# Serie 7

**Content:**
– Framework SimJ

## (20) SimJ

### *A Description of the model*

A clothes factory is composed of the production site and the reparation site for the machines.

The production site has $n$ working stations each having a sewing machine. The factory has a total of $n+m$ sewing machine ; this implies that $m$ machines are in stock.

From times to times a machine will fail. The time to fail follows an exponential law with mean of $m_breakdown$.

In case of breakdown, the broken machine is sent to the reparation site and replaced by one of the machines in the stock. If the stock is empty, the borken machine gets replaced as soon as new machines arrive in the stock. The necessary time to exchange a broken machine can be neglected.

The reparation site has employed $k$ machinists. There exist two types of breakdown :
– Hevy breakdowns : they occur with probability $p_1$ and the reparation time follows a uniform law with parameters $min_1$ and $max_1$.
– Light breakdowns : they occur with probability $p_2$ and the reparation time follows a uniform law with parameters $min_2$ and $max_2$.

After successful reparation the machine is sent back to the stock of the production site.

### *B Simulation Game*

Complete the table describing the evolution of the simulation (Table 2). Take the follwing values for the parameters defined above :
– The capacity of the production site $n$ is 2.
– The number of machines in stock $m$ is 2.
– The capacity of the repair site $k$ is 1.
– The mean time to failures are given in the Table 1

| Numéro | Creation time | Working duration without breakdown | Duration of reparation |
|---|---|---|---|
| 1 | 0 | 10 | 100 |
| 2 | 0 | 20 | 50 |
| 3 | 0 | 200 | – |
| 4 | 0 | 200 | – |

Table 1 – Timings and durations of the factory simulation

### *C Simulation Application*

Create an application simluating the above model. The parameters

$$n, m, m\_breakdown, k, p_1, min_1, max_1, min_2 \text{ et } max_2$$

have to be defined by the user upon starting the simluation.

The production site and the reparation site have to be modeled as *resources* whereas, the sewing machines are modeled as *entities (temporary)* all generated at time 0. The stock can be considered as a queue of the production site.

Use the `SimEventScanResources` class to follow the evolution of the simulation.

## Help

1. The source code of the `SimJ` framework is available online [1].
2. The source code for the simluation of a supermarket is available online [1].
3. Create un new project, starting with the source-code available for the `supermarket`.
4. Modify the file `build.properties` to your needs. The value of `project.name` should be changed to *simj_factory*.
5. Create a new project `factory`.
6. Create a GUI (for example `FactoryFrame`) in the `factory` packet. This will be the main class of the application. Have a look at the class `SuperMarcheFrame`.
7. Adapt the value of the property `main.class` (for example `factory.FactoryFrame`) in the `build.properties` file. Also check the other properties and modify them if necessary.

| $t$ | Evénements | | | Etat de l'usine | | | |
|---|---|---|---|---|---|---|---|
| | $e^{e,r}_{t,n}$ | T | **Actions** | **Production** | | **Réparation** | |
| 0 | $e^{1,\cdot}_{1,1}$ | 0 | create $m_1$ and send to production site<br>create $e^{1,1}_{2,5}$ and schedule for $t+0$ | $m_1$ | | | |
| 0 | $e^{1,\cdot}_{1,2}$ | 0 | create $m_2$ and send to production site<br>create $e^{2,1}_{2,6}$ and schedule for $t+0$ | $m_2$,$m_1$ | | | |
| 0 | $e^{1,\cdot}_{1,3}$ | 0 | create $m_3$ and send to production site | $m_3$,$m_2$<br>$m_1$ | | | |
| 0 | $e^{1,\cdot}_{1,4}$ | 0 | create $m_4$ and send to production site | $m_4$,$m_3$<br>$m_2$,$m_1$ | | | |
| 0 | $e^{1,1}_{2,5}$ | 0 | fait entrer $m_1$ dans l'atelier de production<br>create $e^{1,1}_{3,7}$ and schedule for $t+10$ | $m_4$,$m_3$<br>$m_2$ | $m_1$ | | |
| 0 | $e^{2,1}_{2,6}$ | 0 | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | $m_2$,$m_1$ | $m_4$,$m_3$ | | |
| 199 | $e^{\cdot,\cdot}_{0,0}$ | 0 | end of simulation | | | | |
| | | | never executed | | | | |
| | | | never executed | | | | |

Table 2 – Table of the factory simluation to complete

### Références

[1] Jacques Pasquier. Génie logiciel I, 2013. http://moodle2.unifr.ch/course/view.php?id=1252 (accessed May 13, 2013).