



# JUPYTER NOTEBOOK AND ECLIPSE: DO MODELING WITH A SCRIPT-BASED PROTOTYPING APPROACH

Sébastien Revol

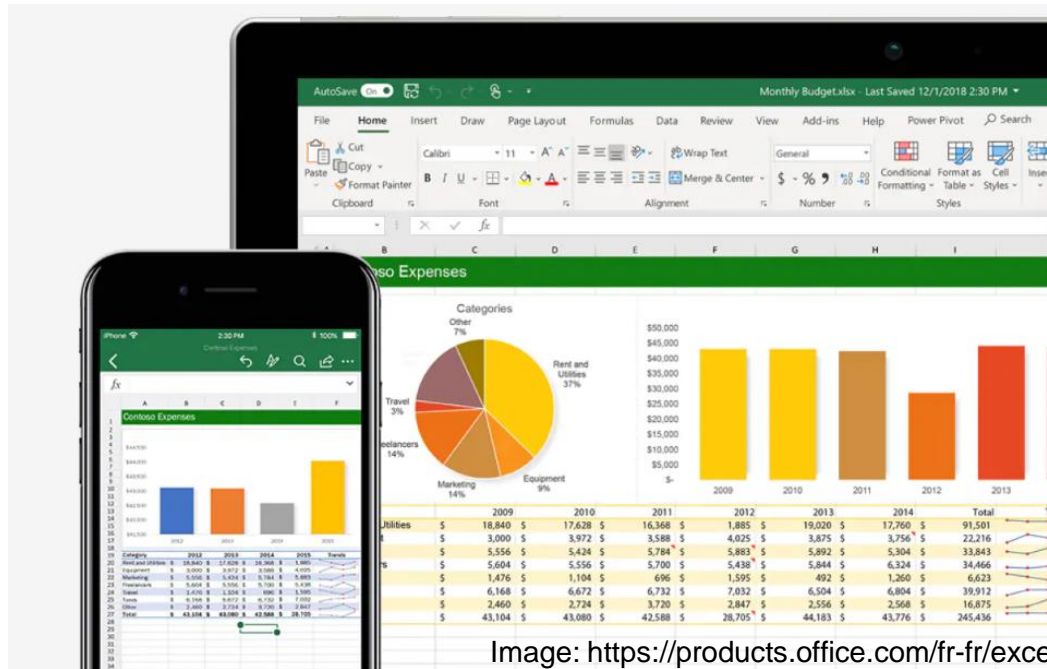
## A SYSTEM ENGINEERING TOOL?



- **SysML is a System Modeling Language**
  - Papyrus is an Eclipse/EMF-based SysML editor
- **CEA provides tools on top of EMF SysML models**
  - Papyrus is a central platform to edit models
- **Different purposes :**
  - Model analysis
  - Design automation (code generation, model transformation...)
  - Model simulation
    - Result analysis
  - Optimization
  - Requirement traceability
  - ...
- **High need to customize our tools for our different end-users**
- **End-users are generally not eclipse developers**
  - Often not eclipse users...
  - Need to build own custom flows

# SYSTEM ENGINEERING TOOL IN REAL LIFE

- Many end-users are using...


















## Microsoft Excel

## BUT WHY?!

- **Available**
  - Installed on most desktop
  - Available on cloud
- **Single tool for :**
  - **Data edition**
    - Simple forms
    - (quite) scalable tables
  - **Analyze, compute ...**
    - Simple language
    - Rich libraries
  - **Visualize**
    - Graphs (scatter, bars, pie charts...)
    - Conditional formatting
    - Filtering, sorting...
- **XLSX (and CSV): de facto format pivot**
  - Many tools provide import/export

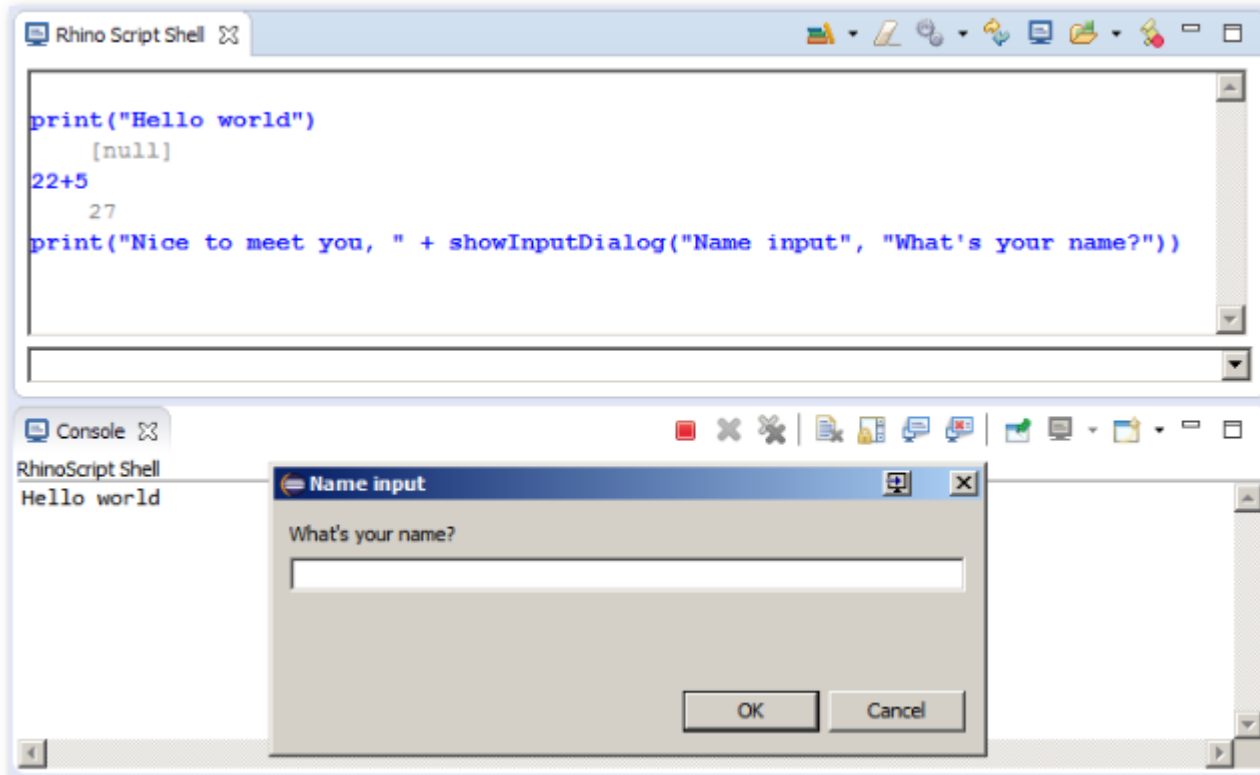
➔ **get results very rapidly**

- Scripting languages
  - Interpreted, no build/deliver issues
- Python
  - Taught at school
  - Very rich community
  - Many libraries
    - Data Science
    - IA
    - ...

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1

<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>

- <https://www.eclipse.org/ease/>

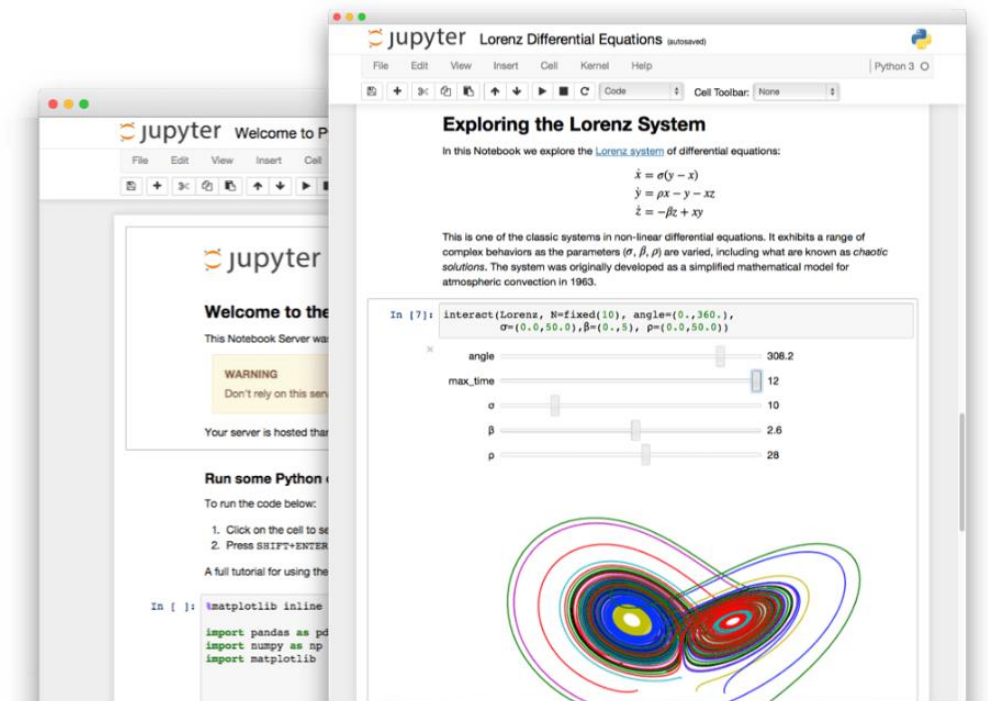


- **multi-languages**
  - Python
  - Javascript
  - Ruby
  - ...
- **Direct access to complete Eclipse Java API from scripting languages**
  - Almost anything coded in Java can be coded in scripting languages
  - Dynamic IDE modifications
    - Add new view, menus, buttons...
- **Can call functionalities implemented in scripting language**
  - Scripting languages inside models for simulation...

**➔ Adding scripting in Eclipse allows engineers with low SW/ skills to adapt the tool to their specific needs and workflows**

- <https://jupyter.org>
- Mix of Markdown and viewer, scripting interpreter
- Simple widget library
- Very rich graph/visualization libs
  - Graphs (plot.ly, matplotlib)
  - Interactive tables

➔ Jupyter provides simple API to Create dedicated Uis, and propose advanced visualization tools.



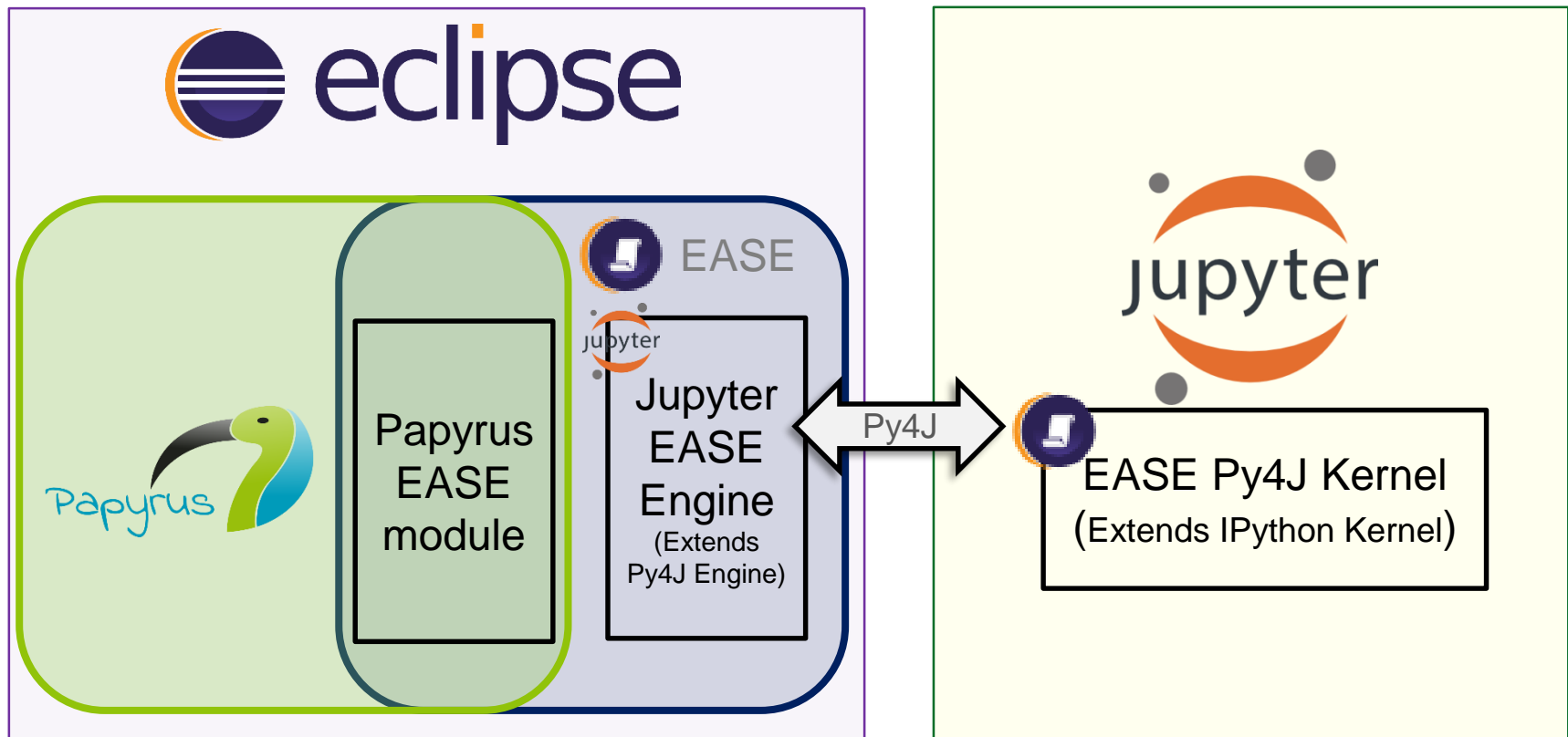


## EMF + SCRIPTING + JUPYTER VS EXCEL?

- **EMF provides a much efficient way to structure information**
  - But still requires important sw developers skills
- **Python allows non experts to build their own workflows**
  - Computation
  - Analysis
  - Optimization ...
- **Jupyter provides simple API to create :**
  - Simple Forms
  - Advanced visualization
- **Moreover, for advanced users :**
  - Many external tools/libs propose a python API
    - Python EASE engine allows to easy use them from JAVA!
      - Simpler than building a dedicated JNI interface
  - Jupyter web approach enables to easily integrate JS objects, libs etc...

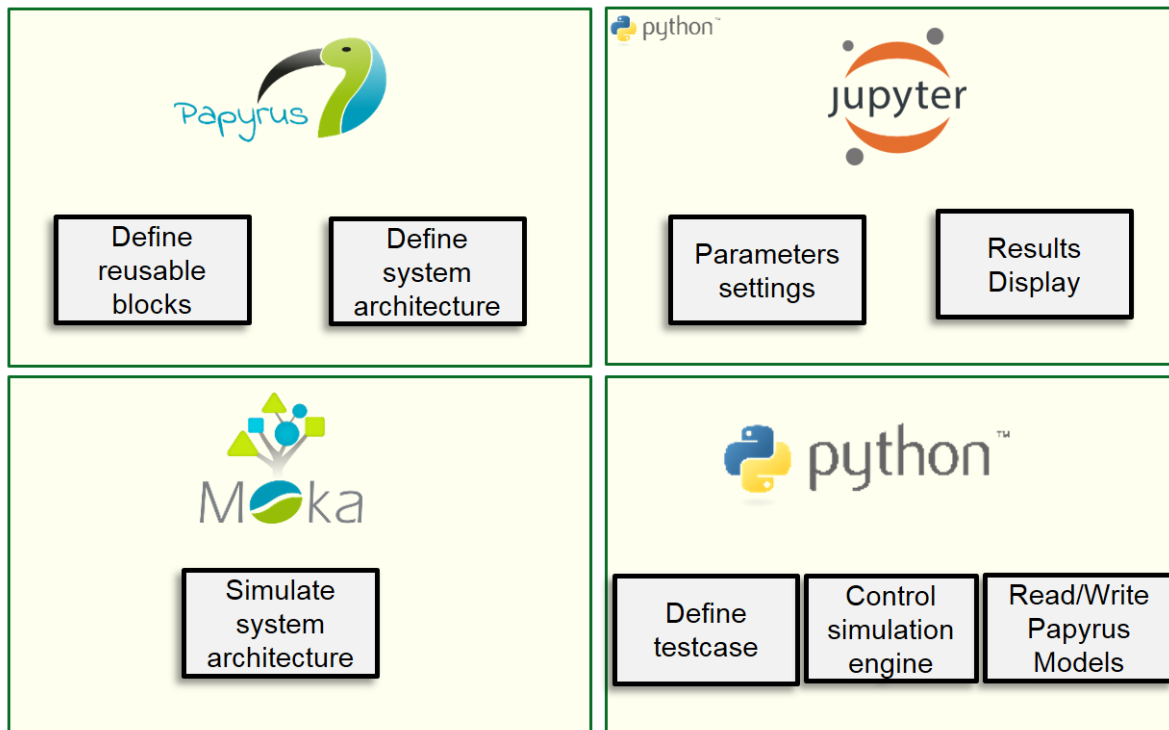
## PAPYRUS + EASE + JUPYTER

- Developped a Jupyter specific engine on top of EASE Py4J engine



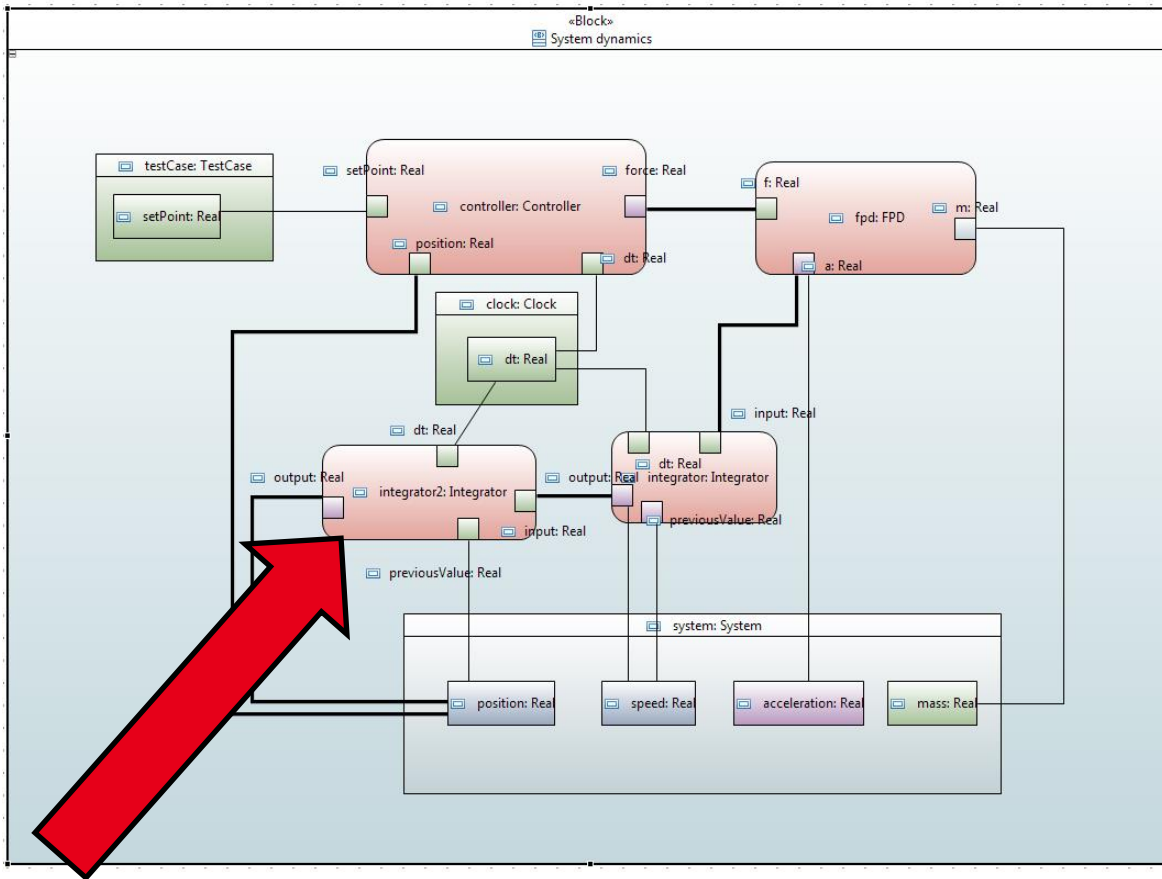
## EXAMPLE : PARAMETRIC ENGINE

- Interactive dashboard for system analysis
- Python used as an action language in models
  - Moka\* interpreter executes Python code



\*Moka : <https://wiki.eclipse.org/Papyrus/UserGuide/ModelExecution>

# PARAMETRIC DIAGRAM INTERPRETER



```

1 def run(block) :
2   block.output = block.input * block.dt + block.previousValue

```

# PARAMETRIC DIAGRAM INTERPRETER

The image displays a parametric diagram interpreter interface. On the left, a block diagram titled «Block» System dynamics is shown. It includes components like testCase: TestCase, controller: Controller, clock: Clock, integrator2: Integrator, and system: System. On the right, a control panel allows for parameter tuning and simulation execution. The parameters are:

- dt: 0.01
- endTime: 15.00
- controller\_G: 15.80
- controller\_Ki: 0.00
- controller\_Kd: 6.00

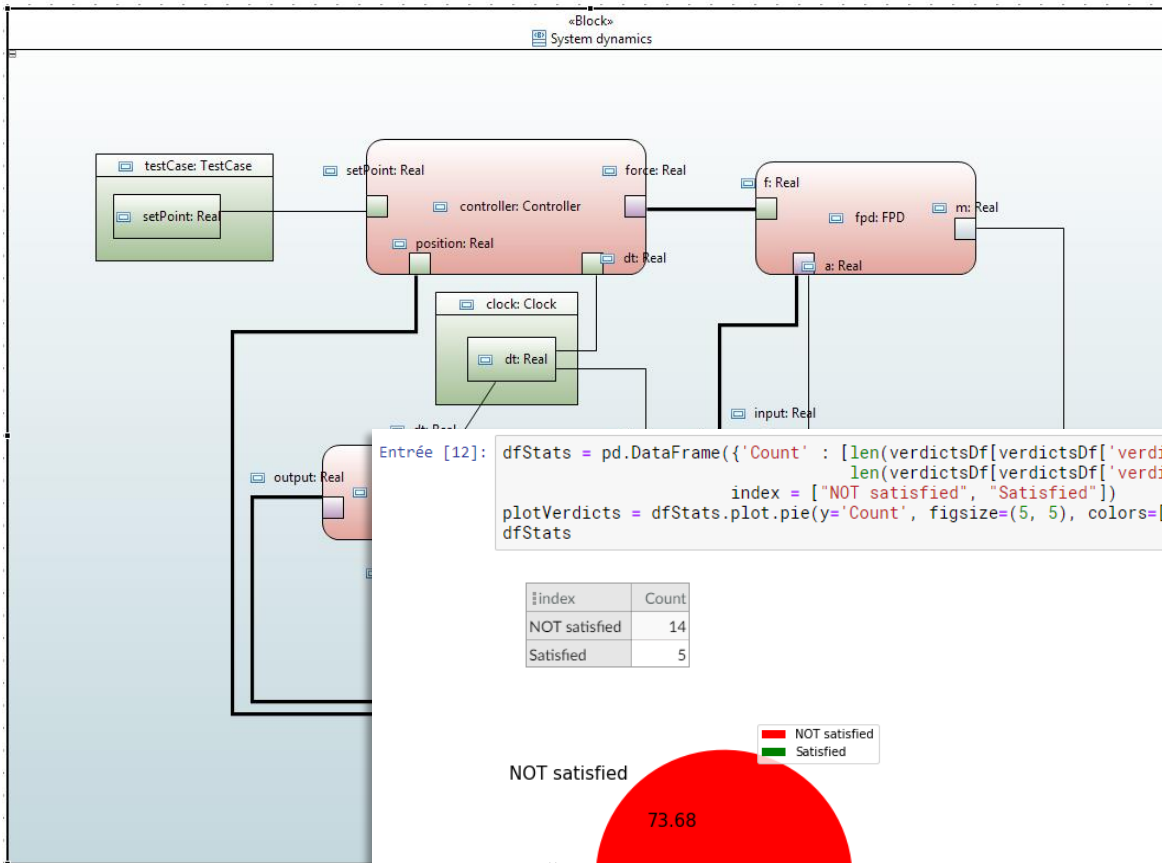
The simulation results are displayed in three stacked plots:

- acceleration:** Shows a sharp initial spike reaching approximately 1500, followed by a rapid decay to zero.
- speed:** Shows a peak of about 25, followed by a smooth decay to zero.
- position:** Shows a smooth, asymptotic rise from 0 to a steady-state value of 10.

Parameter tuning in jupyter

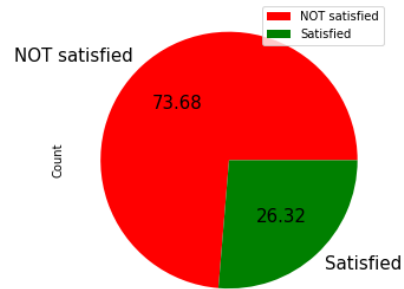
Java simulation control and result display

# PARAMETRIC DIAGRAM INTERPRETER



```
Entrée [12]: dfStats = pd.DataFrame({'Count' : [len(verdictsDf[verdictsDf['verdi
                                len(verdictsDf[verdictsDf['verdi
                                index = ["NOT satisfied", "Satisfied"])
plotVerdicts = dfStats.plot.pie(y='Count', figsize=(5, 5), colors=|
dfStats
```

index	Count
NOT satisfied	14
Satisfied	5



index	T	V	reduction ok	im ok	im
0	-40	6.500	True	False	14.286
1	-40	9.000	True	False	19.780
2	-40	13.500	True	False	29.670
3	-40	16.000	True	False	35.165
4	-20	6.500	True	False	13.099
5	-20	9.000	True	False	18.137
6	-20	13.500	True	False	27.205
7	-20	16.000	True	False	32.243
8	25	6.500	True	False	11.036
9	25	9.000	True	False	15.280
10	25	13.500	True	False	22.920
11	25	16.000	True	False	27.165
12	80	6.500	True	True	9.254
13	80	9.000	True	False	12.814
14	80	13.500	True	False	19.220
15	80	16.000	True	False	22.780
16	120	6.500	True	True	8.282
17	120	9.000	True	False	11.467
18	120	13.500	True	False	17.201
19	120	16.000	True	False	20.386
20	140	6.500	True	True	7.869
21	140	9.000	True	False	10.895
22	140	13.500	True	False	16.342
23	140	16.000	True	False	19.369

# EXAMPLE OF JUPYTER AS ECLIPSE HEADLESS APPLICATION



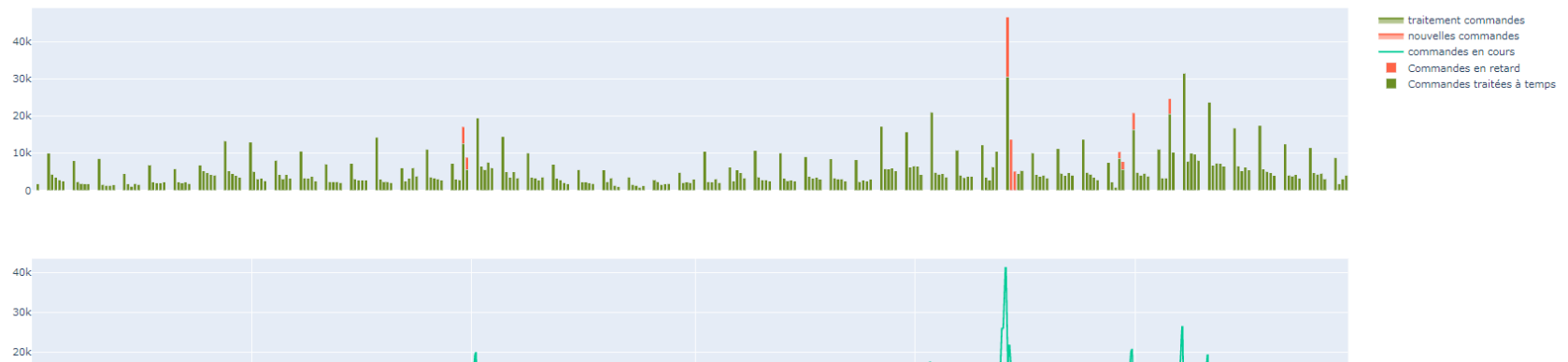
cut-off commandes/livraisons

Facteur volumisation

Equipes:  3/8

**Python** : Excel import and simulation trace post processing  
**Eclipse/Java**: Model Execution

Index	Echéance	Total Commandes	Commandes en retard	Commandes à l'heure	Pourcentage commandes en retard
0	Thu 28/06/2018 15:00:00	17074	4413	12661	26
1	Fri 29/06/2018 15:00:00	8842	3200	5642	36
2	Mon 26/11/2018 15:00:00	46616	16009	30608	34
3	Tue 27/11/2018 15:00:00	13749	13749	0	100
4	Wed 28/11/2018 15:00:00	5108	5108	0	100
5	Thu 27/12/2018 15:00:00	10325	1744	8581	17
6	Fri 28/12/2018 15:00:00	7706	2033	5674	26
7	Mon 31/12/2018 15:00:00	20794	4491	16303	22
8	Thu 10/01/2019 15:00:00	24701	4082	20619	17



- **Jupyter engine released under Papyrus umbrella**
  - <https://download.eclipse.org/modeling/mdt/papyrus/components/ease/2019-03/>
  - Also includes Papyrus modules
- **Jupyter engine will move to EASE project**
  - No dependency on Papyrus
  - Should be available in early 2020.
    - Mainly name refactoring, code convention alignments...

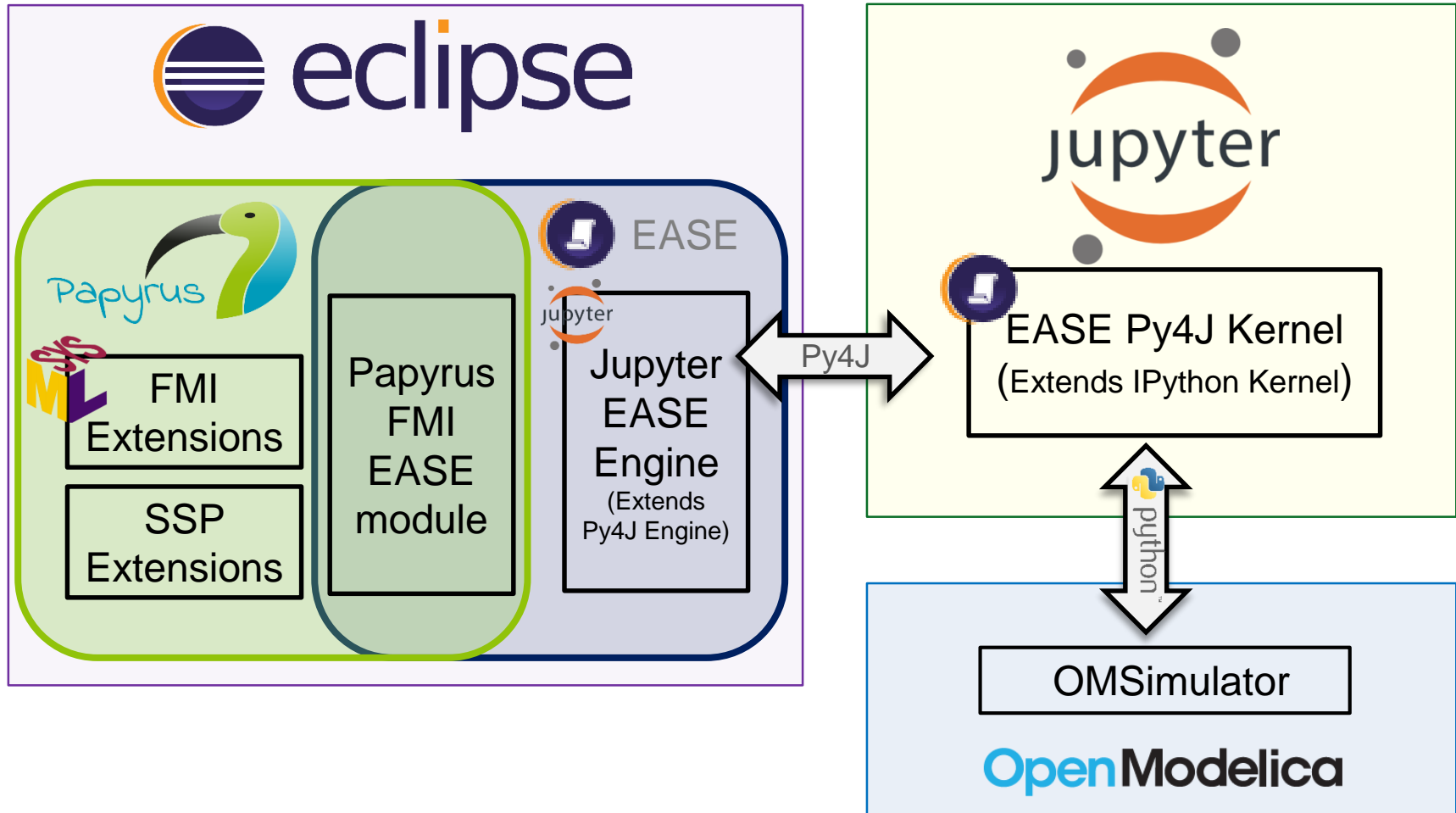


---

Commissariat à l'énergie atomique et aux énergies alternatives  
Institut List | CEA SACLAY NANO-INNOV | BAT. 861 – PC142  
91191 Gif-sur-Yvette Cedex - FRANCE  
[www-list.cea.fr](http://www-list.cea.fr)

Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019

# OPENCPS\* : INTEGRATION OF EXTERNAL FMI SIMULATOR FOR AN INTERACTIVE DASHBOARD



\*OpenCPS: ITEA3 European project, <https://www.opencps.eu>

- **Mix edition/computation/vizualization**
  - Monolithique files
- **Low reuse**
  - No modularity
- **Computation coded in “assembly” ...**

```
=SI(H100*SIGNE(O99)<=0;H100/$M$83/$J$83;H100*$M$83/$J$83)
```

- Only the author has a chance to remember the meaning of each cell...
- **Closed tool :**
  - No easy interactions with other tools