

# Drop The Docs and Embrace the Model with Gaphor

An Introduction to Model Based Systems Engineering with Free and Open-Source tools

Frank Vanbever - 2024/02/03  
FOSDEM 2024 - Tool The Docs



# WHOAMI

---

- Frank Vanbever
- Father of 2
- Bass player successfully daylighting as a software developer for the last 10 years
- Electrical Engineer specialized in Embedded Systems
  - Know just enough about the rest of the stack to be dangerous
- I work for Mind
- Currently at a robotics company working as software architect



# Outline

---

- What is Model Based Systems Engineering (MBSE)?
- The Three Pillars of MBSE
  - Modeling Language
  - Modeling Method
  - Modeling Tool
- Gaphor as a Modeling Tool
- Gaphor as a Documentation Tool
- Tooling around your model

# What is a model?

*"An abstraction of a system, aimed at understanding, communicating, explaining or designing aspects of that system"<sup>1</sup>*

- A model is a central repository for design decisions
- Captured as
  - Model Elements
  - Relationships between elements
- A set of views describe a model

<sup>1</sup> - Dori, D. 2002. Object-Process Methodology: A Holistic System Paradigm. New York, NY, USA: Springer.

# View ≠ Model



La Trahison des Images- René Magritte

# What is MBSE?

- Formalized application of modeling to support
  - System Requirements
  - Design and Analysis
  - Verification and Validation
- During
  - Conceptual design phase
  - Throughout Development
  - All later lifecycle phases
- This is an alternative to the **document-based approach**
- Doesn't this describe your documentation efforts?

# Three Pillars of MBSE

## Modeling Language

- Multiple options
- Typically Graphical Languages
- SysML

## Modeling Method

- Plenty options out there
- Dependent on the processes of your organisation
- Beyond the scope of this presentation

## Modeling Tool

- Commercial closed source tools seem to be very popular
- Gaphor as a FOSS alternative

# “A Good Sketch is better than a long speech”

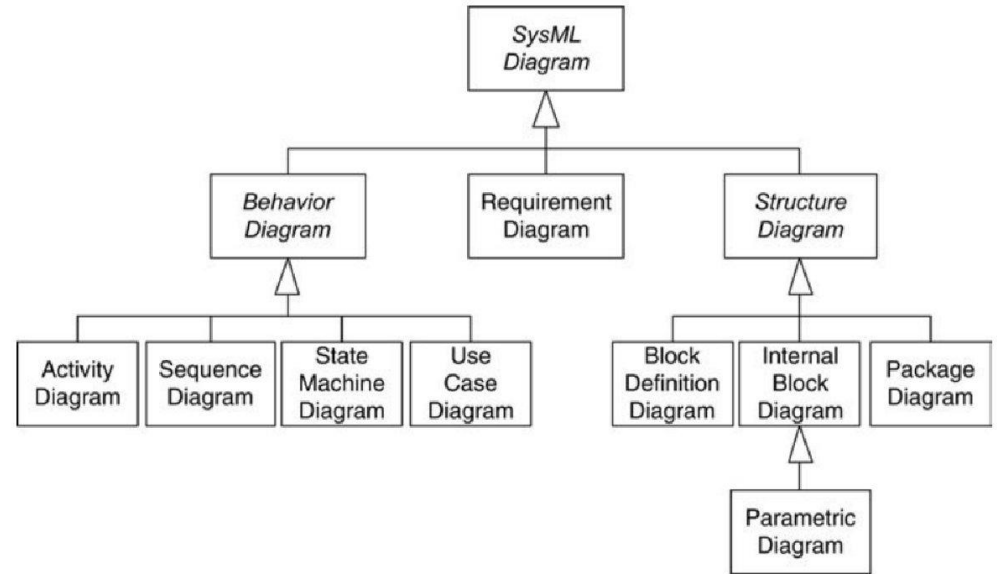
- Napoleon Bonaparte, First MBSE practitioner, did some other stuff too





# SysML – The systems modeling language

- Graphical Language
- Profile (extension) of UML
  - Block vs. Class
  - Requirements
- Systems focus ↔ Software focus
- 9 types of diagrams



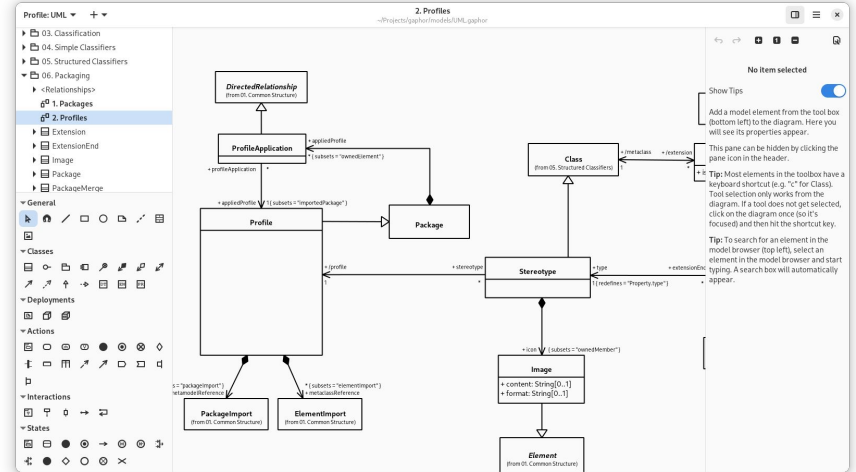
# “Haven’t we tried this before?”

- *Isn’t this UML all over again? Are the 90’s back?*
- Observations in the field
  - Miro board proliferation
  - Block diagrams map well to informal sketches
- A “bad” model is still good documentation
- Software architecture is systems engineering
- Every developer is an architect



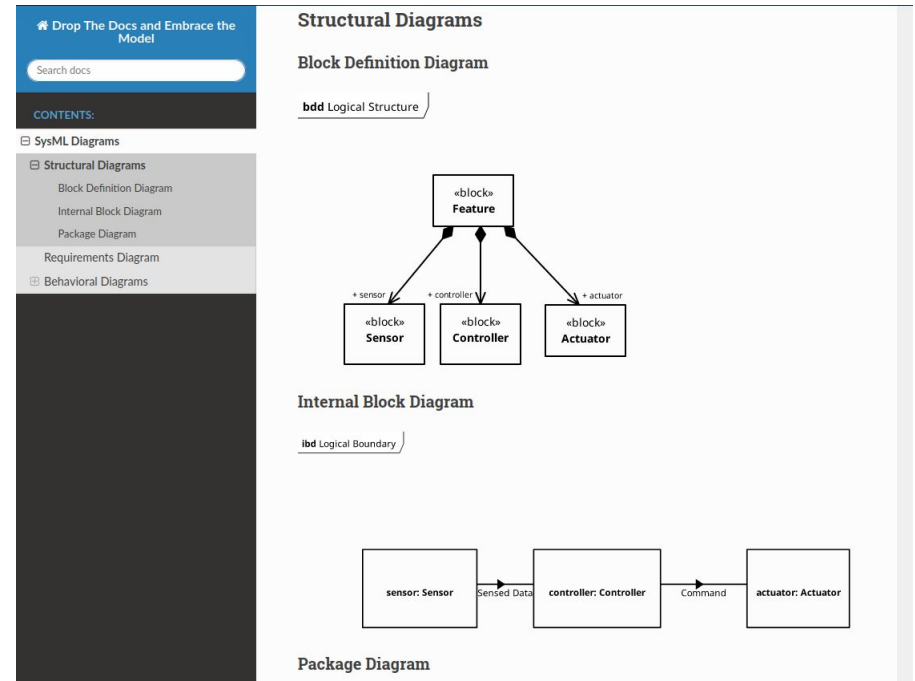
# Gaphor

- Multi-platform graphical modeling application
- Python + GTK
- Multiple modeling languages
  - UML
  - SysML
  - RAAML
  - C4
- Apache 2 licensed
- Extensible
- Not affiliated with the project, just a fan!



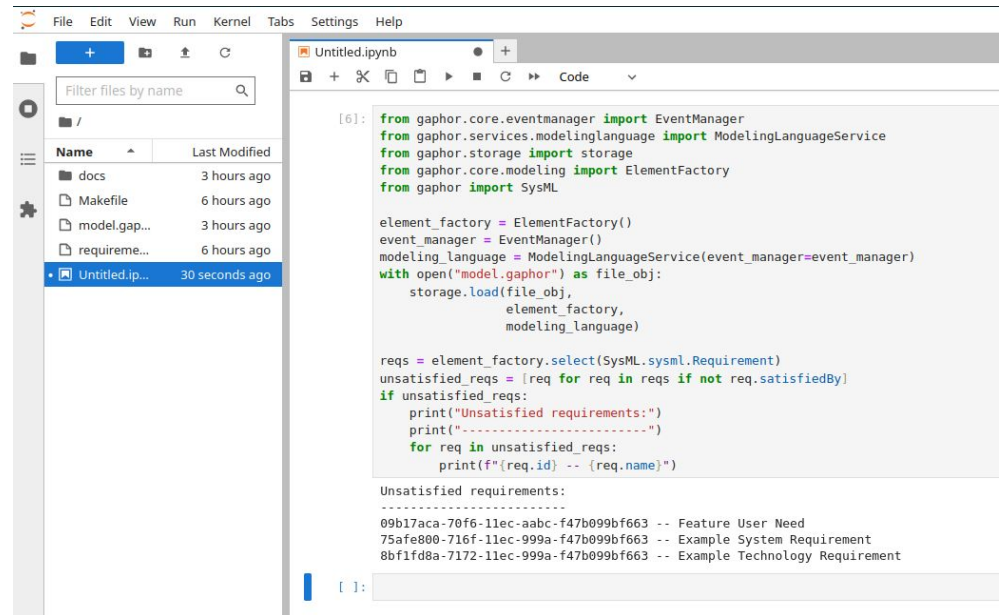
# Gaphor as a Documentation tool

- Sphinx integration
  - Get diagrams straight from model
  - Integration with CI → Automatically updated diagrams in documentation
- Communicate architecture to engineering



# Jupyter Notebooks

- `from gaphor import SysML`
- Interactive programming environment
- Your Model has an API
- Explore your model
- Collaborate



The screenshot shows a Jupyter Notebook window titled 'Untitled.ipynb'. The left sidebar displays a file explorer with a search bar and a table of files. The main area contains a code cell with the following Python code:

```
[6]: from gaphor.core.eventmanager import EventManager
      from gaphor.services.modelinglanguage import ModelingLanguageService
      from gaphor.storage import storage
      from gaphor.core.modeling import ElementFactory
      from gaphor import SysML

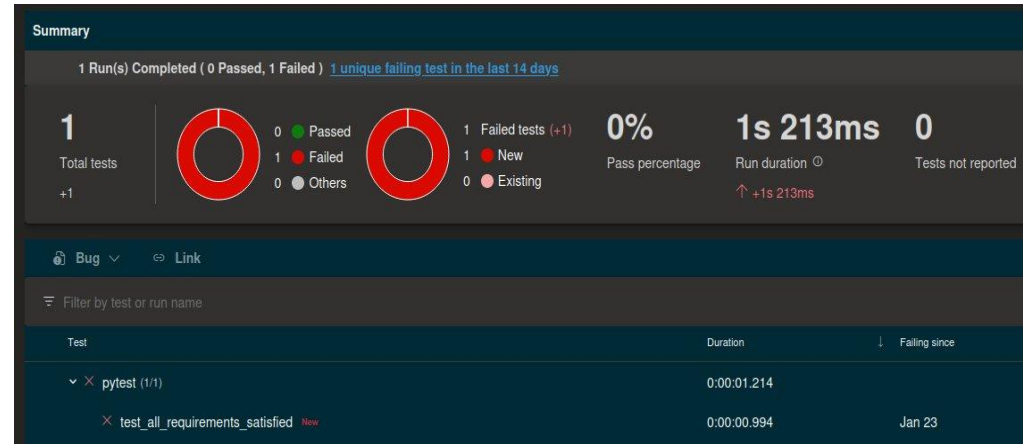
      element_factory = ElementFactory()
      event_manager = EventManager()
      modeling_language = ModelingLanguageService(event_manager=event_manager)
      with open("model.gaphor") as file_obj:
          storage.load(file_obj,
                      element_factory,
                      modeling_language)

      reqs = element_factory.select(SysML.sysml.Requirement)
      unsatisfied_reqs = [req for req in reqs if not req.satisfiedBy]
      if unsatisfied_reqs:
          print("Unsatisfied requirements:")
          print("-----")
          for req in unsatisfied_reqs:
              print(f"{req.id} -- {req.name}")

      Unsatisfied requirements:
      -----
      09b17aca-70f6-11ec-aabc-f47b099bf663 -- Feature User Need
      75afe800-716f-11ec-999a-f47b099bf663 -- Example System Requirement
      8bf1fd8a-7172-11ec-999a-f47b099bf663 -- Example Technology Requirement
```

# Testing your model

- It has an API → You can test it!
- Gaphor + Pytest + CI
- Examples
  - Are all requirements satisfied
  - Does every block have a reason to be there?





# Q & A

<https://vanbever.me/gaphor-presentation-playground/>

<https://github.com/frankvanbever/gaphor-presentation-playground>





## Division of **Essensium nv**

Arenberg Science Park  
Gaston Geenslaan 9  
3001 Leuven  
Belgium  
+32 16 28 65 00

General enquiries  
**[info@mind.be](mailto:info@mind.be)**

Employment enquiries  
**[jobs@mind.be](mailto:jobs@mind.be)**

