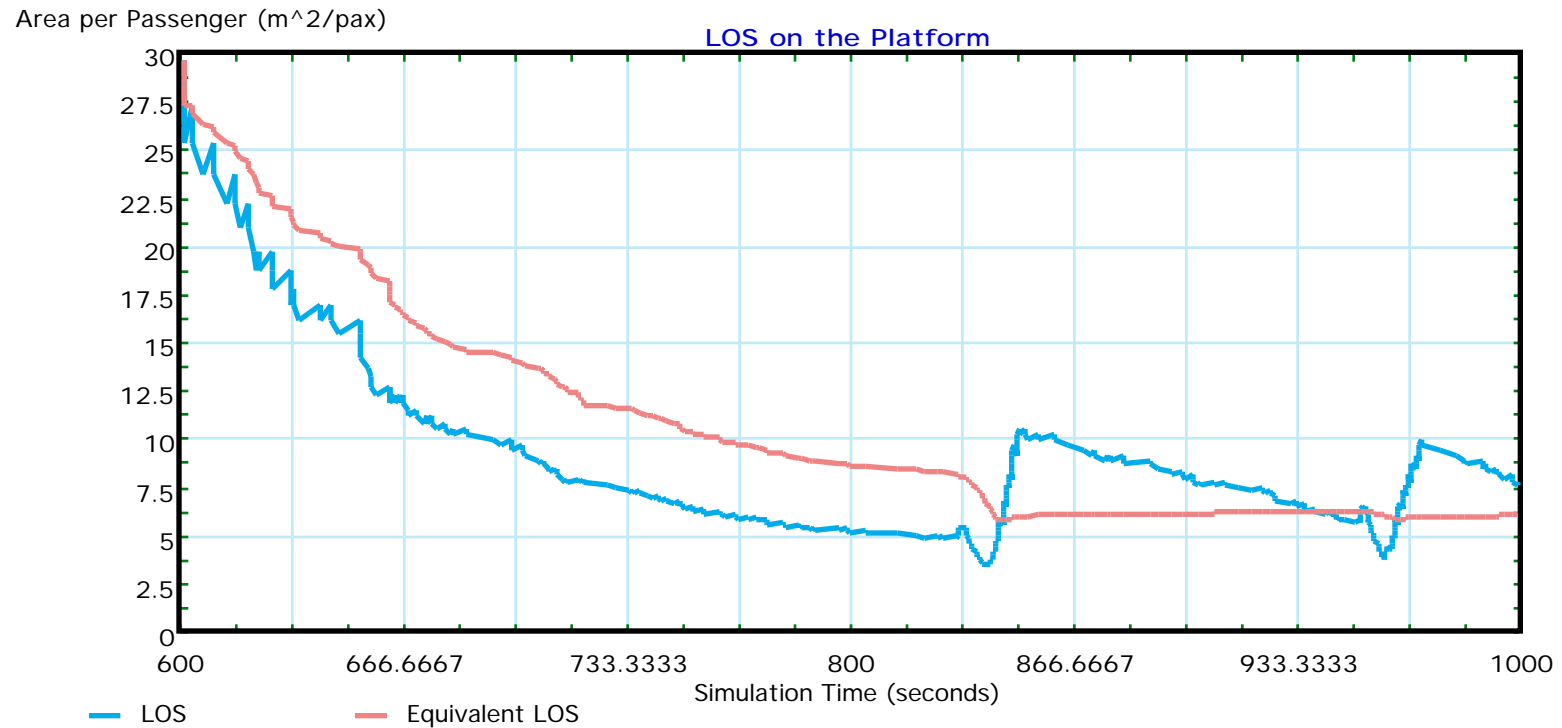
Passengers waiting at a station (on the APM platform area)





ATL APMSIM Results

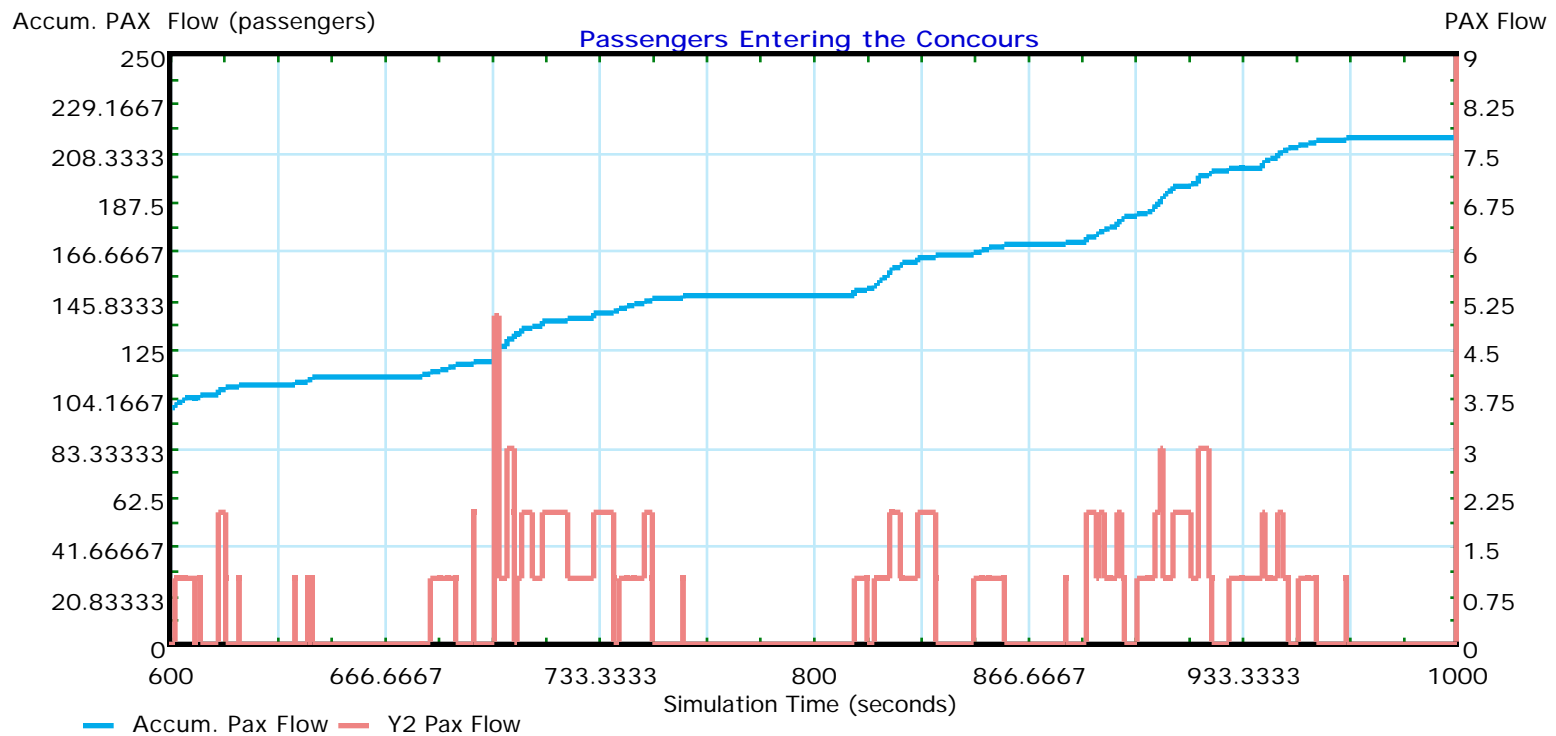
Level of service at the APM platform





ATL APMSIM Simulation Results

Number of Passengers Entering a Concourse



ATL APMSIM Development Conclusions



These are conclusions of this model development effort:

- Able to model various types of APM system configurations
- Useful to estimate the effects of changes in the system by modifying the input data
- Easy to model alternative APM service concepts
- Capable of simulating an APM system for different scenarios



APM or Other Transport Modes in Terminal Airport Models

- As demonstrated in the APMSIM model average statistics about LOS can be obtained in discrete and hybrid simulations
- These statistics can, in the end, be used to compare the static and macroscopic values typically used in airport terminal design
- As engineers and analysts we have to get the point to decision makers that airport terminals cannot be viewed and designed using static metrics
- It is more relevant to know how queues form and dissipate at processing facilities than computing a “perfect-gas molecular” equivalent (area per passenger)



References

- 1) Lin, Y. *A Simulation Model of an Automated People Mover at Airports*. M.S. Thesis. Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.
- 2) Kulkarni, M. *Development of a Landside Terminal Simulation Model*. M.S. Thesis. Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.
- 3) Fruin, J.J. *Designing for Pedestrians*. in *Public Transportation Systems*. Hoel and Gray: Editors. John Wiley and Sons, New York, 1993.
- 4) IATA. *Airport Development Reference Manual: 8th Edition*. International Airline Transport Association, Montreal, 1995.

















