Simplicity is the ultimate sophistication

How to Run Simulations and Interpret the Results

Prof. Dr. Carlo Simon · Lara Zakfeld
✉ {simon, zakfeld}@hs-worms.de

Hochschule Worms

GAPS

IARIA

Agenda

The tutorial demonstrates how to model real world simulation scenarios for production and logistics with the aid of State machines and Petri nets.

1 Introduction of the real-world example

2 Introduction of the simulation environment

3 Simulation Workflow: *feed* → *simulate* → *visualize*

4 Needed master file data

5 From State machines to Petri nets

Participants gain access to the **Process-Simulation.Center (P-S.C)** including a three month trial.

A live demonstration explains fundamental, advanced concepts and best practices.

Goal of the Simulation

A high-bay warehouse for more than 22.000 pallets of chemical and pharmaceutical goods must be

- simulated concerning the goods movement
- without confronting the employees with too many details.

Employees may simulate process variants

- by changing processing times and resources and
- for different sets of „simplified“ orders.

A dashboard may

- visualize the processes as they occur and
- give different overviews.

Process-Simulation.Center (P-S.C) & Dashboard

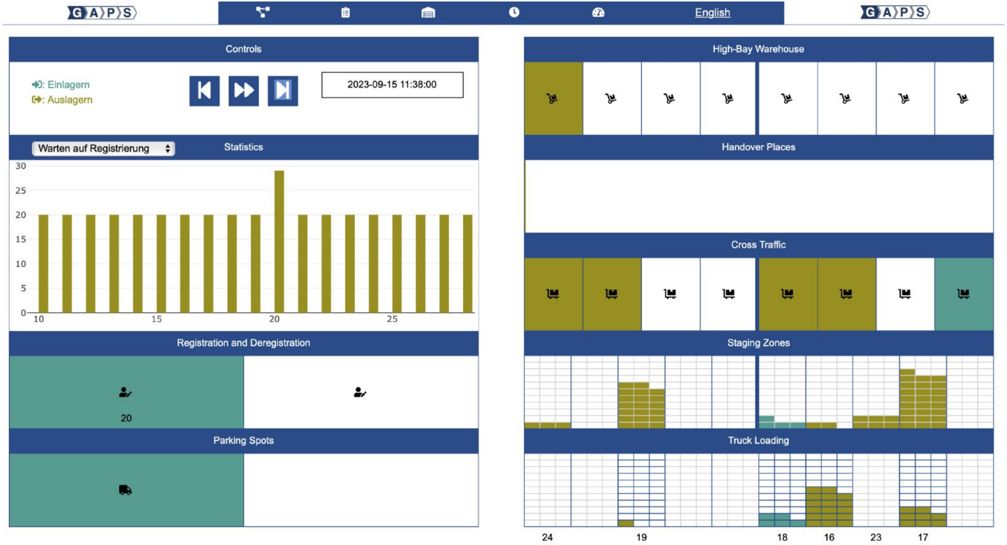
Process-Simulation.Center (P-S.C)

- Petri net based Integrated Management System (IMS)
- >400 registered, almost academic users
- Uses high-level Petri-Nets as a universal modeling and simulation language for dynamic systems
- Simulation of processes controlled by limiting resources

Dashboard

- Dashboard app especially developed for the business partner
- Administration of master file data
- Visualizes the process flow in the course of a day

A first impression



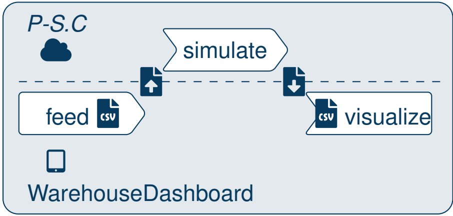
: Einlagern
: Auslagern

2023-09-15 11:38:00

Warten auf Registrierung

Simulation workflow

- **WarehouseDashboard** and **P-S.C** interact.
- Phase *feed*: Master file data management (**WarehouseDashboard**: tabs 📋 , 🏭 & 🕓). Data is supplied in CSV format.
- Phase *simulate*: **P-S.C** simulation of the warehouse movements on the base of this data



- Phase *visualize*: The exported simulation results are loaded up to the **WarehouseDashboard** for visualization (**WarehouseDashboard**: tab ⏲).

We provide a live demonstration!

The following frames deepen your knowledge and can be used for further nodes.

Order data 📋

ID	📋	📦	🪧	🚚	💼	🛒
1	⚗	1-33	📥	📅	📅	1-10
2	💊	1-33	📤	📅	📅	1-10
⋮						

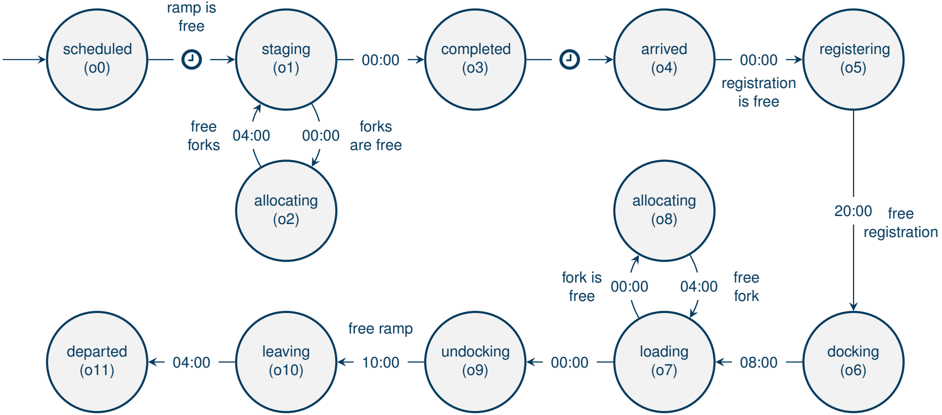
- Order number (auto increment), product and amount
- Transfer into stock or release from stock
- Arrival time and time of placing the goods at the disposal
- Ramp

Limiting ressources 🏭

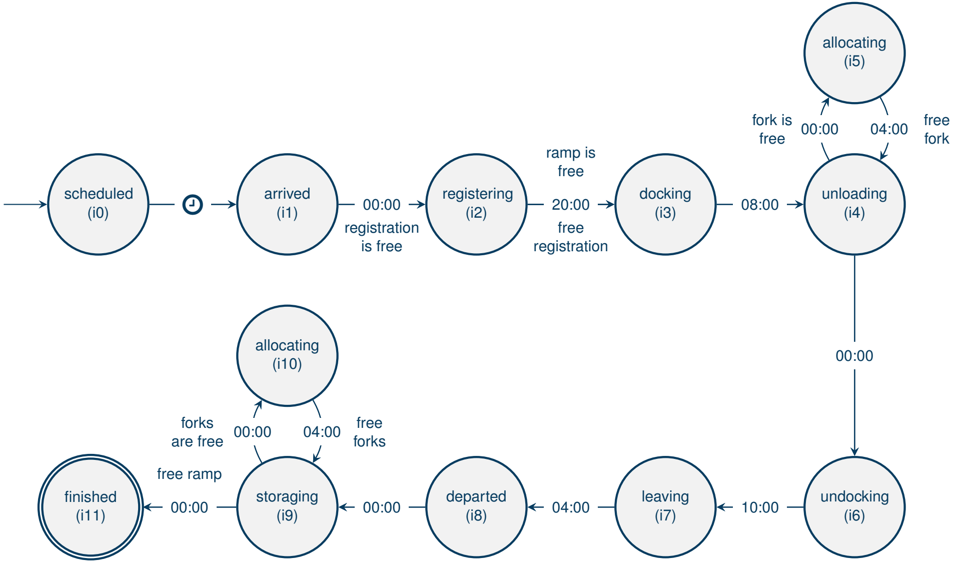
🏭	⚗	💊
🛒	...	...
🛒	...	...
🚚	...	...
👤⚙		

- Number narrow aisle forklift (🛒), forklifts (🛒) and ramps (🚚) per storage area chemestry (⚗) and pharma (💊)
- Number employees at registration (👤⚙)

Finite state machine for releasing from stock



Finite state machine for transferring into stock



Controls 🕒

ID	!	🪧	🕒	⚙
3	1-10	↪	00:04:00	remove a pallet from a truck
7	1-10	↪	00:20:00	Registration time
19	1-10	⇥	00:04:00	Move a pallet to a truck

⋮

- State changes are the core of the simulation model
- Time delays are specified in a minute raster
- Priorisation of activities (!) like into stock before out of stock.

Gain access to the tool!

The name is the address

`https://www.process-simulation.center`

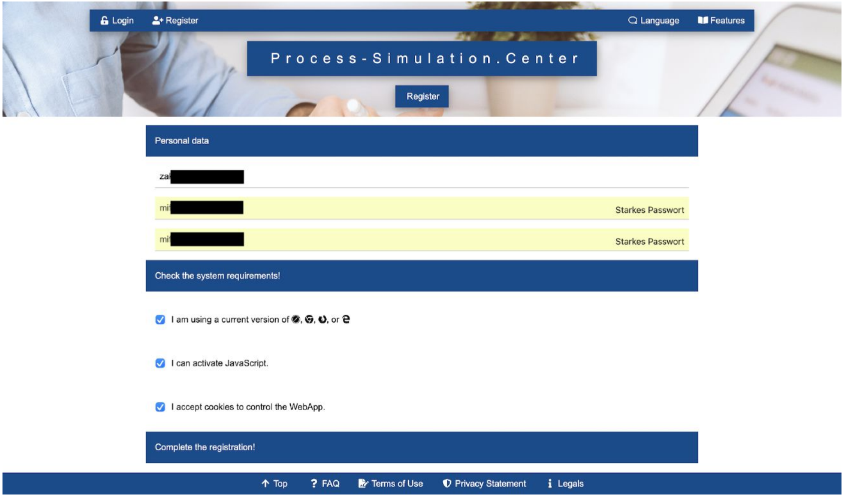
You can register for a three month trial account.

Large tutorial with examples and applications.

Access the dashboard

`https://www.process-simulation.center/WarehouseDashboard`

Registration



Personal data

[redacted]

[redacted] Starkes Passwort

[redacted] Starkes Passwort

Check the system requirements!

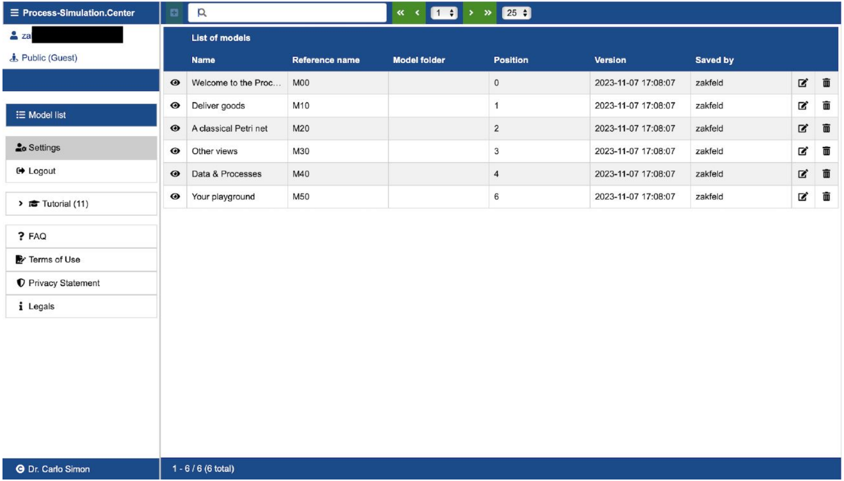
☑ I am using a current version of , , , or

☑ I can activate JavaScript.

☑ I accept cookies to control the WebApp.

Complete the registration!

Some predefined models



List of models

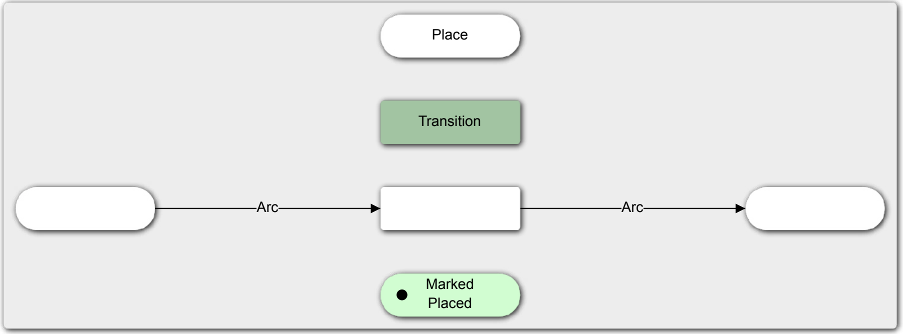
Name	Reference name	Model folder	Position	Version	Saved by
Welcome to the Proc…	M00		0	2023-11-07 17:08:07	zakfeld
Deliver goods	M10		1	2023-11-07 17:08:07	zakfeld
A classical Petri net	M20		2	2023-11-07 17:08:07	zakfeld
Other views	M30		3	2023-11-07 17:08:07	zakfeld
Data & Processes	M40		4	2023-11-07 17:08:07	zakfeld
Your playground	M50		6	2023-11-07 17:08:07	zakfeld

1 - 6 / 6 (6 total)

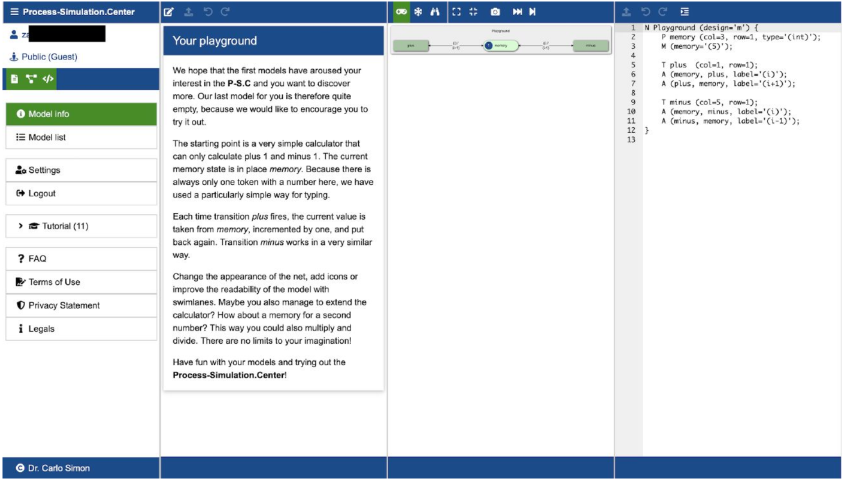
New look and feel for Petri nets

Classical design was good for modeling on a chalkboard.

Especially for processes, the **P-S.C** implements a timely design.



A small playground



Your playground

We hope that the first models have aroused your interest in the **P-S.C** and you want to discover more. Our last model for you is therefore quite empty, because we would like to encourage you to try it out.

The starting point is a very simple calculator that can only calculate plus 1 and minus 1. The current memory state is in place *memory*. Because there is always only one token with a number here, we have used a particularly simple way for typing.

Each time transition *plus* fires, the current value is taken from *memory*, incremented by one, and put back again. Transition *minus* works in a very similar way.

Change the appearance of the net, add icons or improve the readability of the model with swimlanes. Maybe you also manage to extend the calculator? How about a memory for a second number? This way you could also multiply and divide. There are no limits to your imagination!

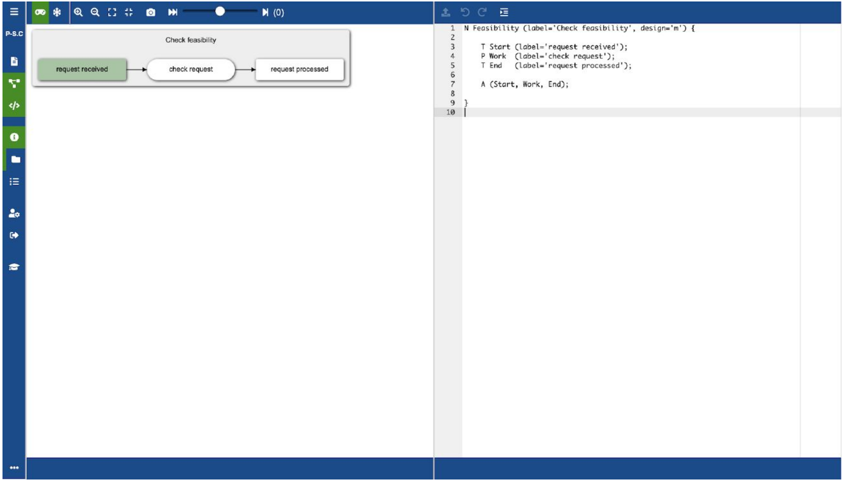
Have fun with your models and trying out the **Process-Simulation.Center**!

```
N Playground (design='m') {
    P memory (col=3, row=1, type='(int)');
    M (memory='(5)');

    T plus  (col=1, row=1);
    A (memory, plus, label='(i)');
    A (plus, memory, label='(i+1)');

    T minus (col=5, row=1);
    A (memory, minus, label='(i)');
    A (minus, memory, label='(i-1)');
}
```

A first small example



```
N Feasibility (label='Check feasibility', design='m') {

    T Start (label='request received');
    P Work  (label='check request');
    T End   (label='request processed');

    A (Start, Work, End);

}

```