Discrete Event Simulation (DEVS) of Strategic Airlift v2

SYSC 5104 – Fall 2019

Prepared By:

Rob Barwell 101092722

**Contents**

[Part 1 – Conceptual Model – Strategic Airlift 3](#_Toc22030564)

[Background 3](#_Toc22030565)

[Model Overview 3](#_Toc22030566)

[Model Assumptions / Constraints 3](#_Toc22030567)

[Model Components 3](#_Toc22030568)

[Model Diagram 4](#_Toc22030569)

[Part 2 – Formal Model Specifications 5](#_Toc22030570)

[Notes 5](#_Toc22030571)

[Atomic Models 5](#_Toc22030572)

[Pallet Generator = <X, Y, S, ta, δext, δint, λ> 5](#_Toc22030573)

[Packing Facility = <X, Y, S, ta, δext, δint, λ> 7](#_Toc22030574)

[Aircraft Loader = <X, Y, S, ta, δext, δint, λ> 9](#_Toc22030575)

[Aircraft = <X, Y, S, ta, δext, δint, λ> 11](#_Toc22030576)

[Destination = <X, Y, S, ta, δext, δint, λ> 13](#_Toc22030577)

[Coupled Models 14](#_Toc22030578)

[Loading Process = <*X*, *Y*, D, {Mi}, IC, EIC, EOC, select> 14](#_Toc22030579)

[Strategic Airlift = <*X*, *Y*, D, {Mi}, IC, EIC, EOC, select> 14](#_Toc22030580)

[Part 3 – Model Experiments 15](#_Toc22030581)

[Notes 15](#_Toc22030582)

[Atomic Experiments 15](#_Toc22030583)

[Pallet Generator (PalletGen\_Test.exe) 15](#_Toc22030584)

[Packing Facility (LoadPacking\_Test.exe) 17](#_Toc22030585)

[Aircraft Loader (AircraftLoader\_Test.exe) 18](#_Toc22030586)

[Aircraft (Aircraft\_Test.exe) 19](#_Toc22030587)

[Destination (Destination\_Test.exe) 20](#_Toc22030588)

[Coupled Experiments 21](#_Toc22030589)

[Loading Process (LoadingProcess\_Test.exe) 21](#_Toc22030590)

[Model Experiments 22](#_Toc22030591)

[Overview 22](#_Toc22030592)

[Experiments 24](#_Toc22030593)

[E1 – Expired Pallet Generation Experiment 24](#_Toc22030594)

[E2 – Pallet Expiration Time Experiment 25](#_Toc22030595)

# Part 1 – Conceptual Model – Strategic Airlift

Copy from other document …

# Part 2 – Formal Model Specifications

## Notes

1. The smallest unit of time in this model is minutes. The EIRational time library was chosen to track time by the minute to support this model. During output, time is converted to hours or days depending on the situation.

## Atomic Models

### Location = <X, Y, S, ta, δext, δint, λ>

|  |  |
| --- | --- |
|  | |
| **State Variables**  Sigma = ∞  Phase = Waiting  Location Name = *Set on initialization*  Location ID = *Set on initialization*  Connections = *Set on initialization*  Waiting Pallets = {} (<id, current location, routing>)  Waiting Aircraft = {} (<id, location, capacity, type, home location, waiting time>)  Waiting Loads = {} (<id, destination, distance, aircraft id, pallets>)  Destination Pallets = {} (<id, current location, routing>)  Processed Loads = {} (<id, destination, distance, aircraft id, pallets>) | **Formal Specifications**  X = {Pallet ∈ <N, N, vector<N>> | Aircraft Status ∈ <N, N, N, N, N, N> | Load ∈ <String, N, N, N, vector<Pallet>>}  Y = {Pallet ∈ <N, N, vector<N>> | Load ∈ <String, N, N, N, vector<Pallet>>}  S = {{Sigma, Phase, Location Name, Location ID, Connections, Waiting Pallets, Waiting Aircraft, Waiting Loads, Destination Pallets, Processed Loads}} |
| δext (mbs, e, x <Pallet | Aircraft Status | Load>)  {  If X is Pallet and destination is this location  Add to waiting pallets queue  If X is Aircraft Status and destination is this location  Add to waiting aircraft queue  Set aircraft wait time to 0  If X is Load and destination is this location  Add to waiting loads  Phase = PROCESSING  Sigma = 0  }  δint (e)  {  **case** phase:  PROCESSING:  For each waiting load sort the pallets into Waiting Pallets and Destination Pallets queue based on whether this location is the final location of the Pallet  Clear Waiting Loads  Sort Waiting Pallets by ID  Determine if there are enough pallets to make a load and if there is an available aircraft. There must be enough pallets to fill at least 50% of the aircraft.  If there are enough Pallets and a Destination aircraft, generate Load and add it to the Processed Loads queue  Check to see if any aircraft are expired (i.e. been waiting 48 hours without a load request). If so return the aircraft to its home base.  If there are Processed Loads  Phase = SENDLOAD  Sigma = 0  Else If there are Destination Pallets  Phase = DELIVER  Sigma = *Delivery Time (i.e. 30 min)*  Else  Phase = WAITING  Sigma = ∞  SENDLOAD:  Clear Processed Loads queue  Phase = PROCESSING  Sigma = 0  DELIVER:  Clear Destination Pallets queue  Phase = PROCESSING  Sigma = 0  }  λ(s)  {  If Phase = DELIVER  Output all Pallets in the Destination Pallets queue  If Phase = SENDLOAD  Output all Loads in the Processed Loads queue  } | |

### Aircraft = <X, Y, S, ta, δext, δint, λ>

|  |  |
| --- | --- |
|  | |
| **State Variables**  Sigma = 0  Phase = INIT  Aircraft ID = *Set on initialization*  Type = *Set on initialization*  Average Speed = *Set on initialization*  Home Location = *Set on initialization*  Distance Remaining = 0  Load = {} (<id, destination, distance, aircraft id, pallets>) | **Formal Specifications**  X = { Load ∈ <String, N, N, N, vector<Pallet>>}  Y = { Aircraft Status ∈ <N, N, N, N, N, N> | Load ∈ <String, N, N, N, vector<Pallet>>}  S = {{Sigma, Phase, Aircraft ID, Type, Average Speed, Home Location, Distance Remaining, Load}} |
| δext (mbs, e, x <Load>)  {  If X is Load and the load is for this aircraft  Add X to aircraft Load  Set Distance Remaining to distance of Load  If Phase = WAITING and Distance Remaining > 0  Phase = FLYING  Sigma = 0  Else If PHASE = FLYING  Sigma = Update time period with e  Else If PHASE = WAITING  Sigma = ∞  }  δint (e)  {  **Case**  INIT:  Phase = WAITING  Sigma = ∞  FLYING:  If Distance Remaining < 0  Phase = UNLOADING  Sigma = *Unload Time* (i.e. 30 min)  Else  Update current position  Phase = FLYING  Sigma = *Update Time* (i.e. 10 min)  UNLOADING:  Phase = WAITING  Sigma = ∞  }  λ(s)  {  If Phase = UNLOADING  Unload the current Load to the destination location  Send Aircraft Status to destination location  If Phase = INIT  Send Aircraft Status to home location  } | |

## Coupled Models

