Scripting Technique to Transform Use Case Scenarios to Test Cases

2012 Annual IV&V Workshop

Jesse Musgrove, Andy Chen, Jim Savarino,
Shirley Savarino

Background

- E-911 emergency communication networks has a complex system of systems architecture.
- New techniques were needed to aid IV&V in analyzing this type of communication network.
- Simulink (Architecture) model and Layered Queuing Network (Performance) model of the overall system were developed to assure the correctness of sub-system interfaces and to evaluate the boundary conditions
- The use of the system is specified in Use Case Scenarios include various call attributes, scripts were utilized to transform these Use Case Scenarios into test cases for the system simulation.
- This presentation describes techniques for the transformation of Use Case Scenarios into Test Cases and how it also enabled us to simulate more realistic real-world operations.

Scripting Products and Process Overview

The Data Dictionary captures a Use Case and feeds the Simulink and Layered Queuing Network (LQN) models. We pulled all variables out of the model and made this aspect table driven

Data Dictionary:
Parameters and Variables for a particular instance (Use Case)

GUI
Translates a use case to input data for each model.

C# based Graphical User Interface (GUI) takes the parametric Use Case and turns it into individual calls and attributes for Simulink, call rate and call type probabilities for LQN.

Architecture (Simulink Model)
- Varying levels of fidelity
- Allows concatenation of Use Cases
- Allows verification of interfaces

Performance (LQN Model)
- Performance across a time range
- Evaluate Boundary Conditions

The Simulink model reflects the system architecture at various levels of abstraction. Erlang distribution used for internal call processing parameters.

Models are validated against each other for correctness

The LQN model allows rapid prototyping and is a "process model". Inputs are parametric, characterizing the use case

Data Dictionary

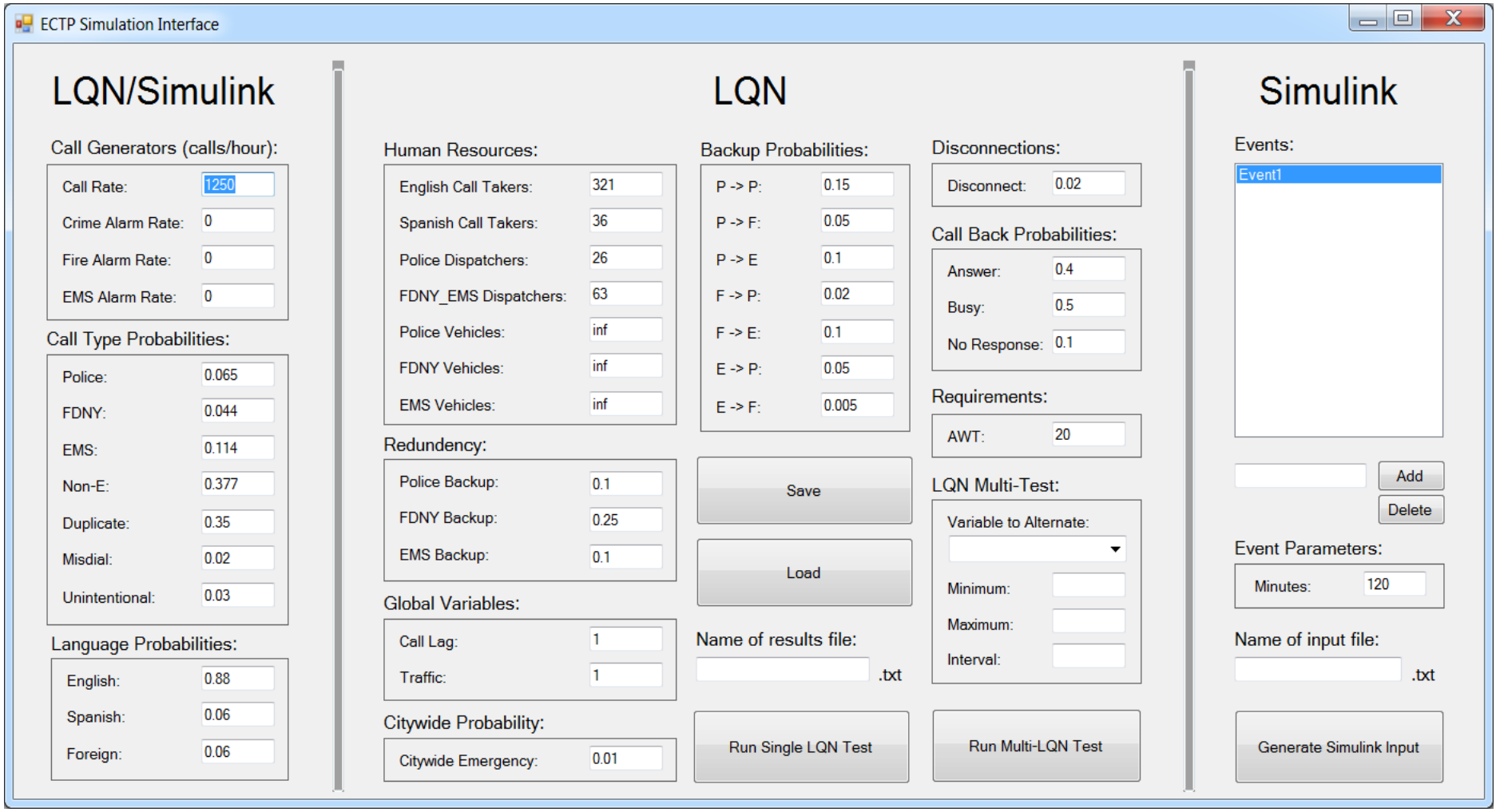
- **Use Case data values captured in Excel, automated feed to each model**
- **Data Dictionary Rigor – required where we obtained data from and can be validated as artifacts become available**

Action	Actor	SSV (seconds)	Variance	SSV obtained from:	Definition
Get_ANI_ALI	CPE_NACD	0	0	Estimated value	Request the ANI/ALI from MSAG.
Find_Call_Taker	CPE_NACD	0	0	Estimated value	Route the call to a call taker.
Phone_to_ALI	MSAG	2	0.3	Estimated value	Lookup the address based on the phone number.
Opening	English_Call_Taker	4	2	Estimated value	During this time the call taker establishes whether or not the call was an accident and either begins to identify the caller's language or begin's getting information.
Hang_Up	English_Call_Taker	3	0	Estimated value	Hang up if the call was a mis-dial.
Evaluate_Language	English_Call_Taker	10	0	Estimated value	If the call wasn't an accident, evaluate the caller's language.
Verify_Accidental_Call	English_Call_Taker	21	6	Estimated value	If the call was unintentional and the caller is unaware, the call taker must stay on the line until they can be reasonably sure it was accidental.
Transfer_Call	English_Call_Taker	3	0	Estimated value	In the case the caller is Spanish speaking, transfer the call.
Get_Info_Location	English_Call_Taker	28	12	Estimated value	If the caller is an English speaker, get information and verify their location.
Explain_Dup	English_Call_Taker	5	1	Estimated value	If the incident occurred at the location of an already-reported incident, explain that it is a duplicate and end the call.
Get_Incident_Type	English_Call_Taker	13	7	Estimated value	Get information about the emergency.
Continue_Getting_Info_1	English_Call_Taker	20	4	Estimated value	
Continue_Getting_Info_2	English_Call_Taker	77	30	Estimated value	
Dial_appropriate	English_Call_Taker	12	4	Estimated value	If the caller is calling about a non-emergency, quickly give them the correct number and end the call.
Contact_FLS	English_Call_Taker	20	13	Estimated value	Contact Foreign Language Servies (this happens when the caller is neither English nor Spanish-speaking).
F_Get_Info_Location	English_Call_Taker	84	28	Estimated value	Same as above, but through the FLS.
F_Explain_Dup	English_Call_Taker	10	7	Estimated value	Same as above, but through the FLS.

Data Dictionary and Ingest

- **From TASC TSAT experience, goal is to have a data dictionary that automatically feeds into the model and simulation**
  - Each instantiation of the data dictionary is a use case (e.g. how system works at a certain time)
  - The use cases can be concatenated to come up with a combination of use cases, or a use case scenario
- **Data Dictionary came from the LQN development, but uses the same information for Simulink simulation**
- **Differences in how Data Dictionary used in LQN and Simulink**
  - LQN model works off a single use case while the Simulink model works off of both a single use case as well as a use case scenario
  - LQN model uses average values and means-standard deviation/ probabilities
  - Simulink model requires these statistical values to manifest themselves in a sequence of calls ← we accomplished this with the C# based GUI to perform the translation

C# Based GUI for LQN and Simulink Inputs



LQN Input and Output Examples

- Input variables extracted from Data Dictionary's steady-state values
- Varying the call rate and observe the performance response of the model
- Important output parameters:
  - Average Wait Time in Queue
  - Percent Utilization
  - Maximum call rate when the system becomes saturated (wait time -> infinite)

Input Example

input.lqn - Notepad

File Edit Format View Help

```
G
"LQN Model for ECTP 911 Model"
1e-05 50 1 0.5 -1

P 3
p PhoneComp f
p PhoneQueue f
p CallTakerProc f i
-1


T 4
t CPE_NACD n Phone_to_ALI -1 PhoneComp i
t Phone_Queue n Phone_Queue_E -1 PhoneQueue i
t English_Call_Taker f Eng_Handle_Call -1 CallTakerProc m 321
t Spanish_Call_Taker f Spa_Handle_Call -1 CallTakerProc m 36
-1


E 4

a Phone_to_ALI 0
s Phone_to_ALI 2 -1
s Phone_Queue_E 0.0001 -1
z Phone_to_ALI Phone_Queue_E 1.0 -1
z Phone_Queue_E Eng_Handle_Call 1.0 -1
A Eng_Handle_Call Opening
A Spa_Handle_Call S_Get_Info_Location

-1


A English_Call_Taker
s Opening 4
s Hang_Up 3
s Evaluate_Language 10
s Verify_Accidental_Call 21
s Transfer_Call 3
z Transfer_Call Spa_Handle_Call 1.0
s Get_Info_Location 28
s Contact_FLS #Contact_FLS_SSV#
s Explain_Dup 5
s Get_Incident_Type 13
s Dial_Appropriate #Dial_Appropriate_SSV#
s F_Get_Info_Location #F_Get_Info_Location_SSV#
s F_Explain_Dup #F_Explain_Dup_SSV#
s F_Get_Incident_Type #F_Get_Incident_Type_SSV#
s F_Dial_Appropriate #F_Dial_Appropriate_SSV#
s Continue_Getting_Info_1 20
s Continue_Getting_Info_2 77
s F_Continue_Getting_Info_1 #F_Continue_Getting_Info_1_SSV#
s F_Continue_Getting_Info_2 #F_Continue_Getting_Info_2_SSV#
:
Opening -> (0.02)Hang_Up + (0.95)Evaluate_Language + (0.03)
Verify_Accidental_Call;
Evaluate_Language -> (0.06)Transfer_Call + (0.88)Get_Info_Location +
(0.06)Contact_FLS;
Get_Info_Location -> (0.263157894736842)Explain_Dup +
(0.736842105263158)Get_Incident_Type;
Get_Incident_Type -> (0.538571428571429)Dial_Appropriate +
(0.461428571428571)Continue_Getting_Info_1;
Contact_FLS -> F_Get_Info_Location;
F_Get_Info_Location -> (0.263157894736842)F_Explain_Dup +
(0.736842105263158)F_Get_Incident_Type;
F_Get_Incident_Type -> (0.538571428571429)F_Dial_Appropriate +
(0.461428571428571)F_Continue_Getting_Info_1;
Continue_Getting_Info_1 -> Continue_Getting_Info_2;
F_Continue_Getting_Info_1 -> F_Continue_Getting_Info_2
-1
```

Output Example

basetestresults.txt - Notepad

File Edit Format View Help

CallRate	AvgECTQueueTime	ECT_Utilization	EngProbOfDelay	EngAcceptablePercent	AvgSCTQueueTime	SCT_Utilization	SpaProbOfDelay	SpaAcceptablePercent
1250	0	0.0919003	5.6159E-194	1	0	0.043752	1.63955E-34	1

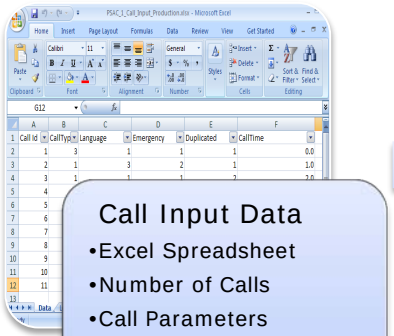
Input to Simulink Generated by Graphical User Interface

Call Id	CallType	Language	Emergency	Duplicate	CallTime	MistakeCalls	EngTime	SpaTime
1	3	1	2	2	0.761016	1	73.69273	0
2	3	1	1	1	0.800369	1	22.33658	0
3	3	1	1	2	1.441605	1	320.2232	0
4	3	1	2	1	1.524414	1	35.0576	0
5	2	1	1	2	1.581899	1	300.7378	0
6	3	1	1	1	1.772106	1	36.52259	0
7	1	1	1	2	1.803306	1	49.88416	0
8	1	1	1	2	1.965316	1	175.3064	0
9	1	1	1	1	1.97221	1	67.78084	0
10	1	1	1	2	2.4721	3	13.48574	0
11	1	1	2	1	2.519719	1	53.81892	0
12	2	1	2	1	2.797478	1	15.76436	0
13	3	1	2	1	2.909964	1	62.32538	0
14	1	1	1	2	3.097967	1	298.2171	0
15	2	1	1	1	3.52046	1	27.99107	0
16	3	1	2	2	3.894295	1	45.51436	0
17	3	1	1	1	3.939857	1	50.57341	0
18	2	1	1	2	3.940611	1	421.0298	0
19	2	1	1	2	4.061791	1	65.4083	0
20	2	1	2	1	4.162769	1	101.9087	0
21	2	1	1	1	4.327254	1	22.33524	0
22	1	1	2	2	4.694573	1	32.27234	0
23	3	1	1	2	4.760872	1	76.07065	0
24	3	1	2	1	4.923177	1	35.74328	0
25	3	1	2	2	5.839144	2	11.56411	0
26	1	1	2	2	6.172656	1	128.0366	0
27	1	1	2	1	6.236681	1	64.58458	0
28	3	1	2	1	6.489637	1	112.1173	0
29	2	1	2	2	6.680112	1	101.1827	0
30	3	1	2	1	6.724102	1	18.47971	0
31	1	1	1	2	6.906209	1	200.4428	0
32	1	1	2	1	7.098536	1	16.12733	0
33	1	1	1	1	7.234759	1	38.45216	0
34	3	1	1	1	7.387739	1	52.64261	0
35	3	1	1	1	7.41498	1	33.36559	0
36	3	1	1	1	7.879871	1	28.17464	0
37	2	1	2	1	8.640952	1	19.56078	0
38	1	1	1	2	8.660624	1	74.09823	0
39	1	2	2	1	9.831435	1	7.663426	32.92
40	3	1	1	2	10.88006	1	100.6139	0
41	3	1	2	1	10.91287	1	31.70782	0
42	3	1	1	2	11.56321	1	152.1454	0
43	3	1	2	1	11.6231	1	109.4732	0
44	3	1	1	2	11.87309	1	42.30147	0
45	2	1	1	1	12.13918	1	33.54747	0
46	3	1	1	1	13.35771	1	46.62342	0
47	1	1	1	2	13.42968	1	257.0162	0
48	3	1	2	1	13.66644	1	4.082378	0
49	2	2	2	1	13.67723	1	18.46082	10.08882
50	2	1	2	1	13.74729	1	33.03506	0

Note: there were actually around 320K calls for the 24 hour period

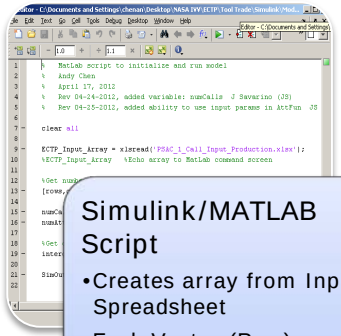
Simulink Data Flow

- The objective is to represents an IP packet switching network with an acceptable fidelity
- Input data (generated by GUI) in Excel spreadsheet format with each row represents a call and each column represents a call attribute
- Script reads in the spreadsheet and saves the input data in MATLAB workspace in an array format
- Simulink model uses SimEvents to generate entities based on array vectors with attributes, and routes them through the system
- Results are shown in graphical format, and data is logged for post analysis



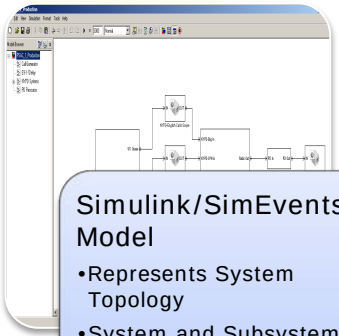
Call Input Data

- Excel Spreadsheet
- Number of Calls
- Call Parameters (Attributes)
- Probabilities are computed by aggregated attribute values
- Time of calls is made an explicit call attribute



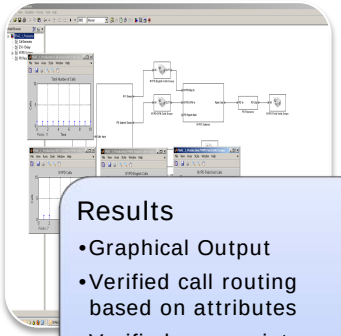
Simulink/MATLAB Script

- Creates array from Input Spreadsheet
- Each Vector (Row) represents a call with designated attributes
- Time increases monotonically within array



Simulink/SimEvents Model

- Represents System Topology
- System and Subsystems
- External MATLAB functions
- Call Routing and Delay

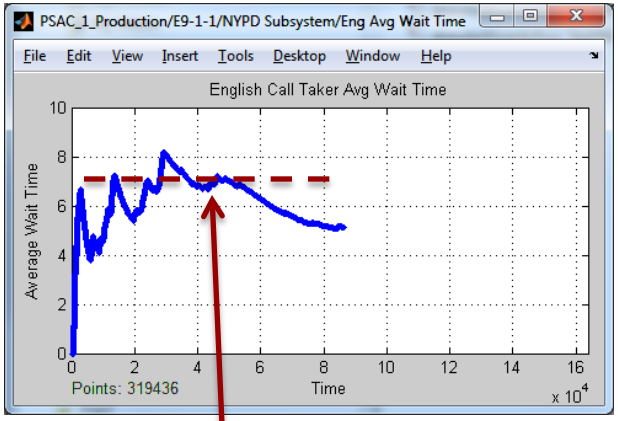


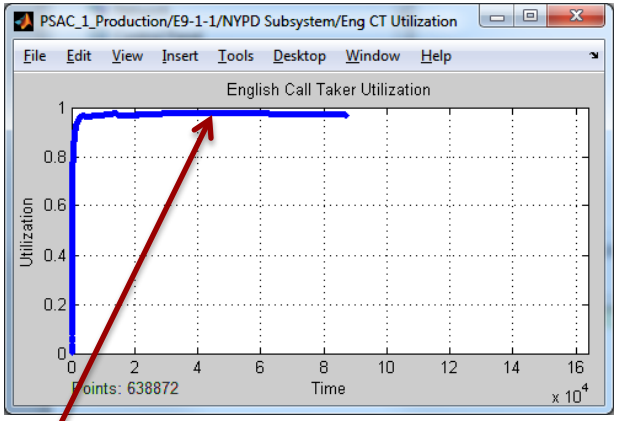
Results

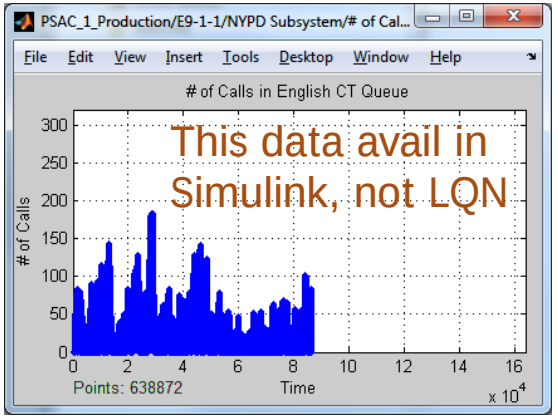
- Graphical Output
- Verified call routing based on attributes
- Verified appropriate implementation of delays (Call Taker and Dispatcher processing)

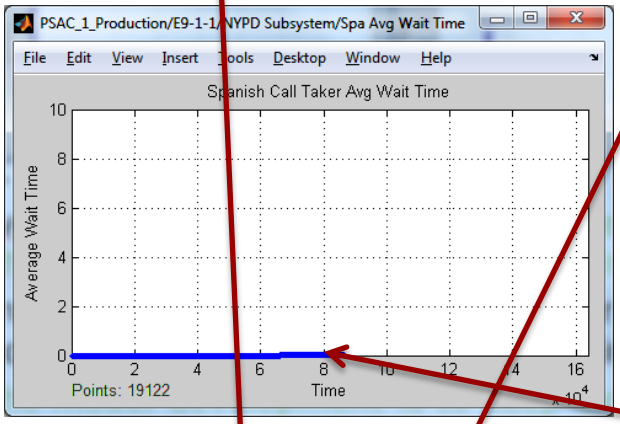
Simulink and LQN Results Comparison at 13k Calls/hour for 24 hrs (we wanted to make sure models were in sync)

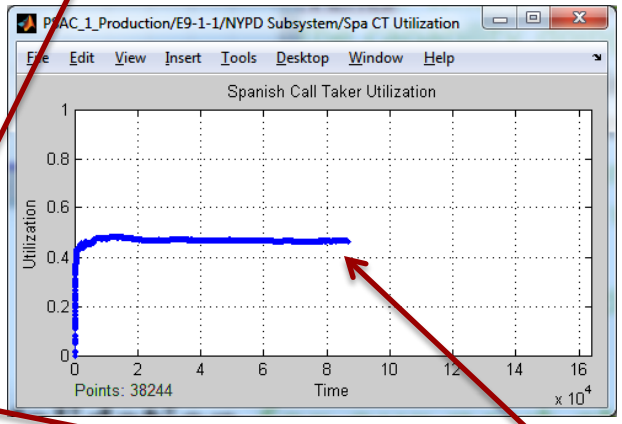
Simulink Results

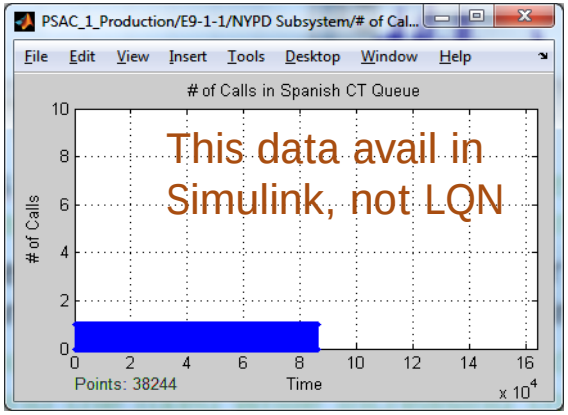












LQN Results

CallRate	AvgECTQueueTime	ECT_Utilization	EngProbOfDelay	EngAcceptablePercent	AvgSCTQueueTime	SCT_Utilization	SpaProbOfDelay	SpaAcceptablePercent
13300	7.1544	0.97782	0.612330436	0.885415365	0.0001	0.465530556	6.55E-05	0.99999948

Note: some variation due to translation (e.g. LQN had 13300 calls and Simulink had ~13500 calls/hour in the 24 hour period)