

Hands-On: Model-Driven Optimization with MOMoT

Quick-start card — download the workspace, run the Stack search, run the Blockly Maze demo

FAME 2026 · University of Málaga · Thursday, 1 October 2026, 12:00–13:00
Stefan Klikovits · JKU Linz, Institute of Business Informatics – Software Engineering ·
stefan.klikovits@jku.at · <https://klikovits.net>



github.com/hadiDHD/fame-26-mdo-workspace

1 Get the workspace and start Eclipse DO THIS BEFORE THE SESSION

1. **Install Docker** (Docker Desktop on Windows/macOS, Docker Engine on Linux) and make sure it is running — <https://docs.docker.com/get-docker/>
2. **Clone the repository** and start the container (the first start downloads a large image):

```
git clone https://github.com/hadiDHD/fame-26-mdo-workspace.git
cd fame-26-mdo-workspace
./run.sh # macOS / Linux (stop later: ./stop.sh)
powershell -ExecutionPolicy Bypass -File .\run.ps1 # Windows (stop later: .\stop.ps1)
```

3. **Open Eclipse in your browser:** <http://localhost:6080> — wait until the workspace has loaded. The **Package Explorer** must list `...examples.stack`, `...modularization.jsme` and `blocky_model / blocky_game / blocky_momot`.

2 Run the Stack example (load balancing)

1. In the **Package Explorer**, open the project `at.ac.tuwien.big.momot.examples.stack`.
2. Select the search specification `src/at/ac/tuwien/big/momot/examples/stack/StackSearchExample.momot`.
3. **Right-click** it → **Run As** → **MOMoT Search** (the MOMoT entry; the project also ships a ready-made `StackSearchExample` run configuration).
4. Watch the **Console**: every algorithm and run prints its start and end, then the indicator tables and the solutions found.
5. When it finishes, select the project and press **F5** (Refresh), then open `example/output/ : overall_objectives.pf` (objective values), `overall_solutions.txt` (the rule sequences), `analysis.txt`, `models/`.

Five stacks in a ring with loads 1, 7, 3, 9, 5 — minimise the standard deviation of the loads *and* the number of rule applications. How few moves make all stacks equal?

3 Run the Blockly Maze demo

1. In the **Package Explorer**, select the project `blocky_momot`.
2. **Right-click** → **Run As** → **Blockly Maze** — the JavaFX maze game opens in the container desktop. (If the entry is not offered: use the **Run** ▼ toolbar drop-down → **Blockly Maze**, or right-click the desktop background → **Blockly Maze**.)
3. Play a level with the graphical blocks; use the **Model** controls in the app to save the solution as an `.xmi` model.
4. To let MOMoT synthesise a program, right-click `blocky_momot/blocky.momot` → **Run As** → **MOMoT Search**. Results land in `blocky_momot/output/` (`objectives.pf`, `solutions.txt`, `models/`).
5. Load a result model back into the game through the **Model** controls and watch the synthesised program solve the maze.

! If something does not work

<code>./run.sh / run.ps1</code> fails	Docker is not running — start Docker Desktop / the Docker service and retry.
Blank page, or port 6080 in use	Wait 30–60 s after the container starts and reload; stop other containers with <code>./stop.sh / .\stop.ps1</code> .
Changes have no effect	Save the file (Ctrl/Cmd+S) and wait until the previous run has finished.
Output files missing	Select the project and press F5 to refresh.
"Could not determine Java class"	Do not give the search block a name — write <code>search = {}</code> .