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
Many end-users are using…
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
    • …

![Image of language ranking chart](https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages)
ECLIPSE ADVANCED SCRIPTING ENVIRONMENT (EASE)

- https://www.eclipse.org/ease/
ECLIPSE 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
JUPYTER NOTEBOOK

- [https://jupyter.org](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 developpers 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…
• Developed 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
def run(block):
    block.output = block.input * block.dt + block.previousValue
PARAMETRIC DIAGRAM INTERPRETER

Parameter tuning in jupyter

Java simulation control and result display
Python: Excel import and simulation trace post processing
Eclipse/Java: Model Execution
STATUS AND NEXT STEPS

- **Jupyter engine released under Papyrus umbrella**
  - 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…
OPENCPS* : INTEGRATION OF EXTERNAL FMI SIMULATOR FOR AN INTERACTIVE DASHBOARD

*OpenCPS: ITEA3 European project, https://www.opencps.eu
EXCEL LIMITATIONS?

- 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