IE 305 Project
PART II - Deliverable

Group 42:
Eylül Yaltır
Sude Naz Arıcı
Tunahan Arslan
Murat Emre Aktan

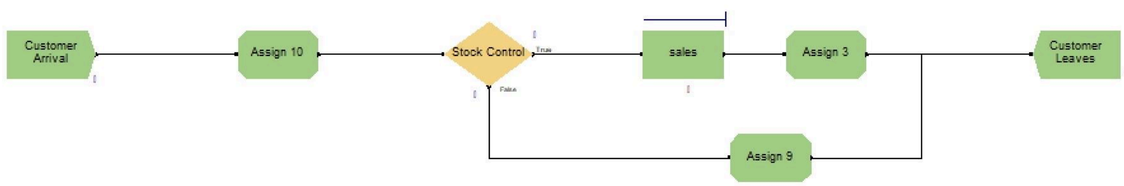
1) Overall structure of the Models

The Arena model is composed of three main parts:

1. **Customer demand & sales flow (order arrival → demand assignment → stock decision → sales/leave)**
2. **Production system (Line 1, Line 2, Line 3 → batching → Assembly station → finished harness output)**
3. **Repair Lines**

First two parts are linked through inventory/stock logic: customers can complete a purchase only if sufficient finished products are available; otherwise the customer leaves (dissatisfied customer).

2) Customer arrival + demand quantity (Empirical discrete)



- **Customer arrivals** are generated via a Create module ("Customer Arrival").
  As specified later in the project instructions, the time between arrivals is modeled as: TBA~EXPO(9.23 minutes)TBA~EXPO(9.23 minutes)
- Demand per arriving customer is assigned using an **empirical discrete distribution** in **Assign 10** (demand attribute):

Demand = DISC(
  0.0255, 30,
  0.0526, 40,
  0.1562, 50,
  0.3063, 60,
  0.4010, 70,
  0.7722, 80,
  1.0,   90
)

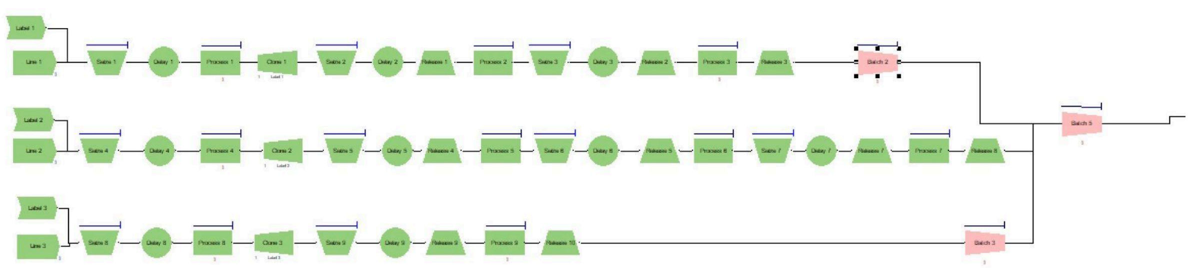
This maps cumulative probabilities (CDF) to observed demand values and preserves the historical frequency of demand.

3) Stock control + sales department

After demand is assigned, the model checks whether the order can be fulfilled:

- **Decide module (“Stock Control”)**: checks if available inventory is sufficient for the entity’s demand.
    - If **True** → entity proceeds to the **Sales** process.
    - If **False** → entity is routed to a “leave” branch (Dissatisfied Customer).
- **Sales process** is modeled as a **Process module** that seizes the resource **SalesDep**.
  If the project requires working hours, SalesDep is defined with a **capacity schedule** (open/close times).
  Recommended **Schedule Rule = Ignore** so that an ongoing sale finishes even if closing time occurs.
- After sales completion, an Assign module updates inventory (e.g., decrements finished goods by Demand) and the customer leaves.

4) Production lines (Line 1, Line 2, Line 3)



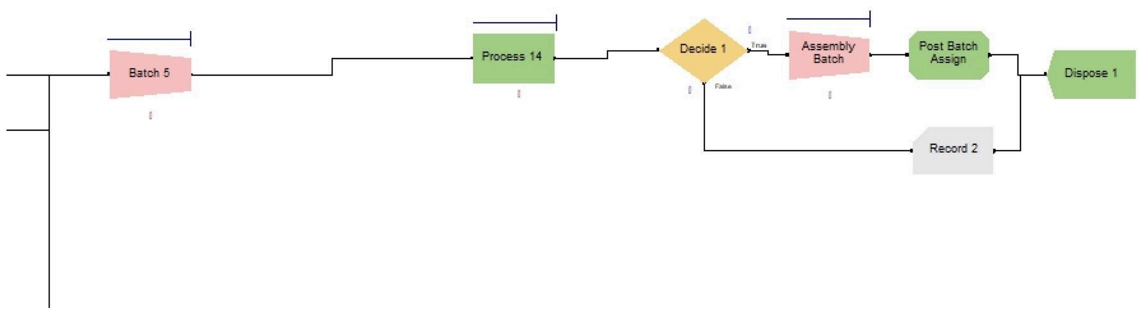
Each production line is modeled as a sequence of:

- **Seize** → **Delay** → **Release** logic blocks (machine as a Resource)
- Intermediate processes represent consecutive stations on the line.
- Line 1, Line 2 and Line 3 contain multiple stations (seize-delay-release blocks) and Clone modules.

The production floor is modeled using distinct segments for each line, converging at the assembly station.

- **Line Configuration:**
    - **Line 1:** 3 Stations (Produces Subassembly 1).
    - **Line 2:** 4 Stations (Produces Subassembly 2).
    - **Line 3:** 2 Stations (Produces Subassembly 3).
- **Kanban Logic:** To simulate JIT principles, each station is modeled with a "Seize-Delay-Release" logic where the resource represents the machine. A "Hold" or "Seize" mechanism ensures that a station cannot process a new part unless a Kanban signal is available from the downstream buffer.
- Each line's output is routed to batching logic so that the correct combination of components can feed Assembly.

5) Batching + Assembly station



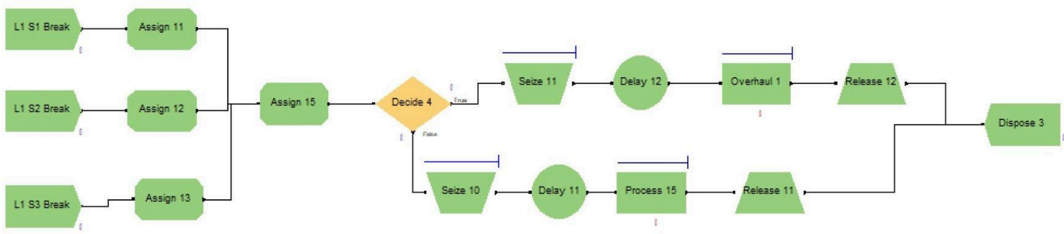
After lines produce components:

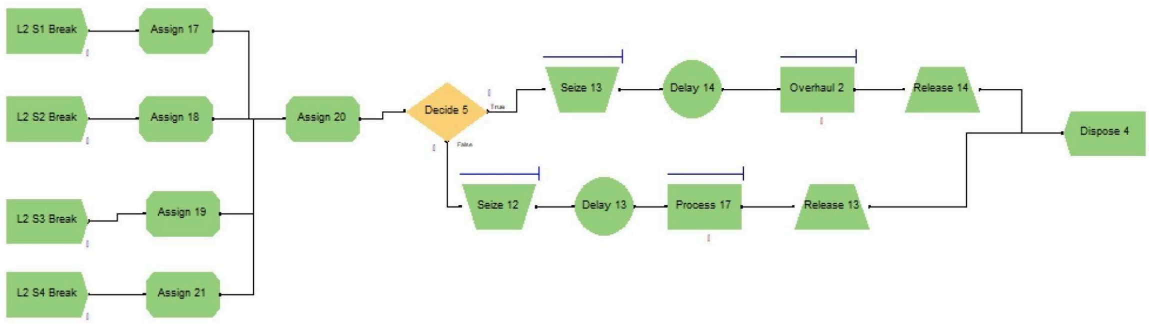
- **Batch modules** combine required components into kits for assembly.
- The merged kit enters the **Assembly station** (Process 14) modeled as:
    - Seize (Assembler resource)
    - Delay (assembly time)
    - Release

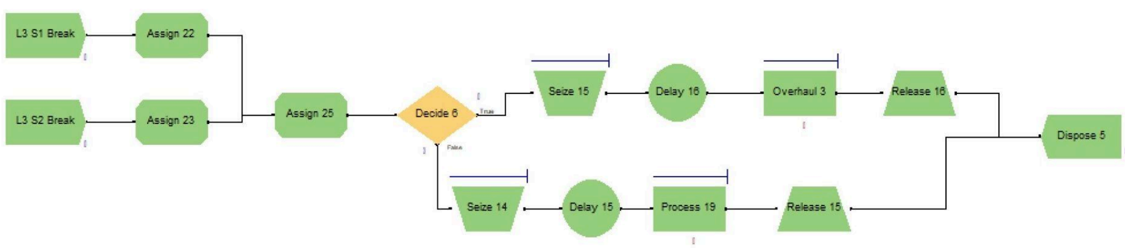
After assembly:

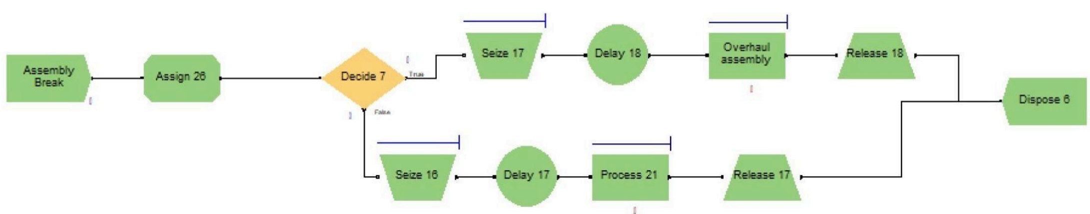
- A Decide module (Decide 1) routes entities:
    - **True**: completed harness goes through final batch/assignment and is disposed as "finished output".
    - **False**: routed to a record block for tracking scraps.

6. Repair Lines









- Independent Clocks: Each machine has its own "Create" module generating failures at intervals of *EXPO(8)* hours. This ensures that if one machine fails, it does not reset the "clock" for other machines.
- Entity Definition: A specific entity type is used for all failures. This separates maintenance activities from production statistics, preventing the inflation of throughput numbers.
- Identification (Indexing): To efficiently handle multiple machines within a single loop, we used Resource Sets and Attributes.
    - Each failure entity is assigned an attribute MyIndex (1, 2, or 3) upon creation.

    - This index allows the logic to seize the specific machine from a Set (Set_L1_Machines) corresponding to the breakdown.

The project requires that every 10th failure triggers a major overhaul. We implemented this using Variable Arrays:

- Tracking: Arrays such as *L1_FailCount(MyIndex)* track the cumulative number of failures for each specific machine.
- Decision Logic: A "Decide" module checks the condition:
  MOD(FailCount(MyIndex), 10) == 0
  If True, the entity proceeds to the Overhaul Path; if False, it proceeds to the Regular Repair Path.

Throughput Estimation

We checked the number of outputs of the ending process at each line to find the throughputs of each line.

A) Standalone throughputs of Line 1 / Line 2 / Line 3 (no breakdowns)

Reported outputs :

- Line 1 standalone throughput (no repairs): 4192 **units/day**
- Line 2 standalone throughput (no repairs): 2013 **units/day**
- Line 3 standalone throughput (no repairs): 8694 **units/day**

B) Assembly throughput without breakdowns/repairs

We run the full integrated model **with breakdowns disabled** and measure output at the final Dispose.

The completed output count is:

- **Assembly throughput (no breaks)** = **1097 output entities / day**

C) Assembly throughput with breakdowns/repairs

We could not run the model with the repairs properly. But our full design is included in the file.

Daily demand rate (Harnesses/Day) — Verification calculation

Daily demand rate is used to verify whether the simulated demand scale is reasonable.

1. Expected demand per arriving customer:

$E[D]=\sum x\cdot p(x)E[D]=\sum x\cdot p(x)$

Using the empirical distribution:

$E[D]\approx 72.859E[D]\approx 72.859$

2. Expected number of arrivals per day depends on **operating minutes** MM:

$E[Narrivals/day]=M9.23E[Narrivals/day]=9.23M$

- If your system is open 24 hours: $M=1440M=1440$ → $\approx 155.99\approx 155.99$ arrivals/day
- If open HH hours: $M=60HM=60H$

3. Therefore expected daily demand in harness units:

$E[Demand/day]=M9.23\times 72.859E[Demand/day]=9.23M\times 72.859$