

FRAMily Meeting 2026

Workshop 2: Software Tools: Current Development

Dr.-Ing. Niklas Grabbe

Technical University of Munich

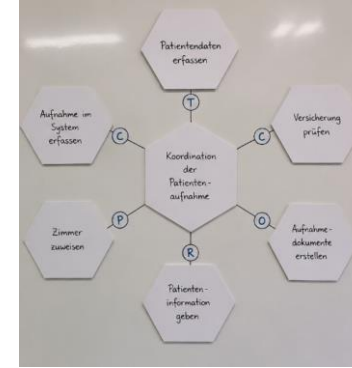
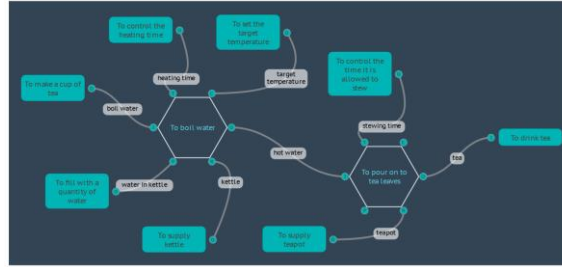
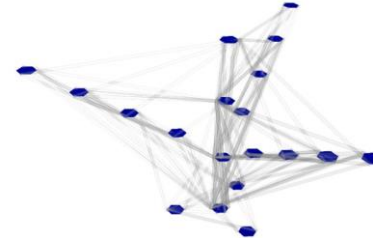
TUM School of Engineering and Design

Chair of Ergonomics

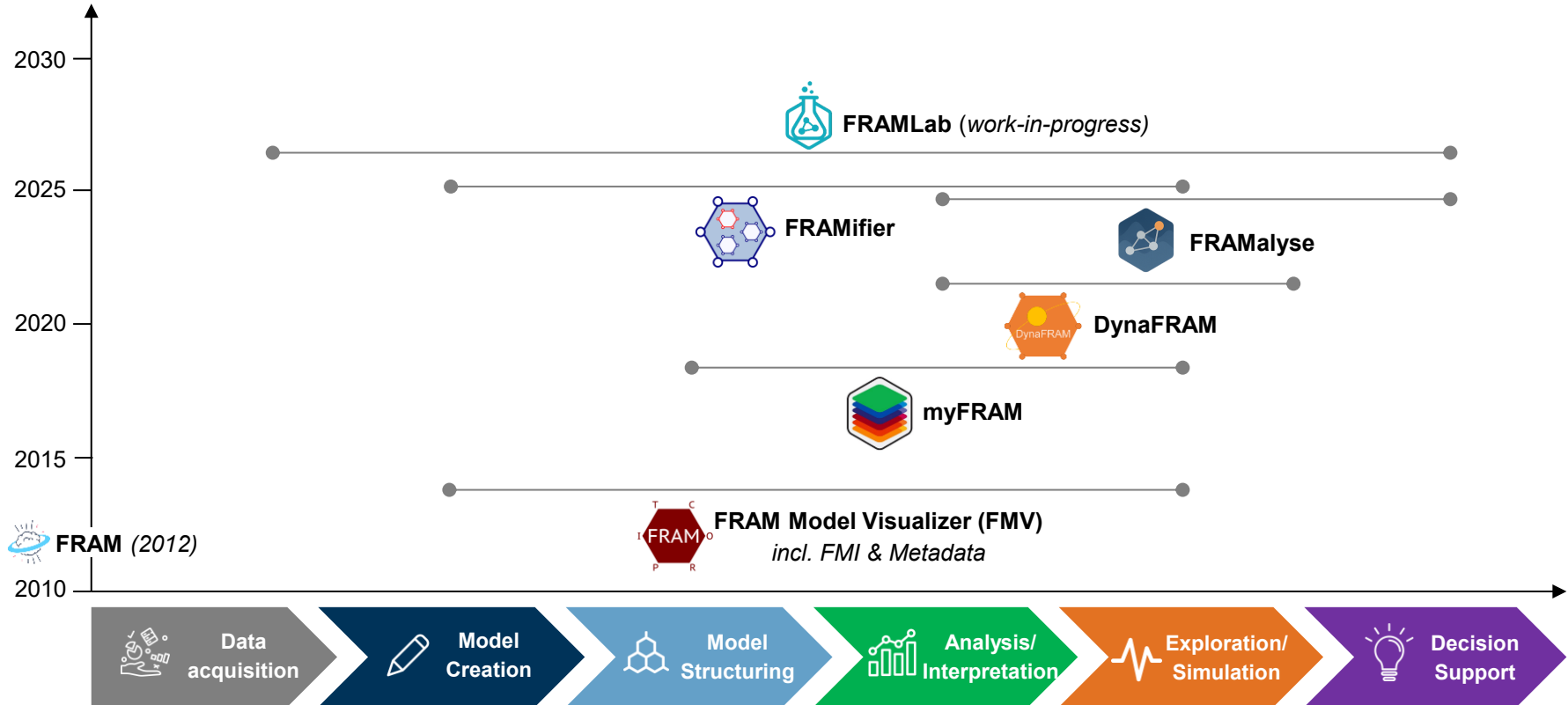
n.grabbe@tum.de



-
- The diagram illustrates a railway network with the following components:
- Stations (Hexagons):**
 - Finish Gading:** Located at the top left, highlighted with a blue border.
 - Manharbour Station:** Located at the bottom left.
 - Close windows:** Located at the top right.
 - Mainharbour Station:** Located at the bottom right.
 - Intermediate Points:** Labeled with letters T, C, P, R, and O, representing various junctions or smaller stations along the lines.
 - Connections and Labels:**
 - loading is completed:** A line connecting the top of the 'Finish Gading' station to the top of the 'Close windows' station.
 - Chain in place:** A line connecting the 'Finish Gading' station to the 'Mainharbour Station'.
 - Clear to call for locomotive shunting:** A line connecting the 'Manharbour Station' to the 'Mainharbour Station'.
 - Arrival of new loads:** A line connecting the 'Mainharbour Station' to the 'Close windows' station.
 - harbour stations are manned:** A line connecting the 'Mainharbour Station' to the 'Close windows' station.

[illegible]

Tool Landscape



FMV: What & Why?



- visual model creation & editing
- syntactic & consistency checking
- cycle-by-cycle interpretation (FMI)
- quantitative simulation via metadata & equations
- instantiation of potential into actual couplings



- verified FRAM model (orphans, missing aspects)
- FMI Log & cycle-by-cycle activation trace
- coloured result bars on hexagons (display)
- exported tables (CSV) & images (SVG)
- scenario instantiations under varying conditions



- Open tool – Community Edition (free)
- Progressive Web App, runs in any modern browser
- Blazor WebAssembly (.NET 8.0), offline-installable
- Pro / Metadata extension via GitHub sponsorship tier
- Available via [GitHub - functionalresonance/FMV_Community_Edition: FMV - FRAM Model Visualiser · GitHub](https://github.com/functionalresonance/FMV_Community_Edition/wiki/FRAM-Model-Visualiser)

Hill R, Hollnagel E (2023) Instructions for use of the FRAM model visualiser (FMV).
https://github.com/functionalresonance/FMV_Community_Edition/wiki/FRAM-Model-Visualiser-Instructions

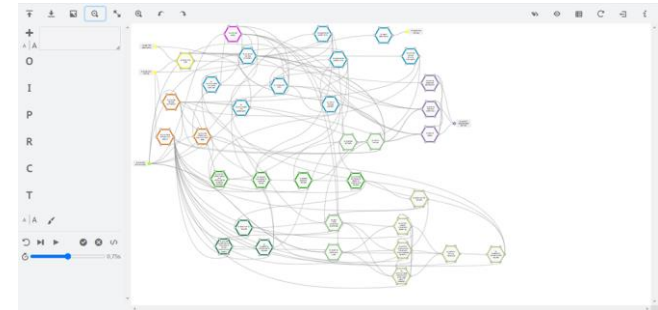
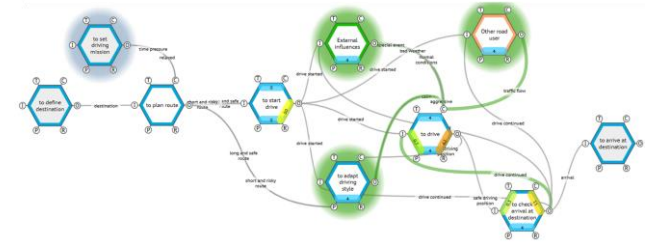
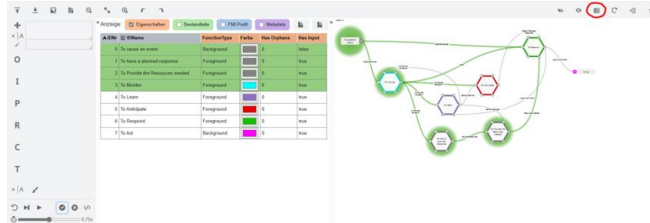


***“From qualitative FRAM models
to dynamic, quantitative
simulation of emergence.”***

FMV: How it works?



- Visual model building
- foreground / background, entry / exit functions
- Orphan & syntax checks during modelling
- FMI – real-time, cycle-by-cycle interpretation of the model
- Interpretation Profiles per function: ALL ($\forall$) · ANY ($\exists$) · NONE ($\emptyset$)
- Metadata (Pro): user-defined key-value pairs + equation parser
- Conditional activation & branching via the Activate key (Plan A / B)
- Tabular view, CSV export, SVG image export, .xfmv project files



Workflow strip:

build functions → add aspects → couple via shared names → run FMI (cycle-by-cycle) → add metadata & equations → simulate instantiations → export (CSV / SVG)

myFRAM: What & Why?



- systematic, data-driven FRAM analysis
- consistency & completeness of large models
- extending FMV's graphical-only approach
- tabular pre- & post-processing of FRAM data
- handling of large-scale models (50+ functions)



- tabular function & aspect definitions (.xlsm)
- automatic completeness check of couplings
- PDF report with model network metrics
- Resilience Analysis Matrix (RAM, $n \times n$)
- FMV-compatible graphical export (.xfrm)



- Open tool – add-in for Microsoft Excel
- Developed in VBA (Visual Basic for Applications)
- Runs on Windows and Mac environments
- Interfaced with FMV via .xfrm export
- Available via Resilience Perspectives - myFRAM

Patriarca, R., Di Gravio, G., & Costantino, F. (2017, December). myFRAM: An open tool support for the functional resonance analysis method. In 2017 2nd International Conference on System Reliability and Safety (ICSRS) (pp. 439–443). IEEE. 10.1109/ICSRS.2017.8272861

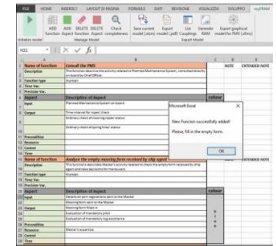
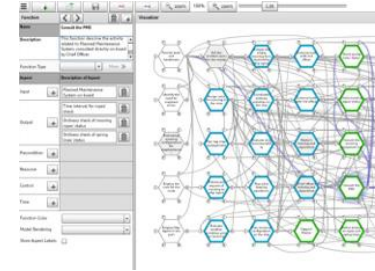
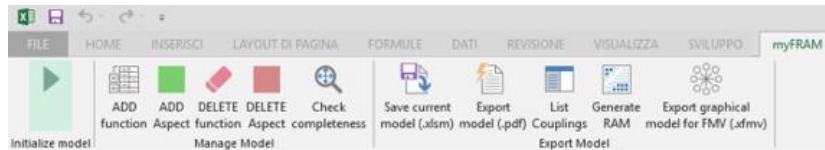


**“From static FRAM diagrams
to systematic, tabular
exploration of complexity.”**

myFRAM: How it works?



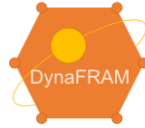
- Excel/VBA add-in – tabular data entry
- Customizable framework: function types, phenotypes, colours
- Iterative add / edit / delete of functions & aspects
- Couplings via shared aspect labels (autocomplete)
- Completeness check – flags inconsistencies in red
- Colour-code by agent / organisation / process
- Resilience Analysis Matrix (RAM): $n \times n$ couplings
- Network metrics: functions, agents, couplings, up-/downstream
- Export to FMV (.xmv) – faster than editing in FMV
- Extensible: Monte-Carlo & Abstraction/Agency-ready



Workflow strip:

initialize framework → add functions (tabular) → define aspects (autocomplete) → check completeness → colour-code by agent → generate reports (RAM / couplings / PDF) → export to FMV (.xmv)

DynaFRAM: What & Why?



- Communication/Explanation
- Visualization
- Create videos/images of scenarios in the context of a FRAM model
- Investigating “work-as-done” variability through data visualization



- Videos of scenario animated over the FRAM model
- Images of the scenario at selected times of the instantiation



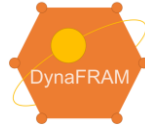
- Open tool
- Stand alone desktop application (Windows only)
- Suggest using Microsoft excel to prepare “scenario files”
- Available via DynaFRAM

Salehi, V.; Smith, D.; Veitch, B.; Hanson, N. (2021). A Dynamic Version of the FRAM for Capturing Variability in Complex Operations. MethodsX. 8(2): 101333. <https://doi.org/10.1016/j.mex.2021.101333>



“Visualizing variability through instantiations of FRAM models”

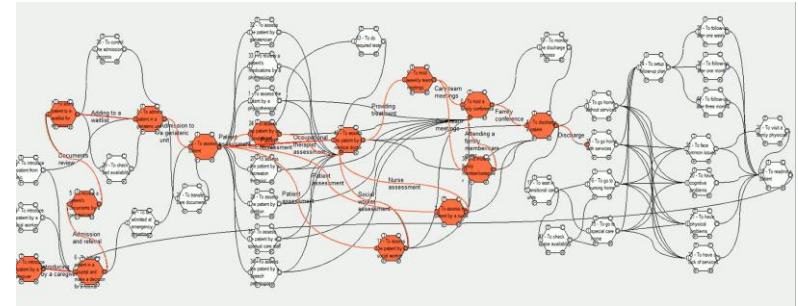
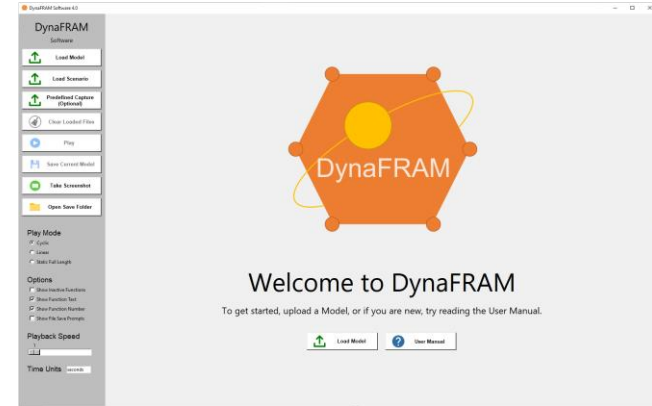
DynaFRAM: How it works?



- Reads models created by the FMV
- Saved videos files of instantiations
- 3 different playback modes
- Display options for text outputs during the video
- Adjustable playback speed
- User manual with brief tutorial/example

Time	ActiveFunction	ActiveFunctionOutput	DownstreamCoupledFunction	CoupledFunctionAspect	TimeTolerance
3	1	No Ice load	4	I	
3	5	Heading = 107.5 Speed = 0 kn	4	I	
10	0	Complete or partial assessment made	0	I	
30	0	Condition update	1	I	
33	2	Med F1 ice class	6	I	
33	7	Class 1C	6	P	
37	1	Medium ice load	4	I	
38	6	RI = -2	4	C	
43	4	Complete or partial assessment made	0	I	
48	0	Ice condition update	1	I	
48	0	Speed change	1	I	
48	0	Ice something	2	C	

Sample Scenario File Format



Workflow strip:

FRAM model in FMV → prepare scenario file (.csv) → upload scenario file → select play settings → play video

FRAMalyse: What & Why?



- structured quantitative analysis
- model evaluation support
- communication of results
- handling of large-scale models



- highly connected functions
- critical variability propagation paths
- structural complexity indicators
- communication-ready summaries
- comparison between model versions



- Open tool
- Free standalone desktop application (Windows 10+)
- Developed using Matlab App Designer
- Available via the Software Tool: FRAMalyse - Chair of Ergonomics

Grabbe, N., & Du, Y. (2025). FRAMalyse: a supporting open tool to analyse and evaluate the characteristics of models derived by the Functional Resonance Analysis Method. In Proceedings of the 35th European Safety and Reliability Conference & the 33rd Society for Risk Analysis Europe Conference (ESREL & SRA 2025), Stavanger, Norway. <https://rpsonline.com.sg/proceedings/esrel-sra-e2025/html/ESREL-SRA-E2025-P5512.html>

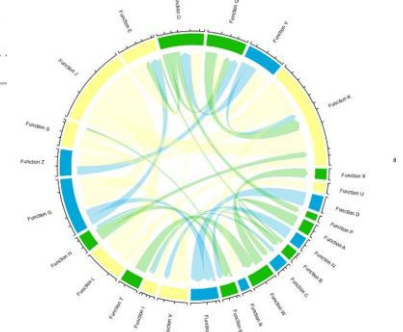
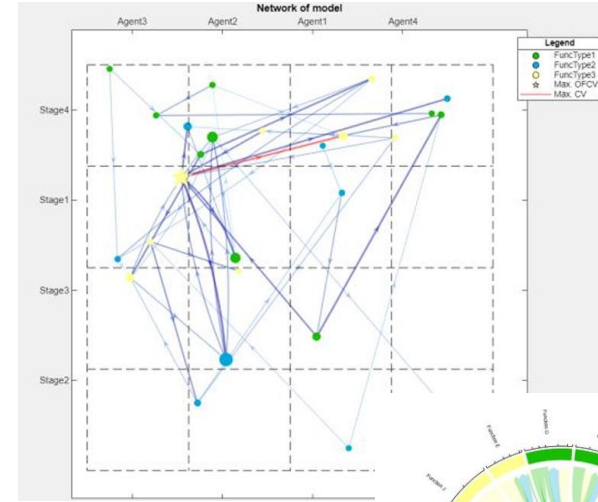
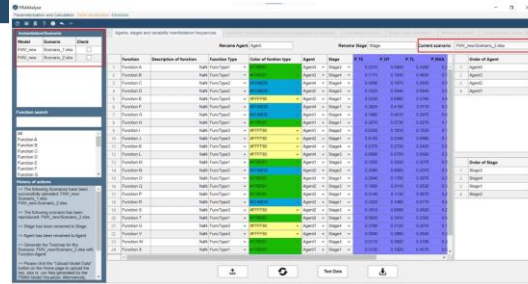


“From visual models to actionable analytical insights.”

FRAMalyse: How it works?



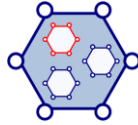
- Quantitative model analysis
- Functional characteristic evaluation
- Structured interpretation support
- Visualization of model metrics
- Analysis of large-scale FRAM models
- Communication-oriented outputs



Workflow strip:

FRAM model in FMV → import → compute metrics → visualize → interpret

FRAMifier: What & Why?



- Intuitive FRAM model builder
- Functionality without fuss
- Build abstraction levels (muti-level)
- Variables and expression (metadata) functionality
- Dynamic simulation monitor



- Navigate easily between abstraction levels
- Verifiable activation logic for (instantiated) FRAM models based on model and expressions



- Open tool
- 100% HTML + CSS + Javascript
- Available via [GitHub](https://github.com/pwgbots/framifier) - <https://github.com/pwgbots/framifier>

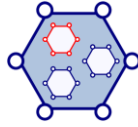
Bots, P. W. G., & Adriaensen, A. (2025). *Semantical challenges in FRAM while developing the FRAMifier*. 17th FRAMily and 7th International Workshop on Safety-II in Practice.

P. W. G. Bots, *FRAMifier: A graphical editor and simulation tool in support of the Functional Resonance Analysis Method*, GitHub, 2025. [Online]. Available: [FRAMifier GitHub repository](#)

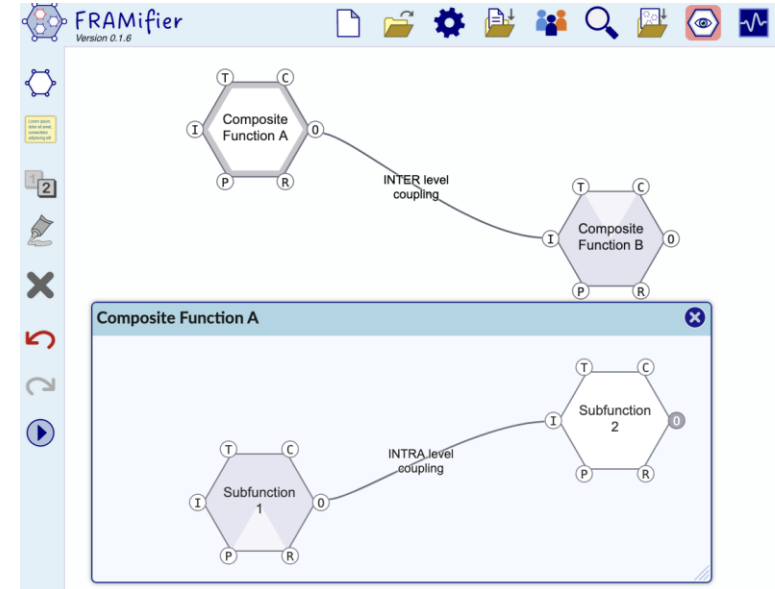
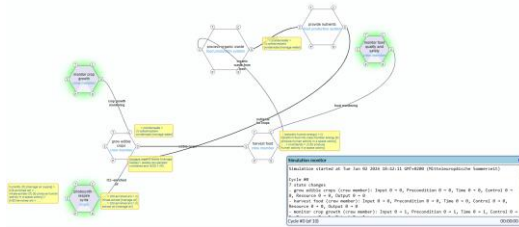


“FRAMifier supports FRAM modelling by making the modelling process visible, constrained, and scalable across abstraction levels.”

FRAMifier: How it works?



- Add functions, couplings and coupling names
- Drag and drop subfunction / composite functions
- FRAMifier will present you the available aspects at
- available levels by constraining logic
- Use the subfunction / composite function viewer
- Build expressions to create the activation logic and use the dynamic simulation monitor



Workflow strip: build functions → draw aspect arrows → add coupling names → add abstraction hierarchy (subfunctions and composite functions) → add metadata (expressions w/ variables & operators) → simulation monitor for dynamic models → optional: export (PNG/SVG)

FRAMLab: What & Why?



work-in-progress



- next-generation FRAM platform — built on FMV
- fully rewritten codebase for compute & scalability
- dynamic simulation beyond pure cycle-based logic
- advanced analytics: Monte-Carlo, metrics, what-if
- open science: modular, extensible, community-friendly



- Abstraction/Agency & Space-Time/Agency matrices
- what-if comparison of multiple instantiations (WAI vs WAD)
- Monte-Carlo distributions & sensitivity insights
- path visualisations beyond direct couplings
- real-time API in-/output to other tools & simulators → Digital Twin



- Open-source research platform (extends FMV sandbox)
- Browser-based — increased computational capacity
- Backwards-compatible with existing FMV models
- Developed at TUM Chair of Ergonomics
- Preview: <https://fmv.benjaoued.de/>

Citation



**“From a static FRAM visualiser
to a dynamic, computational
FRAM laboratory.”**

FRAMLab: How it works?



work-in-progress

Modelling & structure:

- parent-child grouping · Abstraction/Agency & Space-Time matrices
- macros · module libraries · submodels · annotations

Expressions & logic:

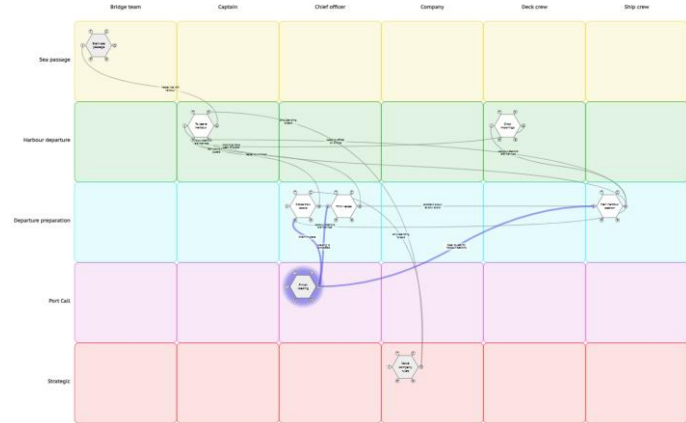
- ifelse, switch/case · loops · arrays/dictionaries
- stateful variables · queuing · error handling · code comments

Simulation & analytics:

- time-based simulation, not only cycle-based · Monte-Carlo
- static & dynamic metrics · path visualisation · diff viewer

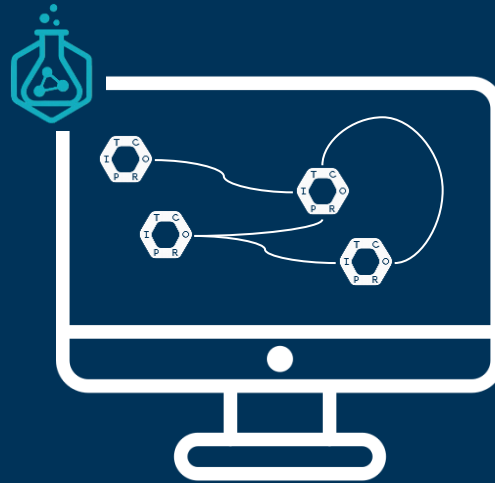
Visualisation & I/O:

- individual 6-bar hex displays · generic plots · what-if · WAI vs WAD overlay
- real-time API · session save · search/filter · key autocomplete



Workflow strip: import / build model → organise via matrices & groups → extend with macros & expressions → run time- or cycle-based simulation → Monte-Carlo & what-if → visualise paths · compare · export via API

Interactive FRAMLab Session





Workshop: FRAMLab Feature Wishlist

What would you want FRAMLab to do next? — 5–6 groups · 30 minutes total

Timeline (30 min)

3 min

Setup

groups by row · grab a flipchart · read task

8 min

Phase 1 — workflow

features within your assigned workflow

7 min

Phase 2 — free

anything else — cross-cutting / wild ideas

10 min

Pitches

≈ 1.5 min per group — Top 3 features only

2 min

Wrap-up

hand in flipcharts



Your task

- Phase 1 — silently write features for your workflow (yellow post-its)
- Phase 2 — anything goes: cross-cutting or wild ideas (pink post-its)
- Place each post-it directly into the matrix where it belongs
- Pick your Top 3 across both phases — then decide who pitches

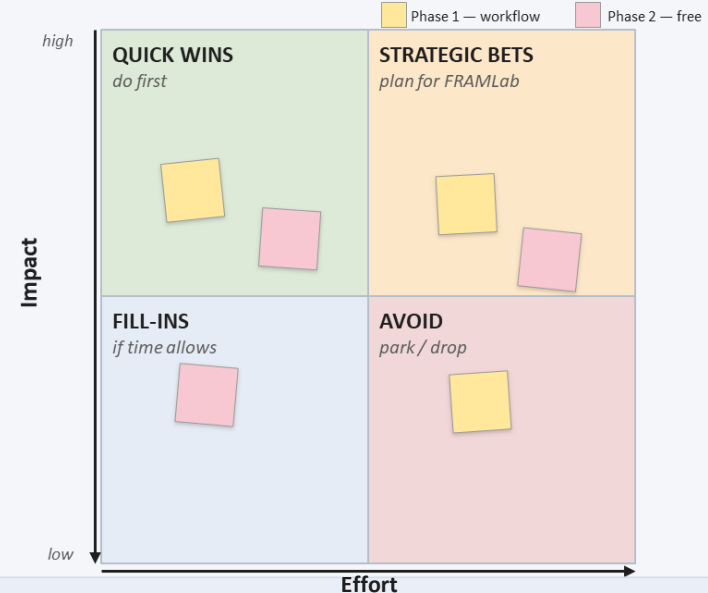


Workflow groups: ① Data acquisition ② Model Creation ③ Model Structuring ④ Analysis / Interpretation ⑤ Exploration / Simulation ⑥ Decision Support

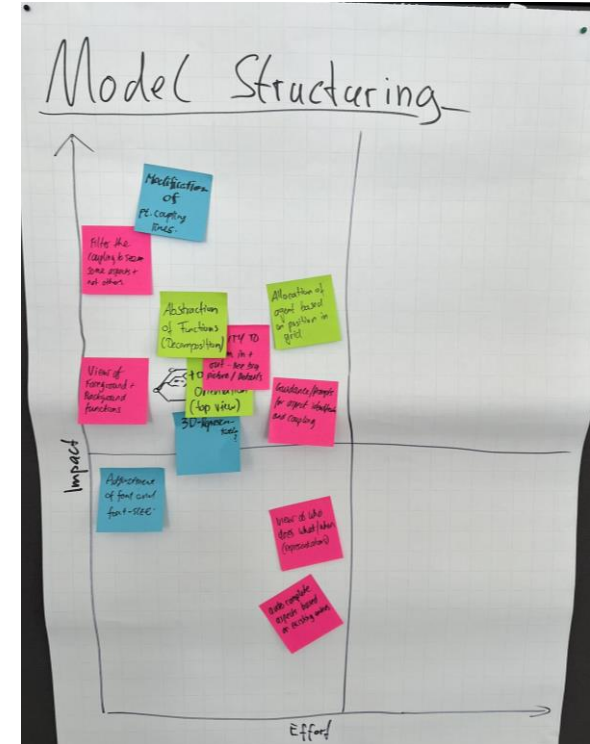
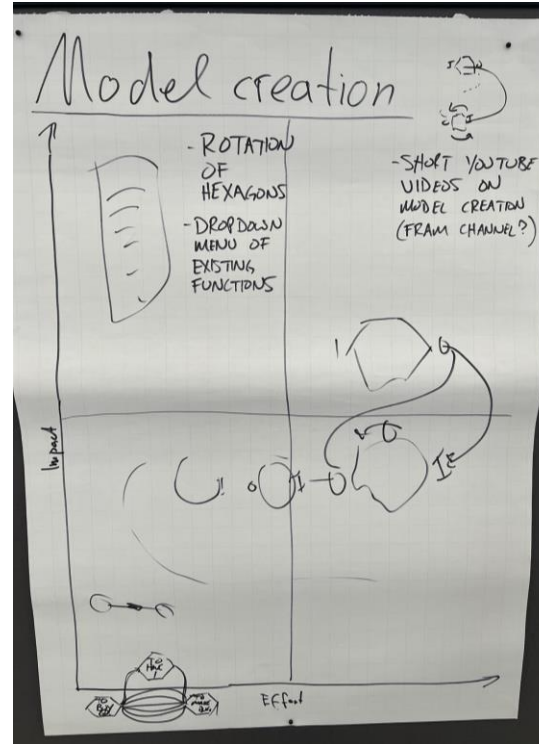
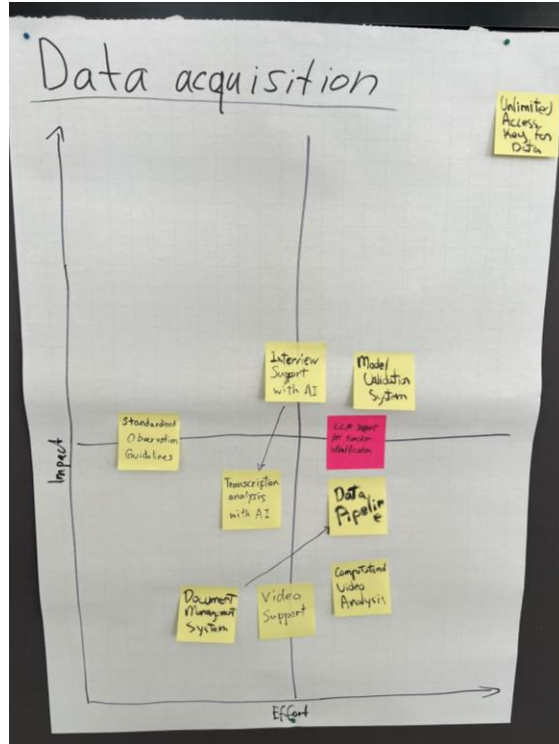


Impact × Effort matrix

Place post-its where they belong — then pick your Top 3

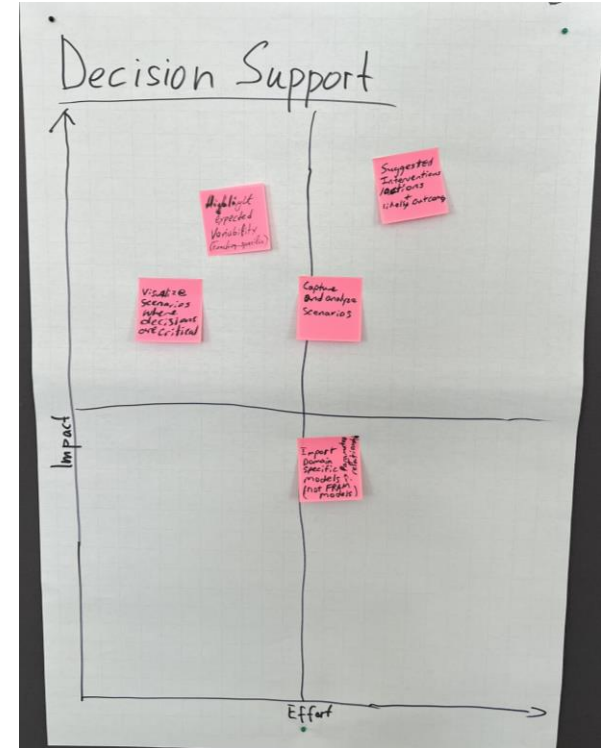
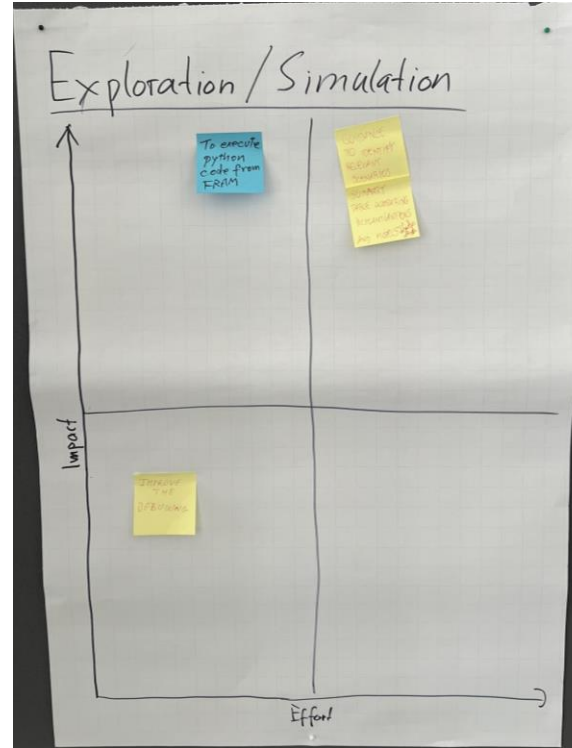
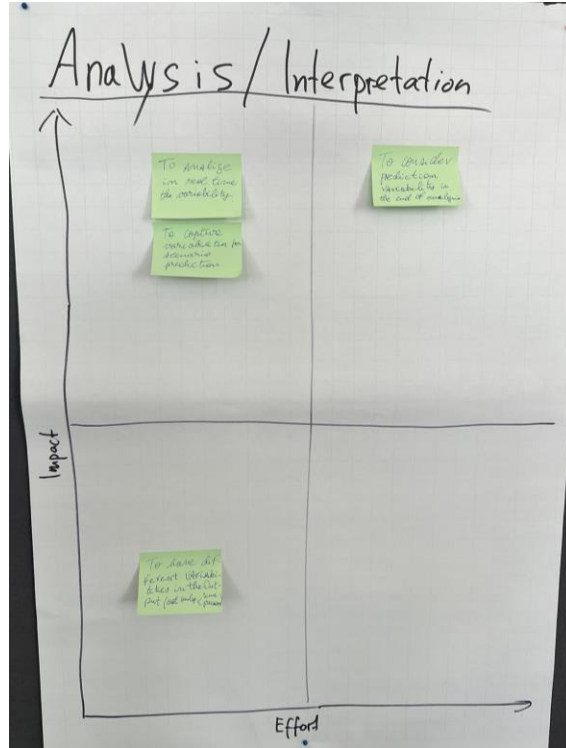


Workshop: FRAMLab Feature Wishlist - Results





Workshop: FRAMLab Feature Wishlist - Results



Discussion & Reflection

