SYSCON 2024

The 18th Annual IEEE International Systems Conference

April 15-18, 2024

Delta Hotels Montreal, Canada

Using Hybrid System Dynamics and Discrete Event Simulations to Identify High Leverage Targets for Process Improvement in a Skill-based Organizational Structure

Eric S. Enos

17 April 2024

WALTER SCOTT, JR.
COLLEGE OF ENGINEERING
COLORADO STATE UNIVERSITY

SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY

Agenda

- Introduction & Motivation
- Characterizing the Work
- Managing the Data
- Building and Iterating the Models
- Preliminary Conclusions and Future Research

Motivation for this Research

- Team is responsible for the enterprise technology stack supporting ~200 hospitals of various types and dozens of ambulatory clinics and physician practices in more than 30 states across the US. The **infrastructure is *big***.
- The team consists of ~150 people organized into smaller, skill-based times. The **teams are *small***.
- The "**DevOps in the public cloud**" model is *the* trend for infrastructure management – if you **write your own applications**…
  - **We do not** – many key clinical systems, including core Electronic Medical Records (EMRs) currently deployed, are **unsupported in the cloud**.
- The key question: can an IT organization like mine **leverage DevOps** processes, techniques and technologies to meet our mandate with **on-premises infrastructure**?

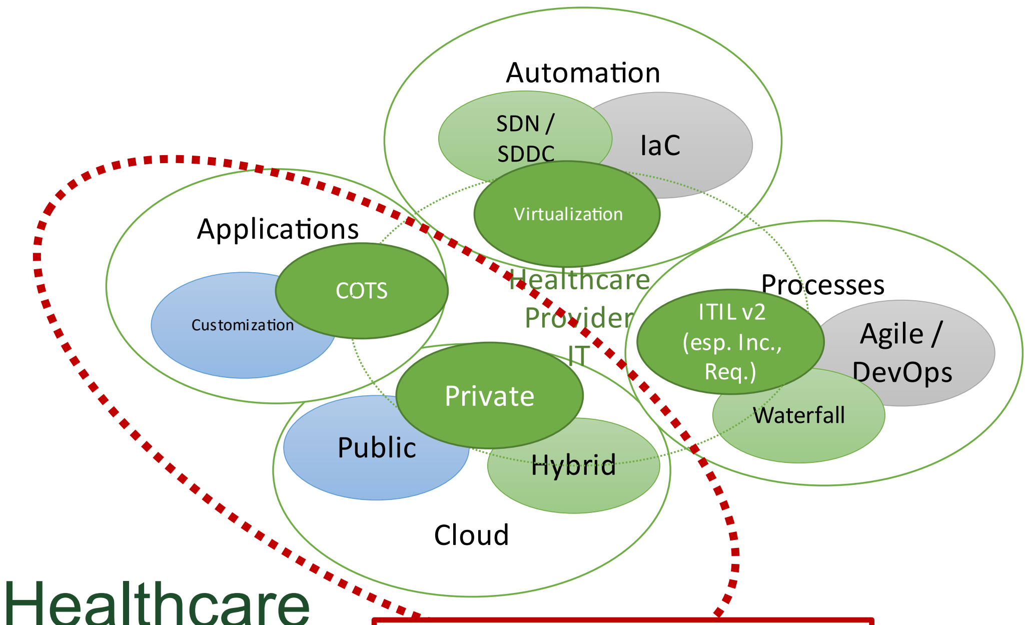
The Healthcare Provider IT Organization

- Following from the nature of the systems involved, many if not most provider IT organizations are built to evaluate, purchase, deploy, integrate, support and upgrade / replace systems (primarily on premises) – **not to build them.** Most **development is focused on integration and analytics**.
- Hospitals and health systems provide services from **extensive facilities** that require IT infrastructure throughout, with network and computing end-user equipment widely deployed.
- Teams are organized around **departmental silos**: *technologies* (such as networks or endpoint computing), *applications* (especially certain types of COTS applications), and *functions* (e.g., project management, integration, or data & analytics). **Cross-functional teams** are **few** and usually **ephemeral**, focused on specific projects.
- Cost pressures keep IT teams relatively small and outsourcing is common, so teams must be **responsible** for ***both* project** and **operational tasks** within their departmental silos.
- Common activities that require **multiple skills** must be **coordinated across departmental silos**.

- J. A. Mitchell, "Basic Principles of Information Technology Organization in Health Care Institutions," Journal of the American Medical Informatics Association, vol. 4, no. 2, pp. s31–s35, 1997, url: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC61488/.

Fundamentally

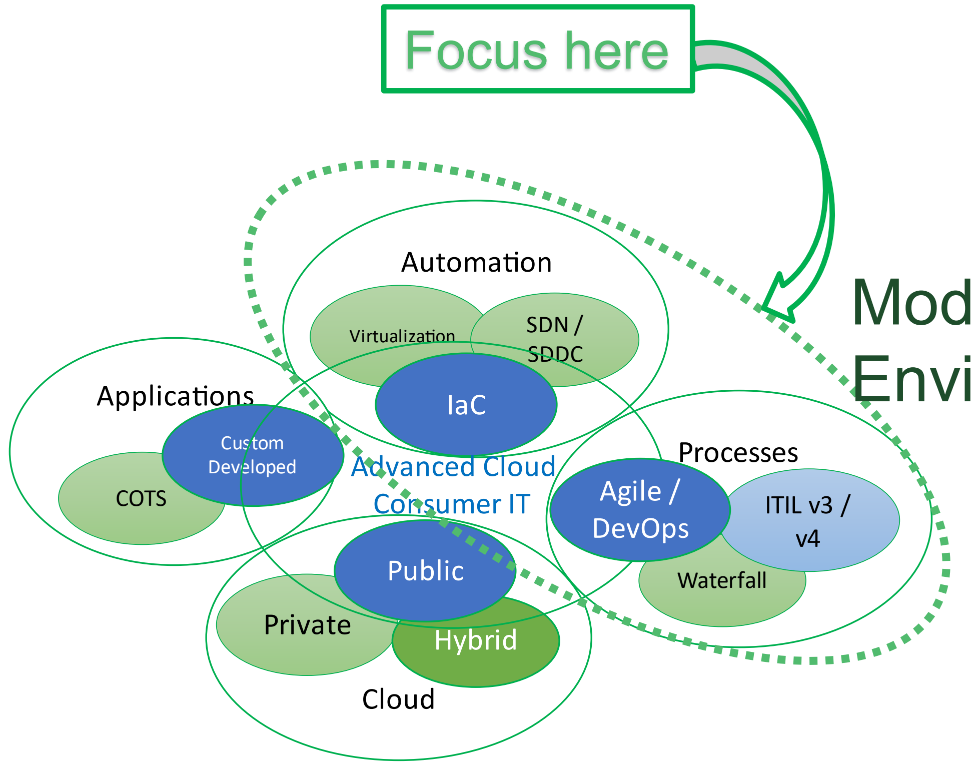
Can we make this…



The Healthcare Provider IT Environment

Not easily changed

Focus here



Modern IT Environments

…look more like this?

Characterizing the Work

Behavior observed in case study team

Basic Organizational Structure

- Teams are built around specific technical **skill**: network engineering, email, virtualization, server administration, etc.
- Teams support **both** operational **tickets** and project **tasks** within their specialty.
  - Managing **multiple projects** within the same team is much more complex than single projects.
  - Managing **both operational and project efforts** – of **varying priorities** – together within a single time is even more complex.
  - Work of any type (tickets or tasks) are effectively **managed as queued, by priority**.
- Teams have engineers of varying **skill levels**.
  - There are generally fewer resources with high skill levels (“senior engineers”).
  - There are usually more with medium and lower levels of skill (“engineers” and “associate engineers”).
  - This varies by team – actual team structures are used in the models.
- The models are intended to reflect the flow of work over relatively **short timelines**, so there is **no ability to flex staff** through hiring (or contract outsourcing).

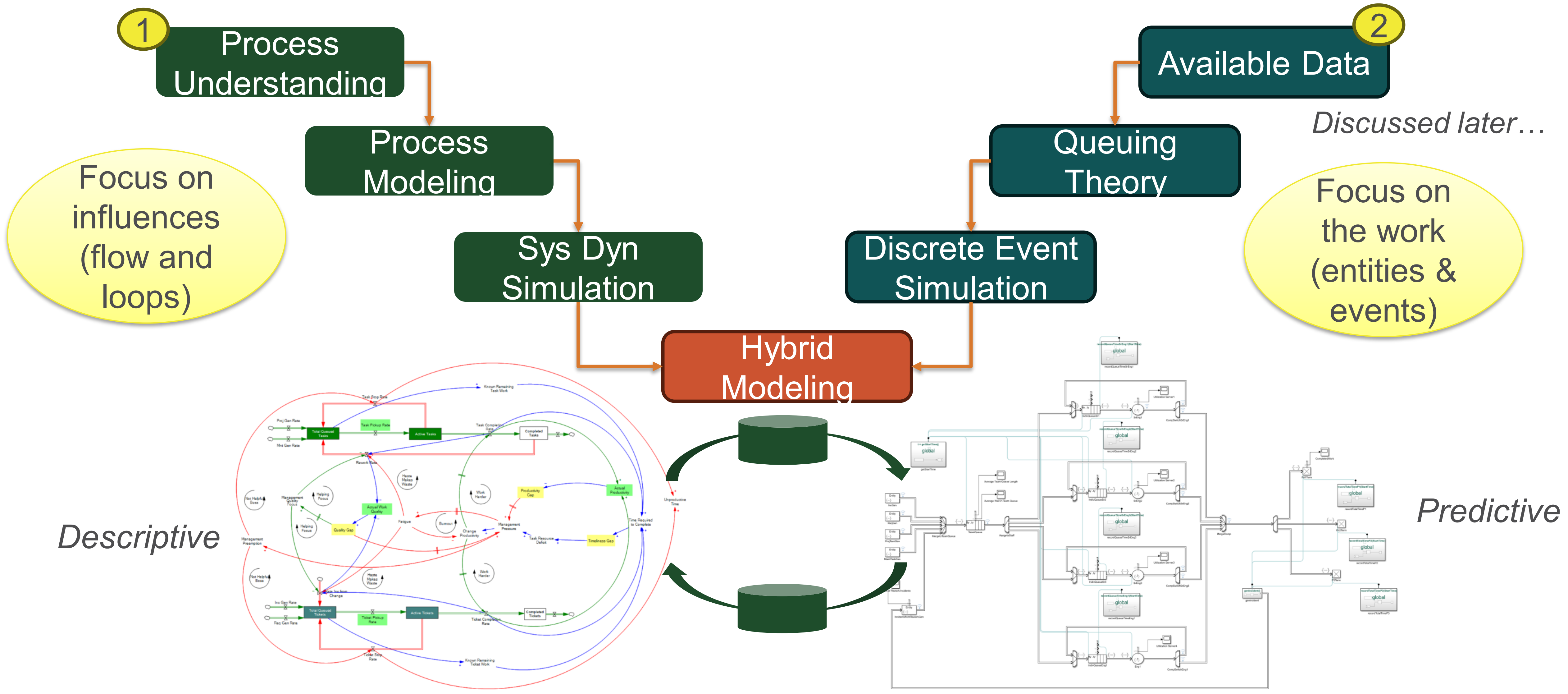
- A. P. Van Der Merwe, “Multi-project management—organizational structure and control,” International Journal of Project Management, vol. 15, no. 4, pp. 223–233, Aug. 1997, doi: 10.1016/S0263-7863(96)00075-0.
- C. Kang and Y. S. Hong, “Evaluation of Acceleration Effect of Dynamic Sequencing of Design Process in a Multiproject Environment,” Journal of Mechanical Design, vol. 131, no. 2, Jan. 2009, doi: 10.1115/1.3066599.
- S. Fricke and A. Shenbar, “Managing multiple engineering projects in a manufacturing support environment,” IEEE Transactions on Engineering Management, vol. 47, no. 2, pp. 258–268, May 2000, conference Name: IEEE Transactions on Engineering Management. doi:10.1109/17.846792.

Basic Work Characteristics

- Work items (tickets and tasks) can **require** one or more resources with a certain **level** and **type** of technical **skill**.
- Work items generally represent <5 hours of work *effort* (distinct from total *duration*)
  - Staff may not have sufficient service time available to complete a ticket or task, resulting in ***re-queuing*** and **switching costs**.
  - Available service time can change based on **re-prioritization** (**escalating** / **expediting**) of other work.
- Work items have a **priority** (low, medium or high) which **can change** over time.
  - Project tasks become more time-sensitive as they get close to – or past – their due date, especially for those on the critical path (“**timeliness**”).
  - Operational tickets are more likely to be escalated the longer they’re open (“**responsiveness**”).
- Quality – or the lack of it - is reflected through the creation of **rework** (in the case of tasks) and **incidents** (in the case of tickets).
  - Probability for errors is initialized based on existing data but can change over time due to dynamic influences.
  - Actual quality is influenced by a difference between the required skill level and the ability of the assigned resource.

- J. M. Lyneis and D. N. Ford, “System dynamics applied to project management: a survey, assessment, and directions for future research,” System Dynamics Review, vol. 23, no. 2-3, pp. 157–189, 2007, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sdr.377. doi: 10.1002/sdr.377.
- J. Wiik and K.-P. Kossakowski, “Dynamics of Incident Response,” Apr. 2017, url: https://www.first.org/resources/papers/2005.

Why Hybrid Modeling?



Identifying the improvement opportunities

- Find where opportunities exist and degree of benefit:
  - Fit available data to determine the distribution of work arrival times using Palisade **@Risk**.
  - Use Discrete Event Simulation (**DES**) – predict the performance of work queuing in terms of total time, with prioritization by ticket or task using **Matlab Simevents** (module of Simulink).
  - Use System Dynamics (**SD**) Simulation – account for influences that affect total work time, especially the impact of managerial pressure on rework and work stoppages driven by re-prioritization using **Vensim**.
- Couple models into a hybrid simulation to obtain benefits of both modeling methods by iterating through each model sequentially.
  - Data sharing currently manual, could be automated.
- Why these tools? In short: **cost**, **accessibility** and **familiarity**.
  - Note: one commercial tool can handle all these functions and would ease integration (AnyLogic).

- R. Jonkers and K. E. Shahroudi, “Connecting Systems Science, Thinking and Engineering to Systems Practice for Managing Complexity,” Journal of the International Society for the Systems Sciences, vol. 65, no. 1, 2021, number: 1. url: https: //journals.isss.org/index.php/jisss/article/view/3875
- B. K. Choi, D. Kang, and B. K. Choi, Modeling and Simulation of Discrete Event Systems. Somerset, US: John Wiley & Sons, Incorporated, 2013, url: http://ebookcentral.proquest.com/lib/csu/detail.action?docID=1402564
- K. Chahal, T. Eldabi, and T. Young, “A conceptual framework for hybrid system dynamics and discrete event simulation for healthcare,” Journal of Enterprise Information Management, vol. 26, no. 1/2, pp. 50–74, Jan. 2013, publisher: Emerald Group Publishing Limited. doi: 10.1108/17410391311289541

Managing the Data

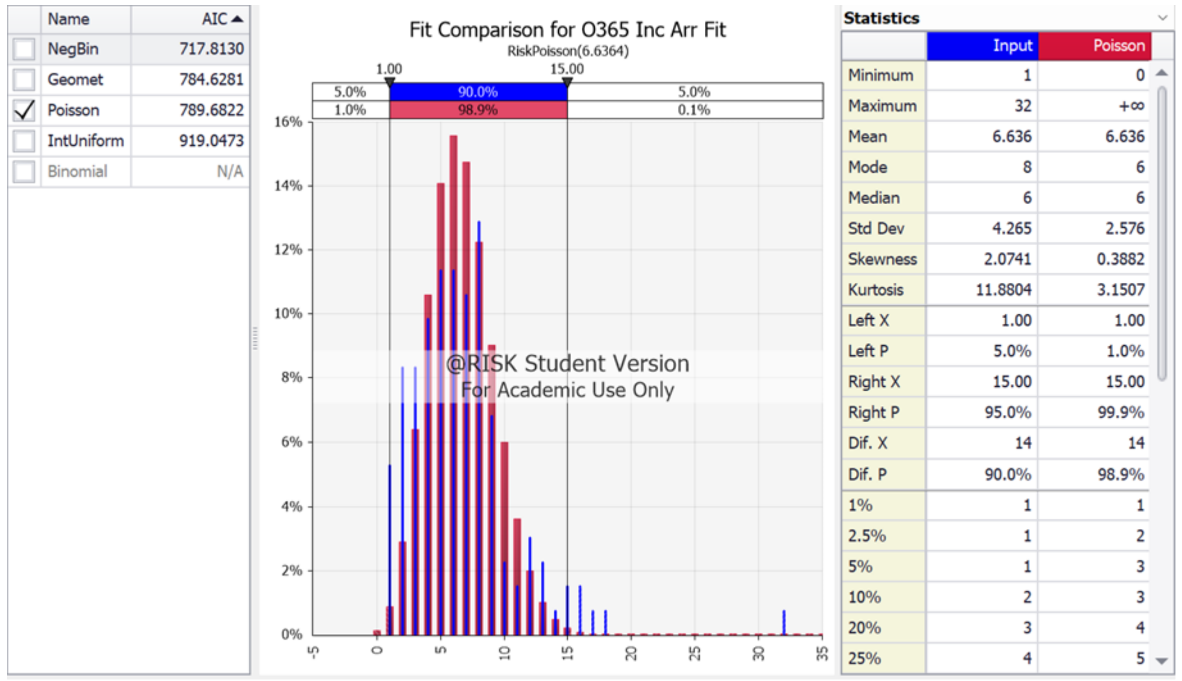
Tool-based data extract and estimates

Available Work Performance Data

- Operational requests and incidents are managed in **ServiceNow**.
  - Data gathered by team, ticket type and priority for model for six months, from 1/1/23 to 6/30/23 including both **arrival** and **closure** dates.
  - Arrival data fit as unique Poisson distributions by team using Palisade @ Risk for use in the DES models as inter-arrival data.
  - Closure data will be used in verification and validation of the DES results.
  - **Priority** breakdowns are based on actual data by team.
    - In the case of the team analyzed: High 0.5%, Medium 40%, Low 59.5%.

Fitting Existing Operational Data

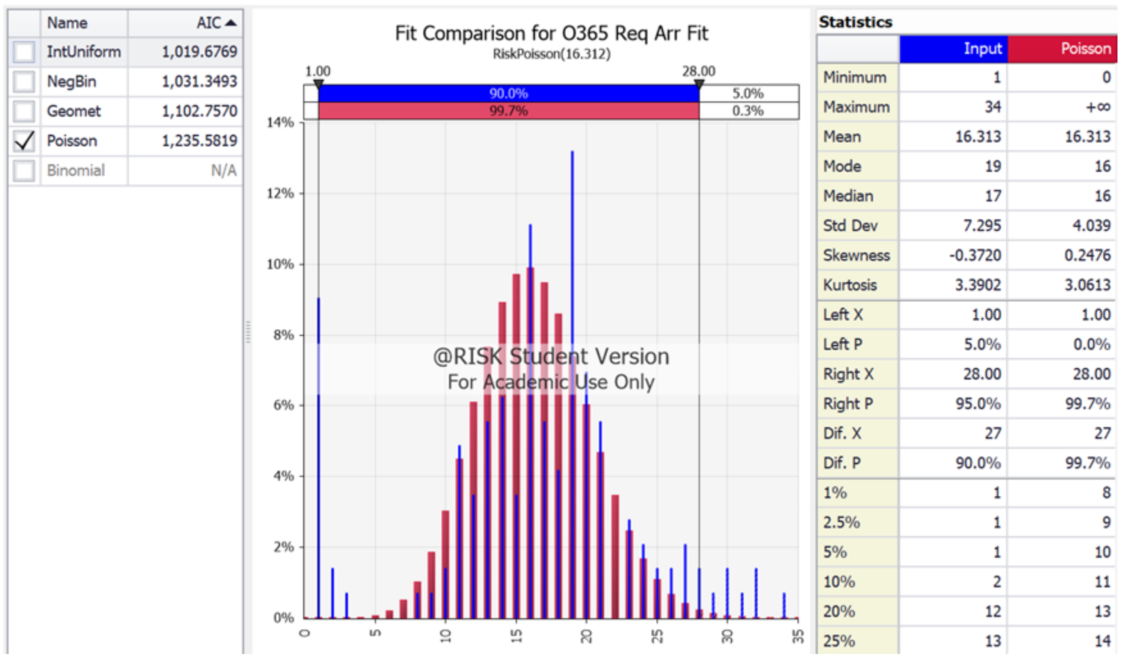
- **Poisson distribution** selected for fit of arrival times as this directly translates into **exponential random inter-arrival times** in DES, defined through Palisade @Risk (note: could be done in Matlab if needed repetitively).
- Example: Incident arrival fitted Poisson distribution mean of 6.6364 yields lambda of 0.150685, resulting in an entity generation equation of dt = -0.150685*log(1-rand()) in SimEvents.



	Name	AIC
☐	NegBin	717.8130
☐	Geomet	784.6281
✓	Poisson	789.6822
☐	IntUniform	919.0473
☐	Binomial	N/A

Statistics

	Input	Poisson
Minimum	1	0
Maximum	32	+∞
Mean	6.636	6.636
Mode	8	6
Median	6	6
Std Dev	4.265	2.576
Skewness	2.0741	0.3882
Kurtosis	11.8804	3.1507
Left X	1.00	1.00
Left P	5.0%	1.0%
Right X	15.00	15.00
Right P	95.0%	99.9%
Dif. X	14	14
Dif. P	90.0%	98.9%
1%	1	1
2.5%	1	2
5%	1	3
10%	2	3
20%	3	4
25%	4	5



	Name	AIC
☐	IntUniform	1,019.6769
☐	NegBin	1,031.3493
☐	Geomet	1,102.7570
✓	Poisson	1,235.5819
☐	Binomial	N/A

Statistics

	Input	Poisson
Minimum	1	0
Maximum	34	+∞
Mean	16.313	16.313
Mode	19	16
Median	17	16
Std Dev	7.295	4.039
Skewness	-0.3720	0.2476
Kurtosis	3.3902	3.0613
Left X	1.00	1.00
Left P	5.0%	0.0%
Right X	28.00	28.00
Right P	95.0%	99.7%
Dif. X	27	27
Dif. P	90.0%	99.7%
1%	1	8
2.5%	1	9
5%	1	10
10%	2	11
20%	12	13
25%	13	14

Estimated Work Performance Data

- Data determined through interviews with department managers due to the lack of reportable data in existing systems:
  - Scheduled tasks also use exponential inter-arrival times but occur less frequently (and often in bunches in reality).
    - Lambda of .5 for project tasks and 1 for maintenance tasks.
  - **Error rates** are initialized on entity generation with a probability of 0.5%.
  - **Service Times**:
    - *Required* Service Time, est. as exponential (lambda of .15), calculated on entity generation.
    - *Actual* Service Time, est. as exponential (lambda of .3), calculated as entity is "picked up" for work by an engineer.
  - Required **skill level**, set on *entity* (work item) generation: High 5%, Medium 25%, Low 75%.
    - Actual skill level is set at each *server* (staff engineer) in the DES model.
    - Modification of error probability and required service times, based on skill level data (discussed on next slide).
- Parameters requiring further investigation / justification:
  - Effects of managerial **pressure** and **fatigue**.
  - Initial **error** and **stoppage** rates.
  - Tuned to fit observation initially.

Dynamic Parameters

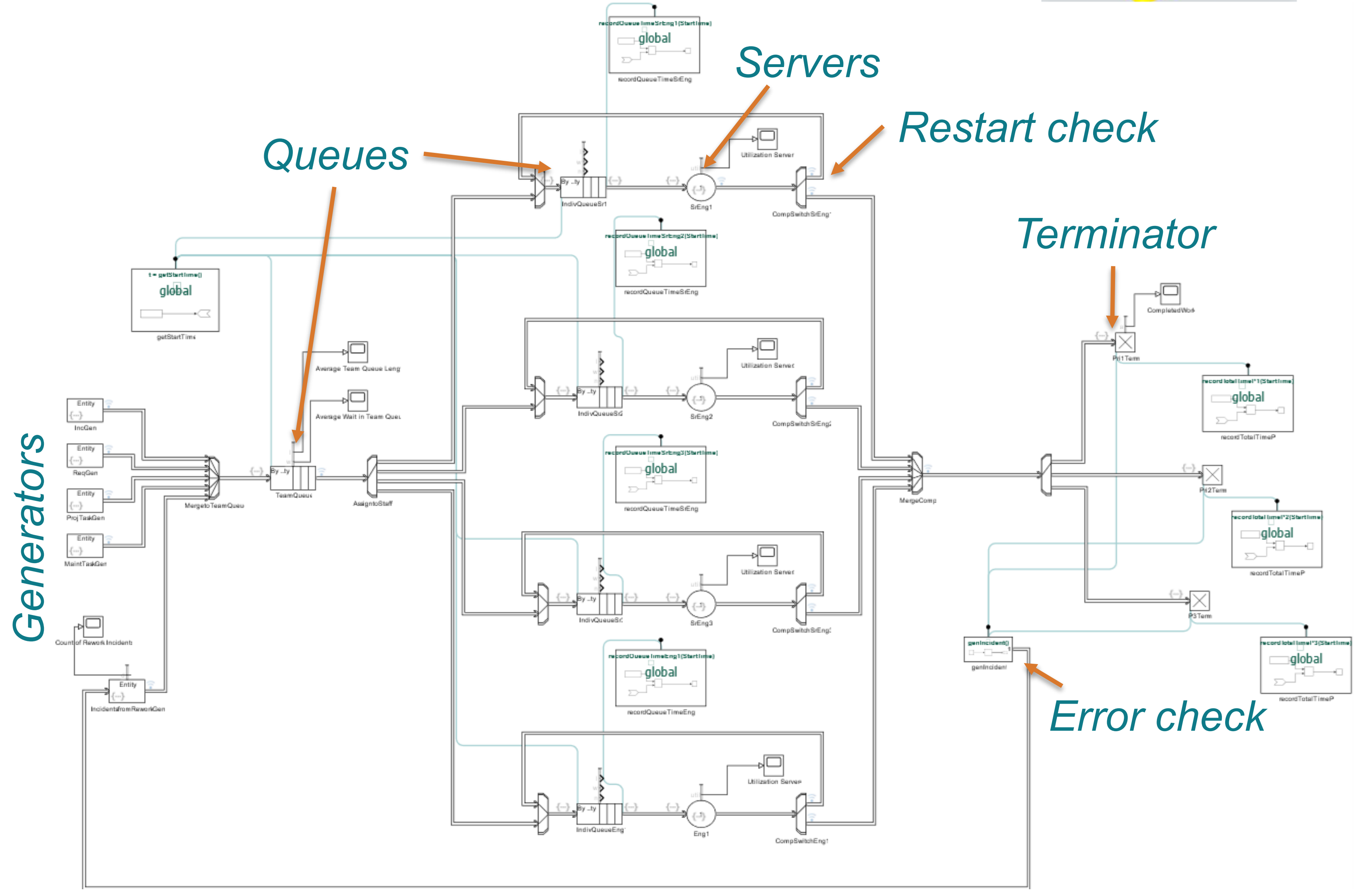
- Work **stoppage** rates:
  - An entity with remaining "Required Service Time" is returned to the queue.
- **Error** generation rates:
  - Initial .5% chance of generating an error (either rework or new incident), set on entity generation.
  - Modified by *difference between required* skill level and *assigned* skill level by rule. Examples:
    - If the "skill difference" is 2 (a high skill level resource with a configured skill level of "3" working on a low skill task with required skill of "1") the rework probability drops to .3%.
    - With a "skill difference" of -2 (a low skill resource working a high skill task) the rework probability increases to 1%.
- Required service time also increases by rule based on **skill level differences**. Examples:
  - A high skill resource working on a low skill task reduces the required service time (set on entity creation) by 20%.
  - A low skill resource working a high skill task increase the required service time by 20%.

Building and Iterating the Models

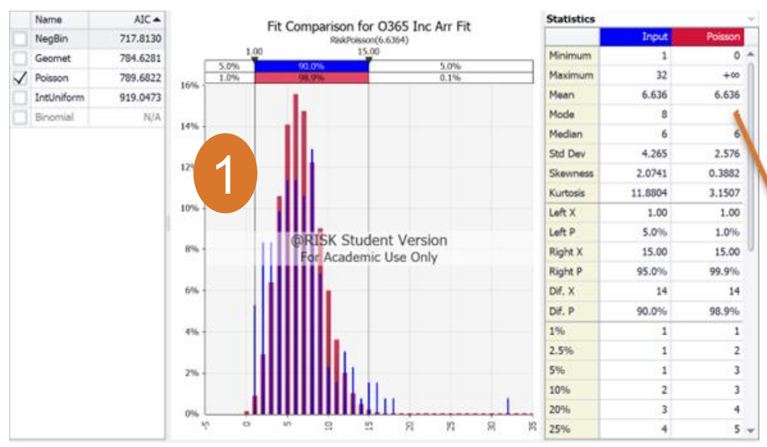
Modeling the process and feeding in the data

Single-Team DES Model

- Five work **generators** with separate inter-arrival functions:
  - Incidents
  - Requests
  - Project tasks
  - Maintenance tasks
  - Incidents from errors / rework
- A single **team queue** with **individual queues** per resource (“**server**”).
- **Restart** and **error generation** functions.
- *NOTE: model does not yet enable managerial escalation of priority while an entity is queued. This may require creation of a custom queue object.*



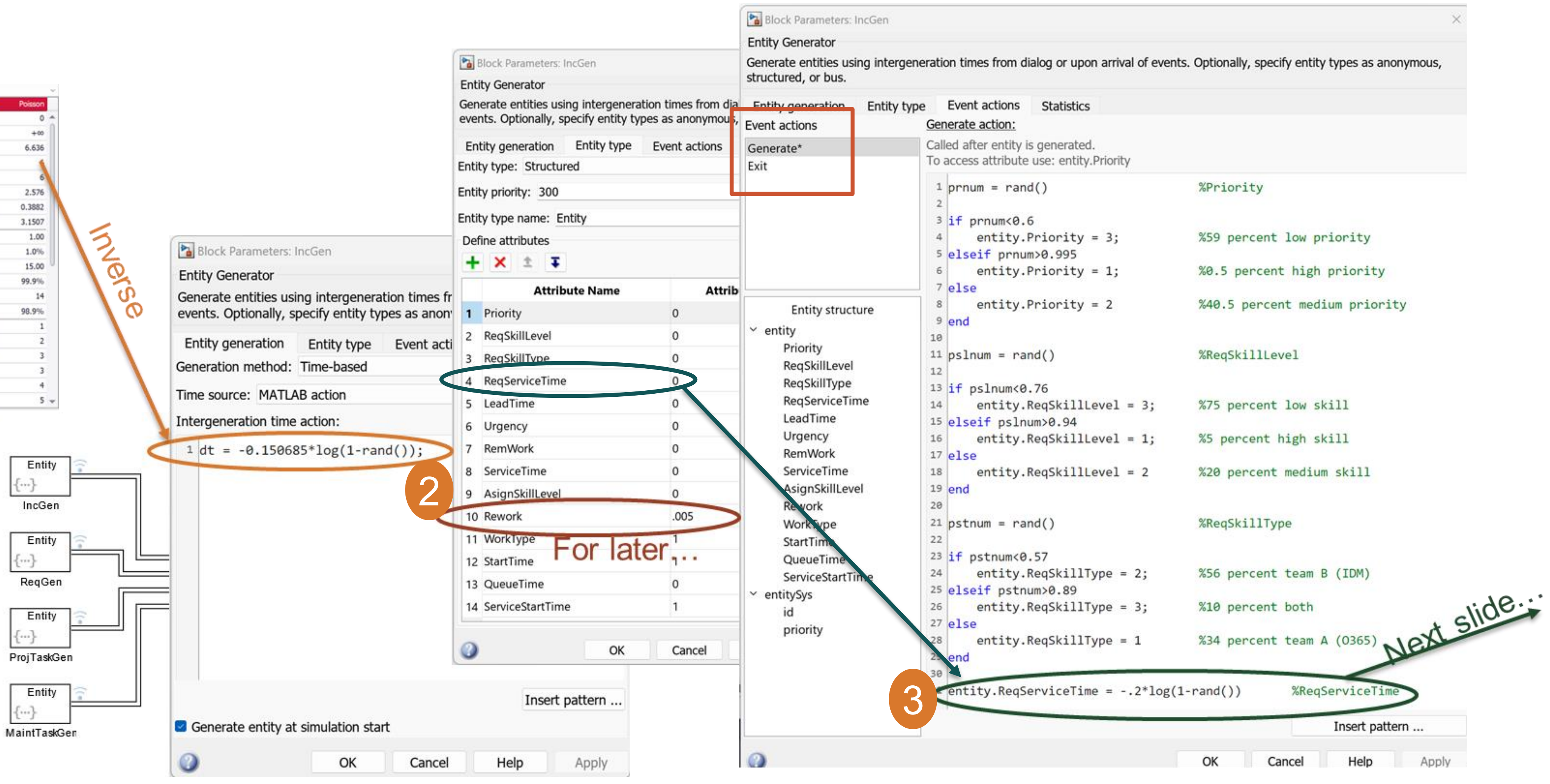
Parameterizing the DES Entity Generators



Name	AIC
NegBin	717.8130
Geomet	784.6281
✓ Poisson	789.6822
IntUniform	919.0473
Binomial	N/A

Statistics

	Input	Poisson
Minimum	1	0
Maximum	32	+∞
Mean	6.636	6.636
Mode	8	6
Median	6	6
Std Dev	4.265	2.576
Skewness	2.0741	0.3882
Kurtosis	11.8804	3.1507
Left X	1.00	1.00
Left P	5.0%	1.0%
Right X	15.00	15.00
Right P	95.0%	99.9%
Dif. X	14	14
Dif. P	90.0%	98.9%
1%	1	1
2.5%	1	2
5%	1	3
10%	2	3
20%	3	4
25%	4	5



Block Parameters: IncGen

Entity Generator

Generate entities using intergeneration times fr… events. Optionally, specify entity types as anon…

Entity generation | Entity type | Event acti…

Generation method: Time-based

Time source: MATLAB action

Intergeneration time action:

```
dt = -0.150685*log(1-rand());
```

☑ Generate entity at simulation start

Block Parameters: IncGen

Entity Generator

Generate entities using intergeneration times from dia… events. Optionally, specify entity types as anonymous…

Entity generation | Entity type | Event actions

Entity type: Structured

Entity priority: 300

Entity type name: Entity

Define attributes

	Attribute Name	Attrib…
1	Priority	0
2	ReqSkillLevel	0
3	ReqSkillType	0
4	ReqServiceTime	0
5	LeadTime	0
6	Urgency	0
7	RemWork	0
8	ServiceTime	0
9	AsignSkillLevel	0
10	Rework	.005
11	WorkType	1
12	StartTime	
13	QueueTime	0
14	ServiceStartTime	1

Block Parameters: IncGen

Entity Generator

Generate entities using intergeneration times from dialog or upon arrival of events. Optionally, specify entity types as anonymous, structured, or bus.

Entity generation | Entity type | Event actions | Statistics

Event actions: Generate*, Exit

Generate action:

Called after entity is generated.
To access attribute use: entity.Priority

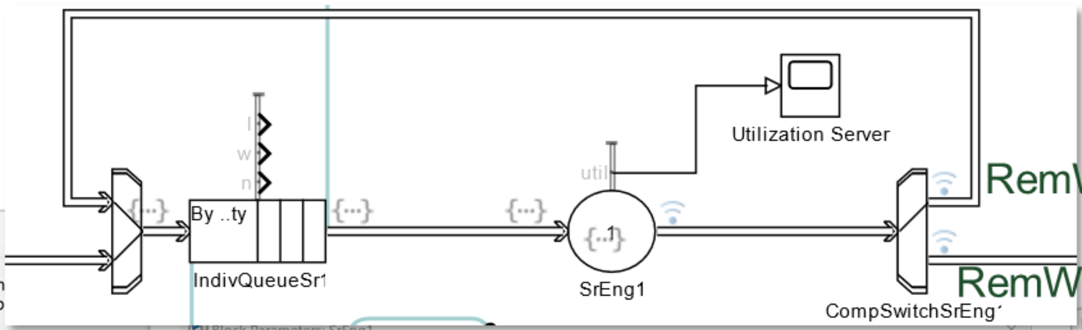
```
prnum = rand()                        %Priority

if prnum<0.6
    entity.Priority = 3;              %59 percent low priority
elseif prnum>0.995
    entity.Priority = 1;              %0.5 percent high priority
else
    entity.Priority = 2               %40.5 percent medium priority
end

pslnum = rand()                       %ReqSkillLevel

if pslnum<0.76
    entity.ReqSkillLevel = 3;         %75 percent low skill
elseif pslnum>0.94
    entity.ReqSkillLevel = 1;         %5 percent high skill
else
    entity.ReqSkillLevel = 2          %20 percent medium skill
end

pstnum = rand()                       %ReqSkillType

if pstnum<0.57
    entity.ReqSkillType = 2;          %56 percent team B (IDM)
elseif pstnum>0.89
    entity.ReqSkillType = 3;          %10 percent both
else
    entity.ReqSkillType = 1           %34 percent team A (O365)
end

entity.ReqServiceTime = -.2*log(1-rand())        %ReqServiceTime
```

Entity structure
- entity
  - Priority
  - ReqSkillLevel
  - ReqSkillType
  - ReqServiceTime
  - LeadTime
  - Urgency
  - RemWork
  - ServiceTime
  - AsignSkillLevel
  - Rework
  - WorkType
  - StartTime
  - QueueTime
  - ServiceStartTime
- entitySys
  - id
  - priority

1) Fitted distribution into the Entity Generator
2) Entity generation initializes parameters
3) Script updates parameters on certain events (in this case, “**Generate**”)

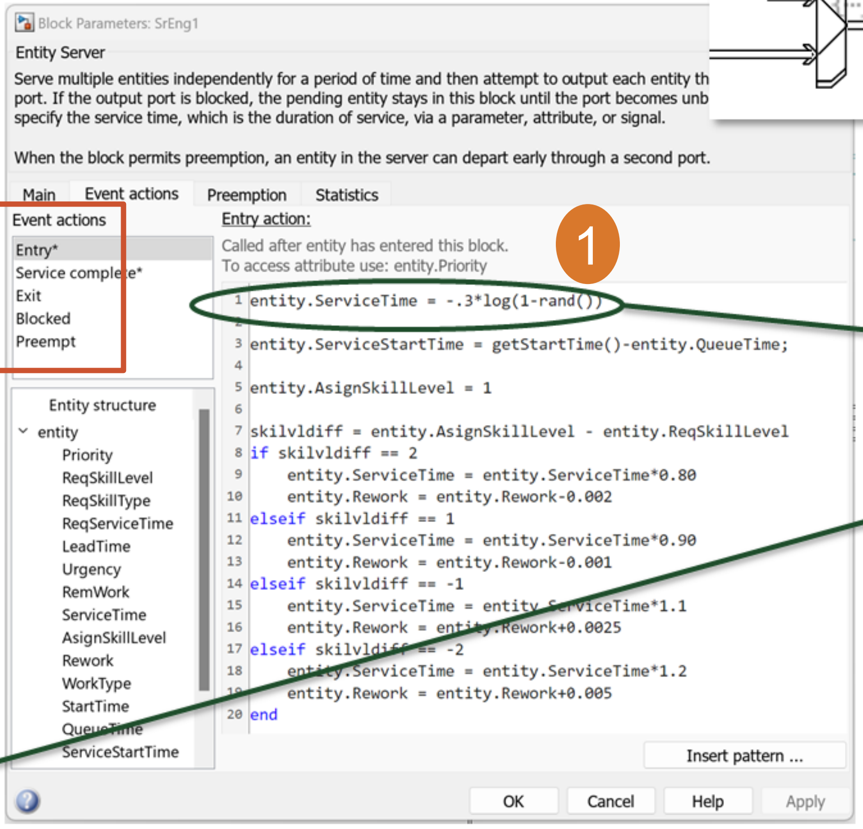
Routing Entities through Attributes

1) Resource **service time** set randomly on **Entry**
2) **Available** service time compared to **Required**; if incomplete (“**RemWork**” >0), work is returned to the queue to complete



RemWork=1 -> Port 1

RemWork=2 -> Port 2



Entity Server

Serve multiple entities independently for a period of time and then attempt to output each entity th
port. If the output port is blocked, the pending entity stays in this block until the port becomes unb
specify the service time, which is the duration of service, via a parameter, attribute, or signal.

When the block permits preemption, an entity in the server can depart early through a second port.

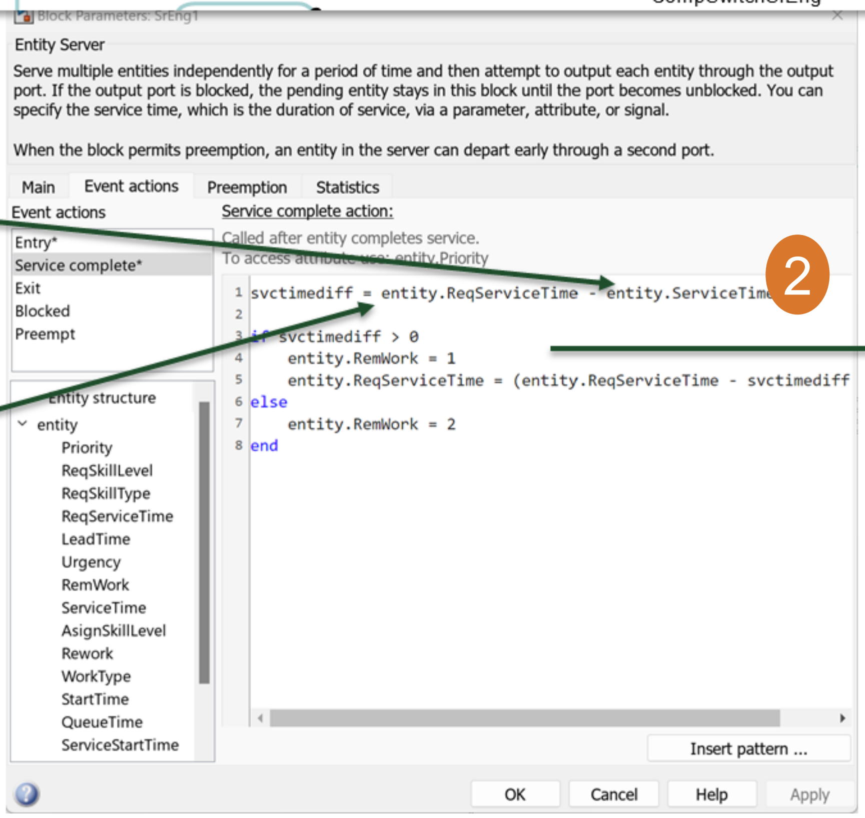
Event actions: Entry*, Service complete*, Exit, Blocked, Preempt

Entity structure: entity — Priority, ReqSkillLevel, ReqSkillType, ReqServiceTime, LeadTime, Urgency, RemWork, ServiceTime, AsignSkillLevel, Rework, WorkType, StartTime, QueueTime, ServiceStartTime

Entry action:

Called after entity has entered this block.
To access attribute use: entity.Priority

```
entity.ServiceTime = -.3*log(1-rand())

entity.ServiceStartTime = getStartTime()-entity.QueueTime;

entity.AsignSkillLevel = 1

skilvldiff = entity.AsignSkillLevel - entity.ReqSkillLevel
if skilvldiff == 2
    entity.ServiceTime = entity.ServiceTime*0.80
    entity.Rework = entity.Rework-0.002
elseif skilvldiff == 1
    entity.ServiceTime = entity.ServiceTime*0.90
    entity.Rework = entity.Rework-0.001
elseif skilvldiff == -1
    entity.ServiceTime = entity.ServiceTime*1.1
    entity.Rework = entity.Rework+0.0025
elseif skilvldiff == -2
    entity.ServiceTime = entity.ServiceTime*1.2
    entity.Rework = entity.Rework+0.005
end
```



Entity Server

Serve multiple entities independently for a period of time and then attempt to output each entity through the output port. If the output port is blocked, the pending entity stays in this block until the port becomes unblocked. You can specify the service time, which is the duration of service, via a parameter, attribute, or signal.

When the block permits preemption, an entity in the server can depart early through a second port.

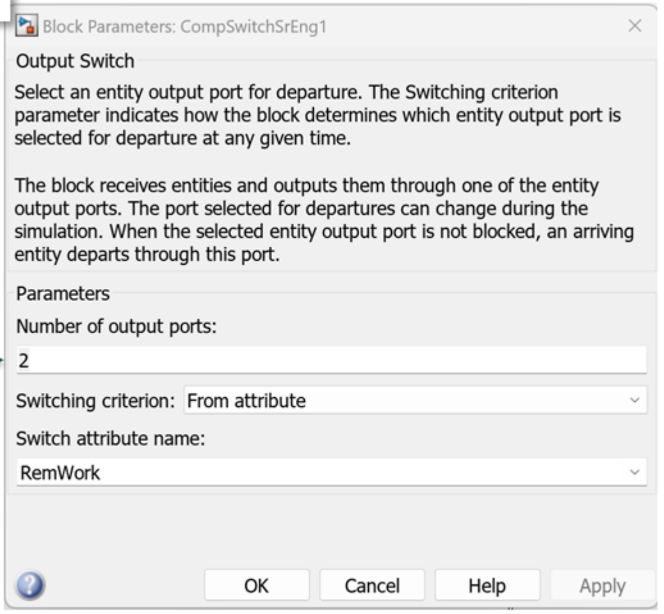
Event actions: Entry*, Service complete*, Exit, Blocked, Preempt

Entity structure: entity — Priority, ReqSkillLevel, ReqSkillType, ReqServiceTime, LeadTime, Urgency, RemWork, ServiceTime, AsignSkillLevel, Rework, WorkType, StartTime, QueueTime, ServiceStartTime

Service complete action:

Called after entity completes service.
To access attribute use: entity.Priority

```
svctimediff = entity.ReqServiceTime - entity.ServiceTime

if svctimediff > 0
    entity.RemWork = 1
    entity.ReqServiceTime = (entity.ReqServiceTime - svctimediff
else
    entity.RemWork = 2
end
```



Output Switch

Select an entity output port for departure. The Switching criterion parameter indicates how the block determines which entity output port is selected for departure at any given time.

The block receives entities and outputs them through one of the entity output ports. The port selected for departures can change during the simulation. When the selected entity output port is not blocked, an arriving entity departs through this port.

Parameters

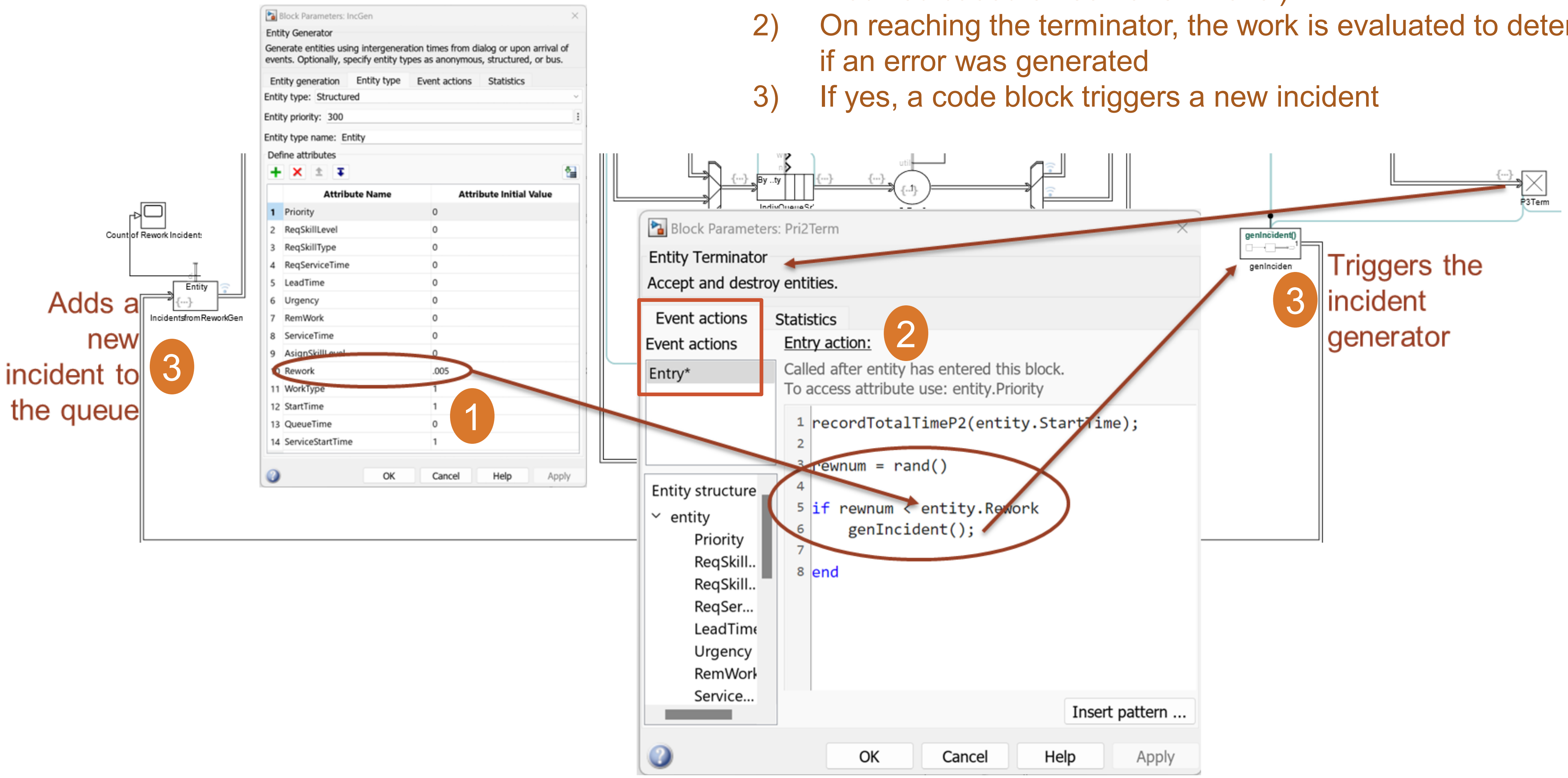
Number of output ports: 2

Switching criterion: From attribute

Switch attribute name: RemWork

Error Generation

1) Probability of **Rework** initialized to .5% on **Generation** (later modified based on server skill level)
2) On reaching the terminator, the work is evaluated to determine if an error was generated
3) If yes, a code block triggers a new incident



Count of Rework Incidents
Entity
IncidentsfromReworkGen
By ...ty
IndivQueueSr
P3Term
genIncident()
genInciden
Adds a new incident to the queue
3
1
2
3
Triggers the incident generator

Block Parameters: IncGen

Entity Generator

Generate entities using intergeneration times from dialog or upon arrival of events. Optionally, specify entity types as anonymous, structured, or bus.

Entity generation | Entity type | Event actions | Statistics

Entity type: Structured

Entity priority: 300

Entity type name: Entity

Define attributes

	Attribute Name	Attribute Initial Value
1	Priority	0
2	ReqSkillLevel	0
3	ReqSkillType	0
4	ReqServiceTime	0
5	LeadTime	0
6	Urgency	0
7	RemWork	0
8	ServiceTime	0
9	AsignSkillLevel	0
10	Rework	.005
11	WorkType	1
12	StartTime	1
13	QueueTime	0
14	ServiceStartTime	1

OK Cancel Help Apply

Block Parameters: Pri2Term

Entity Terminator

Accept and destroy entities.

Event actions | Statistics

Event actions

Entry*

Entry action:

Called after entity has entered this block.
To access attribute use: entity.Priority

```
recordTotalTimeP2(entity.StartTime);

rewnum = rand()

if rewnum < entity.Rework
    genIncident();

end
```

Entity structure
- entity
  - Priority
  - ReqSkill..
  - ReqSkill..
  - ReqSer...
  - LeadTime
  - Urgency
  - RemWork
  - Service...

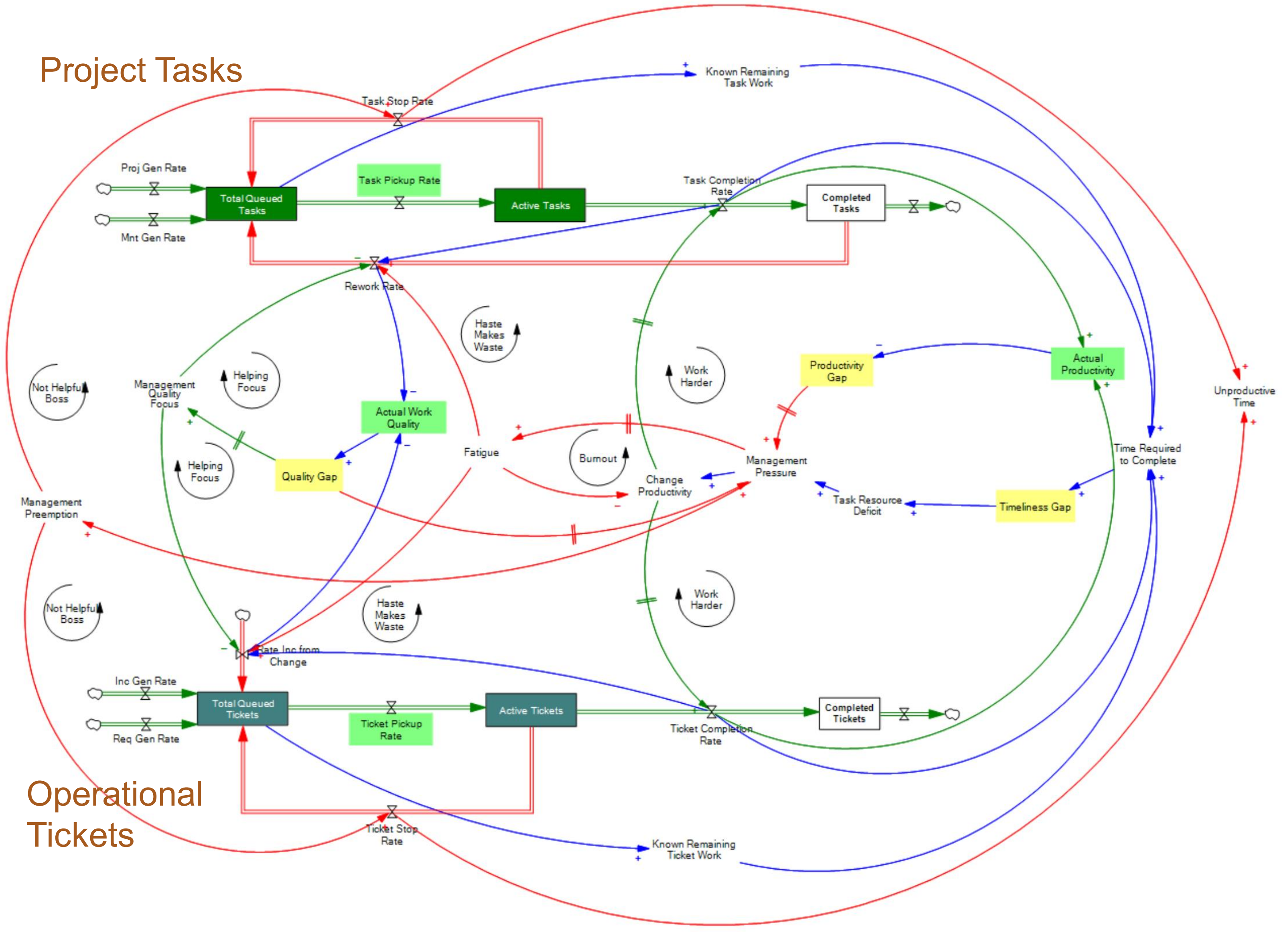
Insert pattern ...

OK Cancel Help Apply

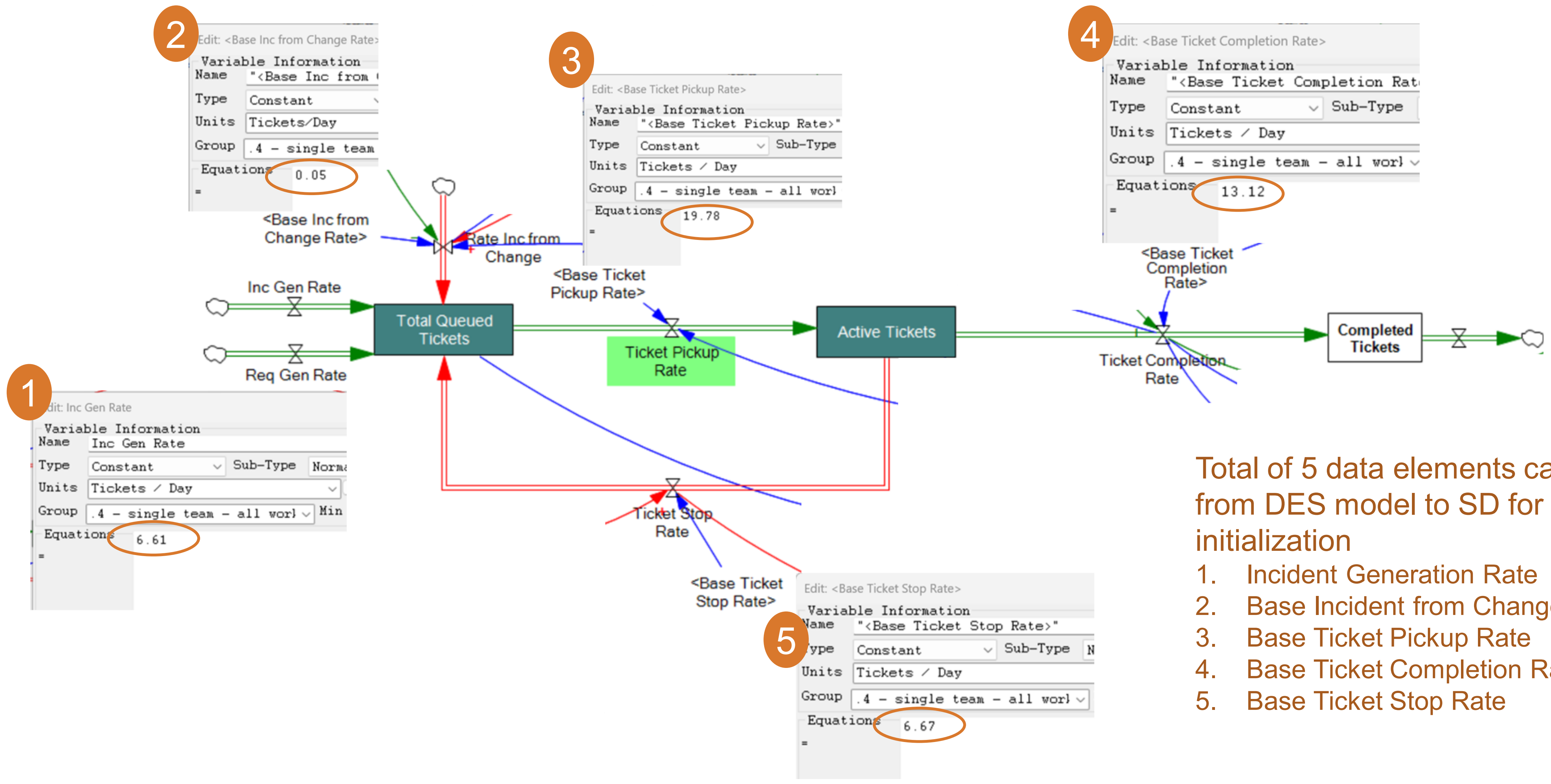
Single-Team SD Model

Vensim
VENTANA systems inc

- Modeled as two parallel flows of work:
  - Enable future differences between tasks and tickets in modeling of *timeliness* vs. *responsiveness*.
  - Enable differences in handling of rework vs. incidents resulting from operational changes.
- Influencing factors affecting both workflows shown in the center of the model:
  - **Gaps** in **quality**, **productivity** and **timeliness**.
  - Managerial **pressure** and **quality focus** as well as effects of **fatigue**.
- Note: internal model parameters not shown in this top-level diagram.



Initializing the SD Model from DES Data



Edit: <Base Inc from Change Rate>
Variable Information
Name "<Base Inc from
Type Constant
Units Tickets/Day
Group .4 - single team
Equations 0.05
=

Edit: <Base Ticket Pickup Rate>
Variable Information
Name "<Base Ticket Pickup Rate>"
Type Constant Sub-Type
Units Tickets / Day
Group .4 - single team - all worl
Equations 19.78
=

Edit: <Base Ticket Completion Rate>
Variable Information
Name "<Base Ticket Completion Rat
Type Constant Sub-Type
Units Tickets / Day
Group .4 - single team - all worl
Equations 13.12
=

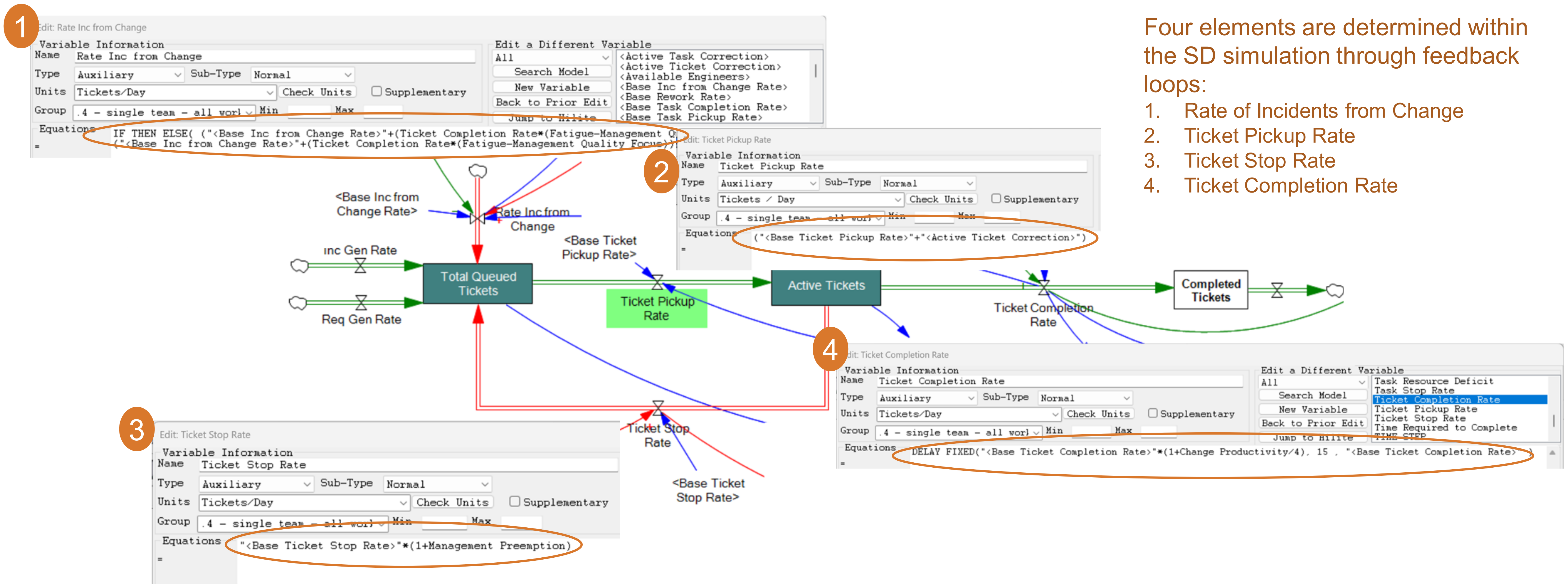
dit: Inc Gen Rate
Variable Information
Name Inc Gen Rate
Type Constant Sub-Type Norma
Units Tickets / Day
Group .4 - single team - all worl Min
Equations 6.61
=

Edit: <Base Ticket Stop Rate>
Variable Information
Name "<Base Ticket Stop Rate>"
Type Constant Sub-Type N
Units Tickets / Day
Group .4 - single team - all worl
Equations 6.67
=

Total of 5 data elements carried from DES model to SD for initialization

1. Incident Generation Rate
2. Base Incident from Change Rate
3. Base Ticket Pickup Rate
4. Base Ticket Completion Rate
5. Base Ticket Stop Rate

Parameterizing SD Behavior



Edit: Rate Inc from Change

Variable Information
Name: Rate Inc from Change
Type: Auxiliary — Sub-Type: Normal
Units: Tickets/Day — Check Units — ☐ Supplementary
Group: .4 - single team - all worl — Min — Max

Edit a Different Variable: All — Search Model — New Variable — Back to Prior Edit — Jump to Hilite
<Active Task Correction>
<Active Ticket Correction>
<Available Engineers>
<Base Inc from Change Rate>
<Base Rework Rate>
<Base Task Completion Rate>
<Base Task Pickup Rate>

```
Equations = IF THEN ELSE( ("<Base Inc from Change Rate>"+(Ticket Completion Rate*(Fatigue-Management Q
("<Base Inc from Change Rate>"+(Ticket Completion Rate*(Fatigue-Management Quality Focus))
```

Edit: Ticket Pickup Rate

Variable Information
Name: Ticket Pickup Rate
Type: Auxiliary — Sub-Type: Normal
Units: Tickets / Day — Check Units — ☐ Supplementary
Group: .4 - single team - all worl — Min — Max

```
Equations = ("<Base Ticket Pickup Rate>"+"<Active Ticket Correction>")
```

Edit: Ticket Stop Rate

Variable Information
Name: Ticket Stop Rate
Type: Auxiliary — Sub-Type: Normal
Units: Tickets/Day — Check Units — ☐ Supplementary
Group: .4 - single team - all worl — Min — Max

```
Equations = "<Base Ticket Stop Rate>"*(1+Management Preemption)
```

Edit: Ticket Completion Rate

Variable Information
Name: Ticket Completion Rate
Type: Auxiliary — Sub-Type: Normal
Units: Tickets/Day — Check Units — ☐ Supplementary
Group: .4 - single team - all worl — Min — Max

Edit a Different Variable: All — Search Model — New Variable — Back to Prior Edit — Jump to Hilite
Task Resource Deficit
Task Stop Rate
Ticket Completion Rate
Ticket Pickup Rate
Ticket Stop Rate
Time Required to Complete
TIME STEP

```
Equations = DELAY FIXED("<Base Ticket Completion Rate>"*(1+Change Productivity/4), 15 , "<Base Ticket Completion Rate>" )
```

Four elements are determined within the SD simulation through feedback loops:

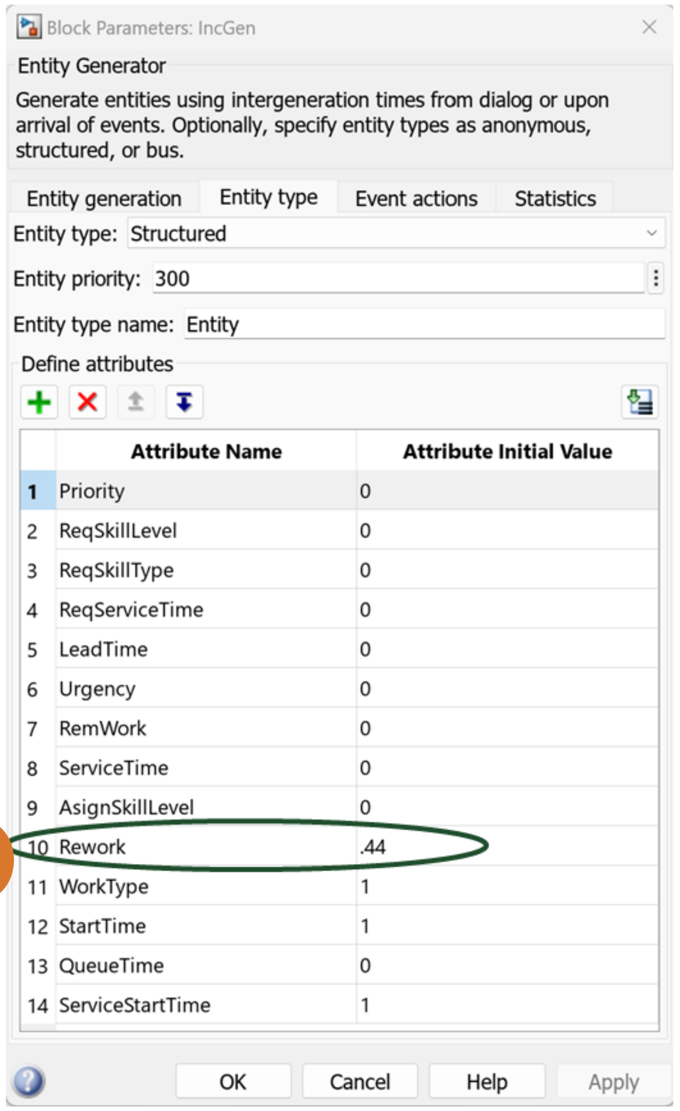
1. Rate of Incidents from Change
2. Ticket Pickup Rate
3. Ticket Stop Rate
4. Ticket Completion Rate

Iteration 2: Feed SD data back to DES Model

Two data elements fed from SD back to DES model during iteration

1. Rework Probability
2. Service Time

Captures increase in error generation due to management pressure



Block Parameters: IncGen

Entity Generator

Generate entities using intergeneration times from dialog or upon arrival of events. Optionally, specify entity types as anonymous, structured, or bus.

Entity generation | Entity type | Event actions | Statistics

Entity type: Structured

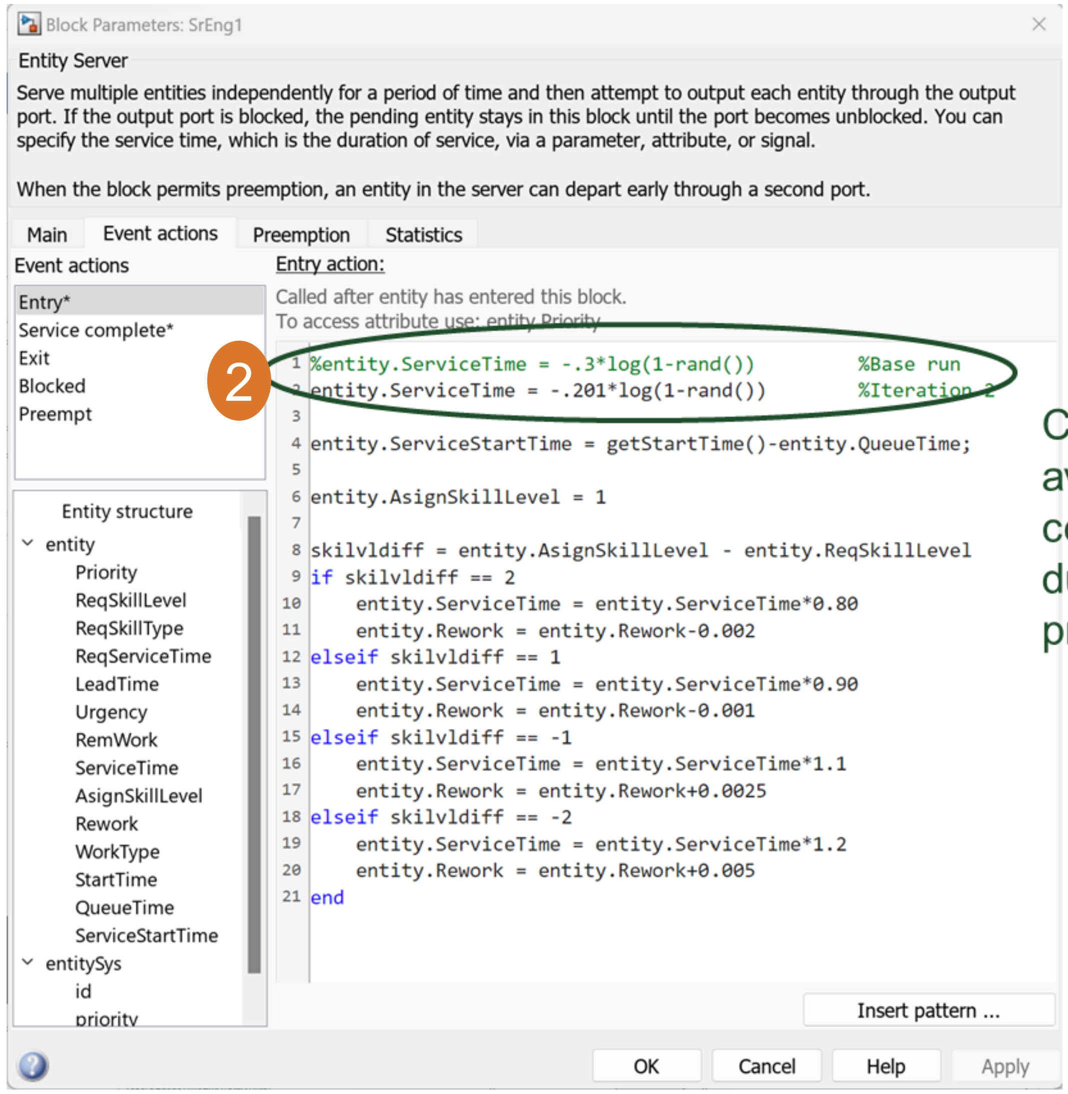
Entity priority: 300

Entity type name: Entity

Define attributes

	Attribute Name	Attribute Initial Value
1	Priority	0
2	ReqSkillLevel	0
3	ReqSkillType	0
4	ReqServiceTime	0
5	LeadTime	0
6	Urgency	0
7	RemWork	0
8	ServiceTime	0
9	AsignSkillLevel	0
10	Rework	.44
11	WorkType	1
12	StartTime	1
13	QueueTime	0
14	ServiceStartTime	1

OK | Cancel | Help | Apply



Block Parameters: SrEng1

Entity Server

Serve multiple entities independently for a period of time and then attempt to output each entity through the output port. If the output port is blocked, the pending entity stays in this block until the port becomes unblocked. You can specify the service time, which is the duration of service, via a parameter, attribute, or signal.

When the block permits preemption, an entity in the server can depart early through a second port.

Main | Event actions | Preemption | Statistics

Event actions: Entry*, Service complete*, Exit, Blocked, Preempt

Entry action:

Called after entity has entered this block.
To access attribute use: entity.Priority

```
%entity.ServiceTime = -.3*log(1-rand())        %Base run
entity.ServiceTime = -.201*log(1-rand())       %Iteration 2

entity.ServiceStartTime = getStartTime()-entity.QueueTime;

entity.AsignSkillLevel = 1

skilvldiff = entity.AsignSkillLevel - entity.ReqSkillLevel
if skilvldiff == 2
    entity.ServiceTime = entity.ServiceTime*0.80
    entity.Rework = entity.Rework-0.002
elseif skilvldiff == 1
    entity.ServiceTime = entity.ServiceTime*0.90
    entity.Rework = entity.Rework-0.001
elseif skilvldiff == -1
    entity.ServiceTime = entity.ServiceTime*1.1
    entity.Rework = entity.Rework+0.0025
elseif skilvldiff == -2
    entity.ServiceTime = entity.ServiceTime*1.2
    entity.Rework = entity.Rework+0.005
end
```

Entity structure
- entity
  - Priority
  - ReqSkillLevel
  - ReqSkillType
  - ReqServiceTime
  - LeadTime
  - Urgency
  - RemWork
  - ServiceTime
  - AsignSkillLevel
  - Rework
  - WorkType
  - StartTime
  - QueueTime
  - ServiceStartTime
- entitySys
  - id
  - priority

Insert pattern ...

OK | Cancel | Help | Apply

Captures decrease in available time to complete a work item due to management pre-emption

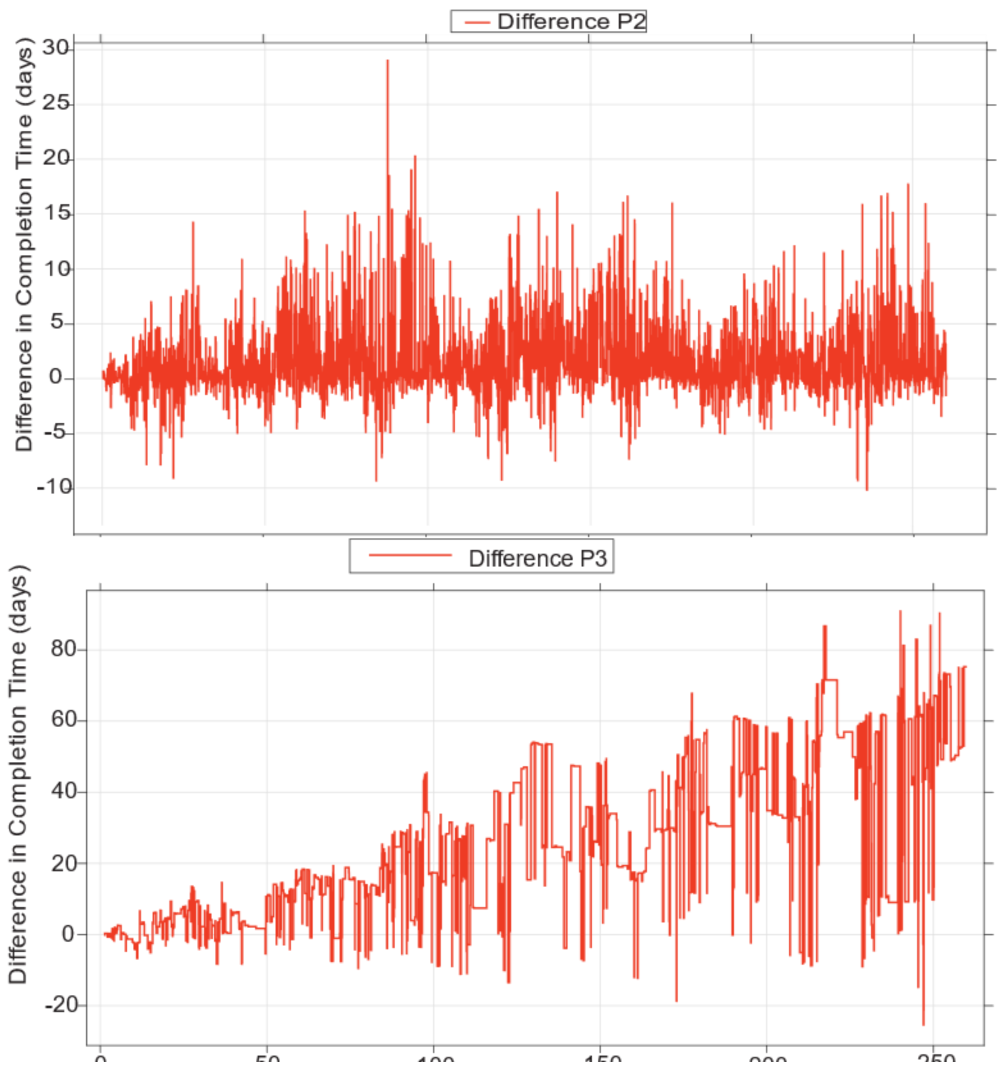
Preliminary Conclusions and Future Research

Expanding the simulations

Results after two iterations

- Key parameters shared between models:
  - Inter-arrival rates.
  - Work pickup and stoppage rates.
  - Error generation rates.
- After a full iteration of the DES and SD model, injecting SD results back into DES results in:
  - Significant **increase in incidents from errors** overall, as **queuing delays** led to **increased managerial pressure**.
  - Minimal to no impact to completion times in **high priority** work.
  - Significant increase in completion times of **medium and low priority work.**

NOTE: Management would intervene differently well before extremes seen at the end of the iteration



Observations from Simulation

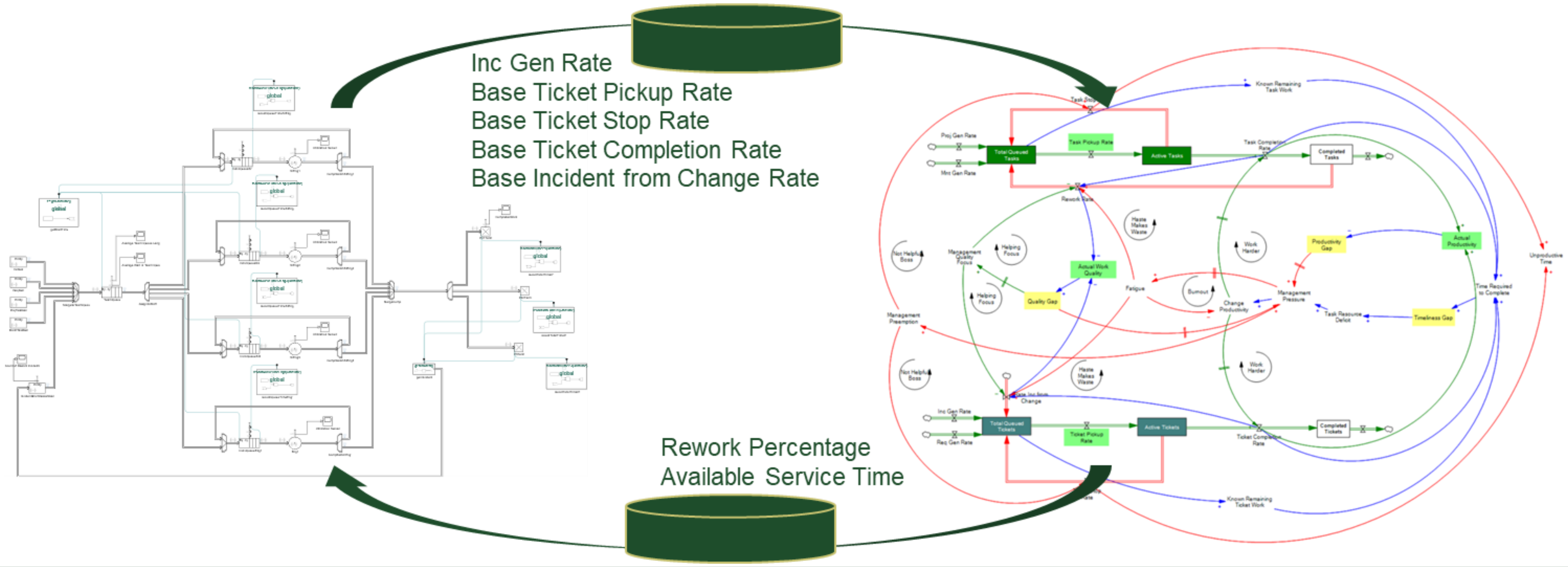
- The team as modeled is **not in control** in either baseline or second iteration:
  - Medium and high priority work items are handled reasonably quickly initially, but low priority items experience significant and growing delays.
  - With the impact of managerial pressure and fatigue added to a second iteration, performance on medium priority work degrades and low priority work delays grow increasingly long.
- 1/3 more work items were **stopped and re-queued due to management pressure**, and there was a very large **increase in Incidents from Rework** (errors).
- This resulted in a 25% increase in work completed in response to incidents (because there were so many more of them due to errors) as well as an **increase in all completion times (**18% for P1s and a 125% increase in P2s).
- For all intents and purposes, by the end of the simulation many **P3s simply remained queued** with 75% less completed during the run.
- These characteristics are **directionally consistent with observations** from the case study team in the real world.

Preliminary Conclusions

- Potential improvement areas that can be influenced by automation:
  - Reducing completion times by **reducing queue times**, either from stop and restart or from transfers.
  - Improving operational responsiveness through **triggering automation on arrival** (e.g., **incident** or **request** arrival).
  - Improving actual work quality / reducing the rate of rework and the generation of new incidents by **reducing or eliminating human error and variability**.
  - **Increasing pickup and completion rates**.
  - NOTE: focus on relatively ***high-volume*** and ***repeatable*** work for automation to be worth the effort.
- Additional options for improvement beyond automation are suggested by the models – and many have been studied previously – but are not the focus of this research:
  - Close coordination between project and functional leaders on work priorities.
  - Limit the number of projects in process through governance and prioritization.
  - Separate operational and project responsibilities between different staff.
  - Hire and (and train) multi-skilled resources.

Limitations of the Models

- Each model iteration currently run for very long periods to indicate accurate **direction**, not **magnitude**.
- To closer approximate reality, feedback between the two models should be ***much more rapid – weekly if not daily***.
- From a practical standpoint this requires ***automation of a test harness*** encompassing both simulation tools to initialize and transfer data outputs from on model / iteration to inputs in the next model / iteration – or the use of a modeling tool that supports both simulation types.



Expanding Simulation to Two Teams

- Adding a second team to the SD and DES models introduces additional rules for managing work passing between them:
  - Each team supports mixed work types – e.g., both unscheduled and scheduled – that arrive at differing rates.
  - Each team supports multiple concurrent projects / products at different stages of planning and execution, as well as multiple concurrent incidents and requests.
  - Active work in progress can be returned to the queue for a new reason: **it can be "reassigned" to another team’s queue** due to a lack of certain technical skill in the originating team.
    - Note that this can happen multiple times with a given work item.
  - **Reassignments** between teams occur at a certain **rate** (determined by the **degree of coupling** between the teams) and require coordination to ensure efficient completion and are an indicator of a higher level of overall work item complexity.
  - The models will only depict interactions between two skill-based teams – interactions become much more complex as additional teams become part of the work process.
- **New dynamics are expected** that could introduce new criteria for identification and prioritization of automation targets.

Ongoing Data Analysis

- The comparison of baseline simulation results to actual team performance data for validation remains in process.
- Research to better justify the internal parameters in the SD model regarding the effects of management pressure and fatigue is needed.
  - To date the literature indicates the *direction* of these effects, but not the *magnitude.*
- Sensitivity analysis of overall completion times to restart and error rates, and upstream influences on those in the SD model, is needed to better inform the prioritization of managerial interventions to improve the process.

Summary

- Hybrid modeling including SD and DES predicts results that neither method would independently, particularly for a non-equilibrium system.
- Higher fidelity results require an automated test harness – or a single tool that supports integration of both methods.
- Target process automation at high-volume, repeatable activities that lead to increased queue time, error rates and switching costs.

Link to the Models

https://www.engr.colostate.edu/~drherber/publication?key=Enos2024a

Questions?

Thank you

WALTER SCOTT, JR.
COLLEGE OF ENGINEERING
COLORADO STATE UNIVERSITY

SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY