Ansys

WOST

WORKSHOP 2022

Webservice based Framework for Automated Modular Electric Drivetrain Simulations

24.06.2022

ZF Friedrichshafen AG, Helmut Schmid

Ansys

Preliminary Considerations

Look at this powerful tool…

… a strictly modular system
… high quality from Switzerland
… very compact size
… the ideal tool for a wide range of work

Just kidding!

In fact, that's a toy, not a serious tool.

It's like a duck: a duck can walk, swim and fly – but nothing very well

Preliminary Considerations

The MODULAR approach…

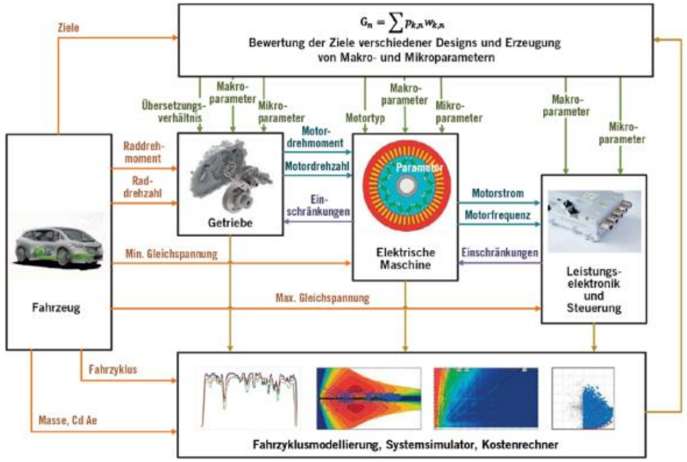
… like the Metabo slogan: “Work. Don’t play”
… thus, if you need a screw-driver, use a screw-driver
… we say: one task, one tool

And this is now a truly modular, anytime extendable toolset:
just pick out what you really need and don’t care about the rest

System Simulation & Optimization

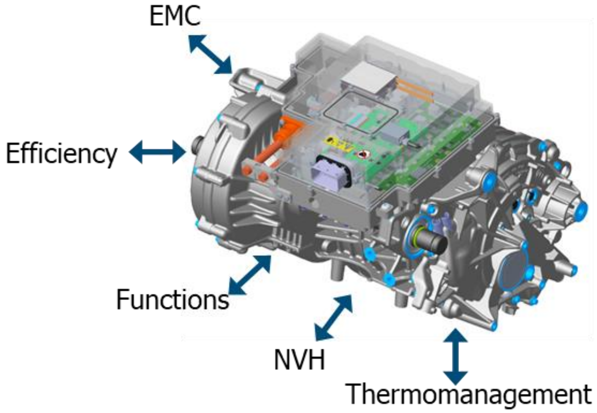
- Why

- **No holistic system optimization possible**, only not connected component simulations
- Prediction based on experience of experts and single simulations
- **Reducing costs, increasing efficiency & speed up development time**
- HPC ready



- How

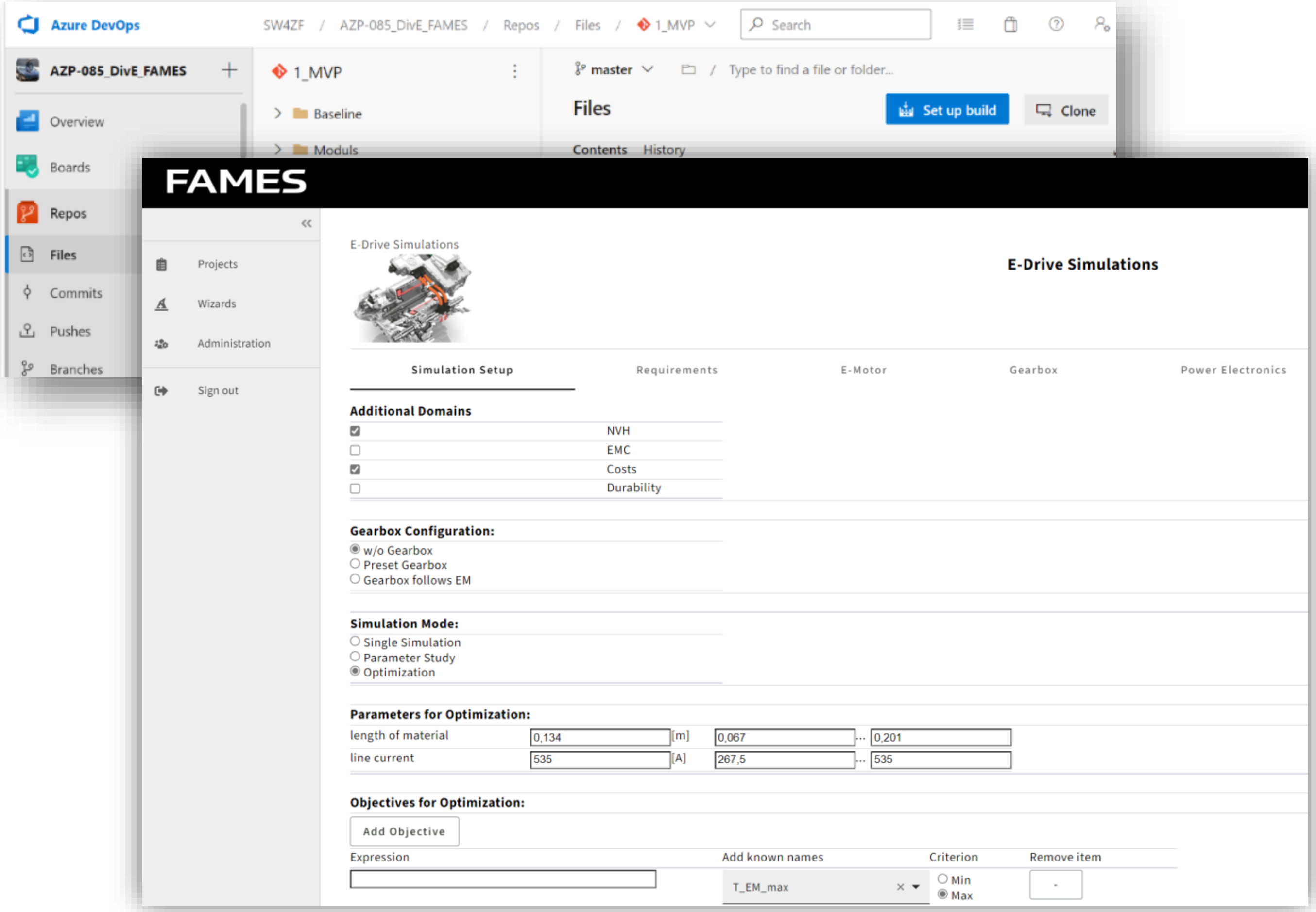
- Improving the system simulation
- Usage of multiphysics Simulation
  → Combination of simulations on System level, involving more than one physical discipline
- **Build up a modular Framework**



FAMES

(**F**ramework for **A**utomated **M**odular **E**lectric drivetrain **S**imulations)

We focus strongly on modular simulation workflows:



E-Drive Simulations

E-Drive Simulations

Simulation Setup | Requirements | E-Motor | Gearbox | Power Electronics

Additional Domains

☑ NVH
□ EMC
☑ Costs
□ Durability

Gearbox Configuration:

● w/o Gearbox
○ Preset Gearbox
○ Gearbox follows EM

Simulation Mode:

○ Single Simulation
○ Parameter Study
● Optimization

Parameters for Optimization:

length of material	0,134	[m]	0,067 ...	0,201
line current	535	[A]	267,5 ...	535

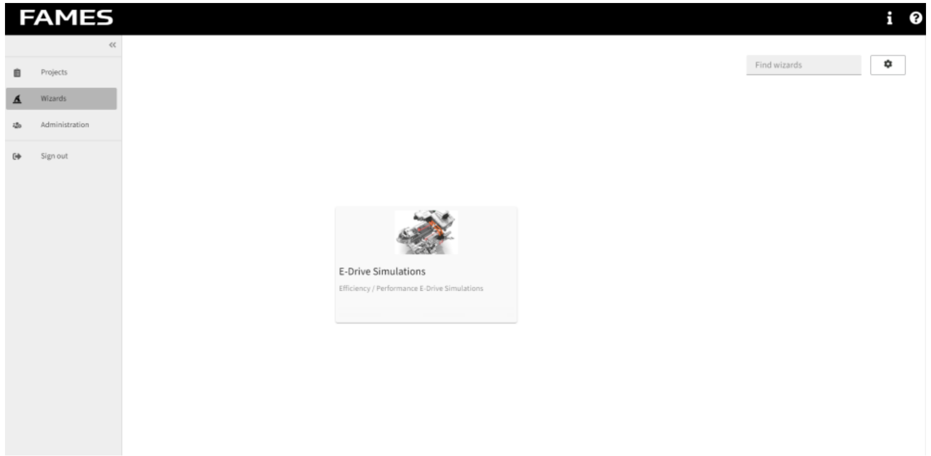
Objectives for Optimization:

Add Objective

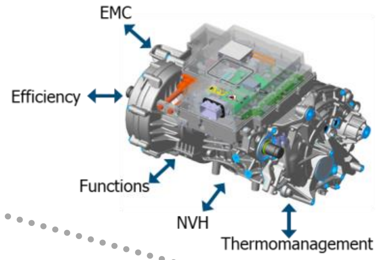
Expression	Add known names	Criterion	Remove item
	T_EM_max	○ Min ● Max	-

Software Architecture & Development

- Running in Browser
- GIT-repository for version control
- Multi-user compatibility
- Ticketing system ready (MS Azure)
- Continuously feature add
- Ready for different locations
- High-performance cluster as backbone
- Python based



- eDrive system optimization
- Component optimization
- Performance, Efficiency
- EMC, NVH, Thermal
- Gearbox design
- A lot more to come!



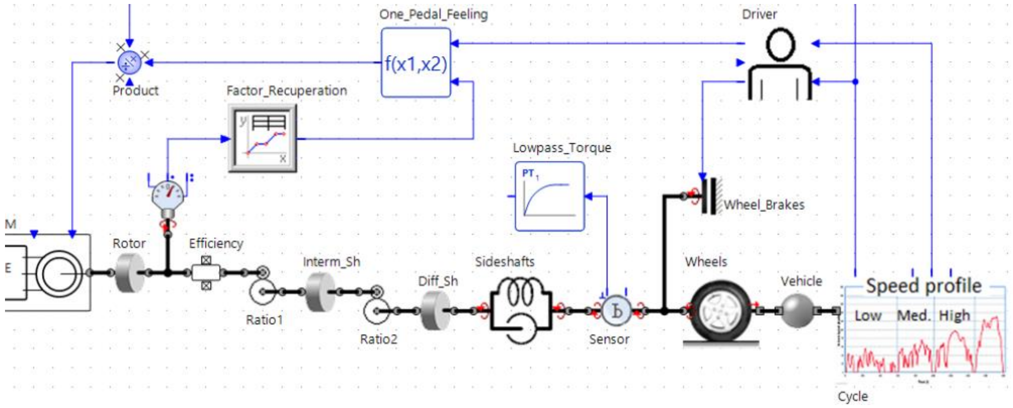
Community:
If you have tools / ideas which can benefit from the framework -> feel free to contact us!

The Modular Approach in FAMES

SimX Library (or other)

Collection of modules like road, battery, bearing losses, …

- Build the model(s) with the modules needed for the current app
- Don’t try to build “complete” models for all purposes: an E-drive doesn’t need an idle-speed controller

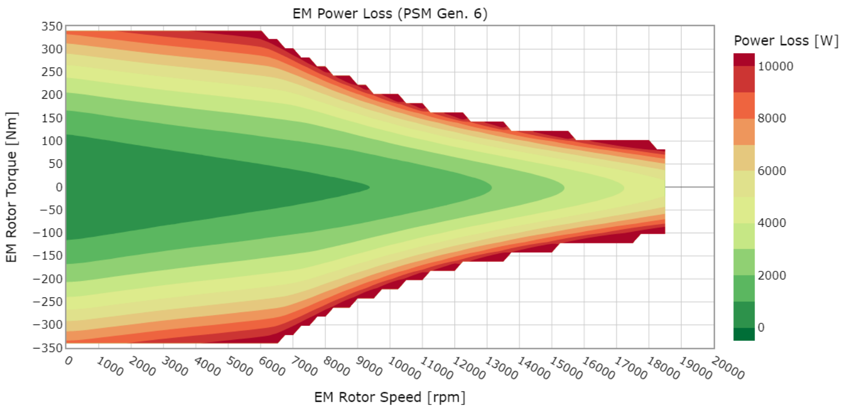


Python Library

Collection of functions for batch runs, plots, reports, …

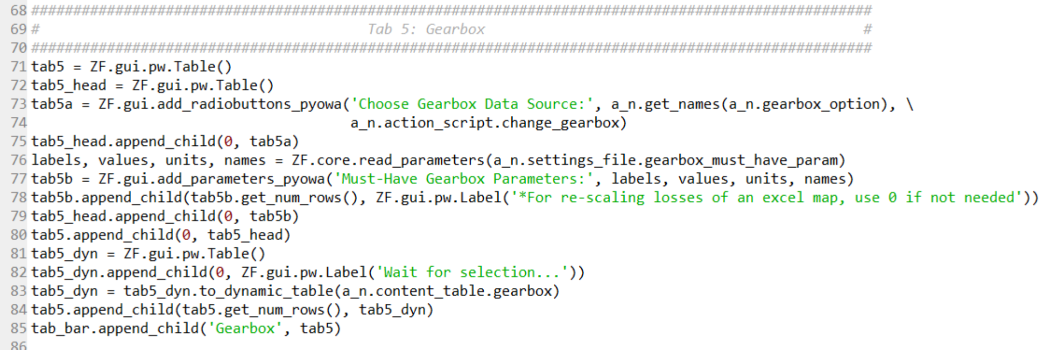
- Build the scripts with the library functions for the current app
- Closed for modifications, open for extensions

```python
def plot_powerloss(P_Loss, str_EM, filename):
  n, T, Pl = ZF.core.get_xyz_from_data_map(P_Loss)
  ZF.plot.prepare()
  ZF.plot.PLOT.colormap = ZF.plot.Colormap.ZF_green_yellow_red
  ZF.plot.contourf(n, T, Pl)
  ZF.plot.title('EM Power Loss (' + str_EM + ')')
  ZF.plot.xlabel('EM Rotor Speed [rpm]')
  ZF.plot.ylabel('EM Rotor Torque [Nm]')
  ZF.plot.zlabel('Power Loss [W]')
  ZF.plot.xlim(0, 20000, 1000)
  ZF.plot.ylim(-350, 350, 50)
  ZF.plot.zlim(0, 10000, 1000)
  ZF.plot.save(filename)
  ZF.plot.close()
```



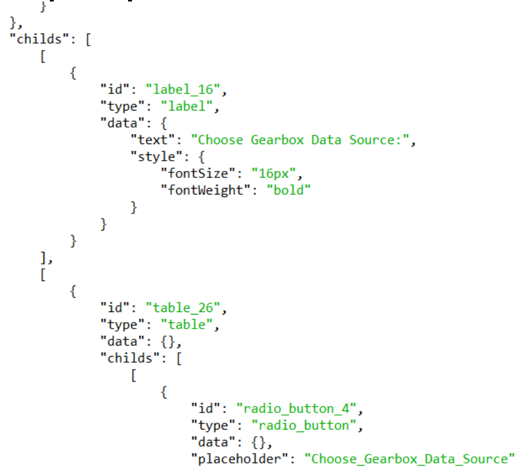
Webapps Code Snippets

Initial Frontend Definition:



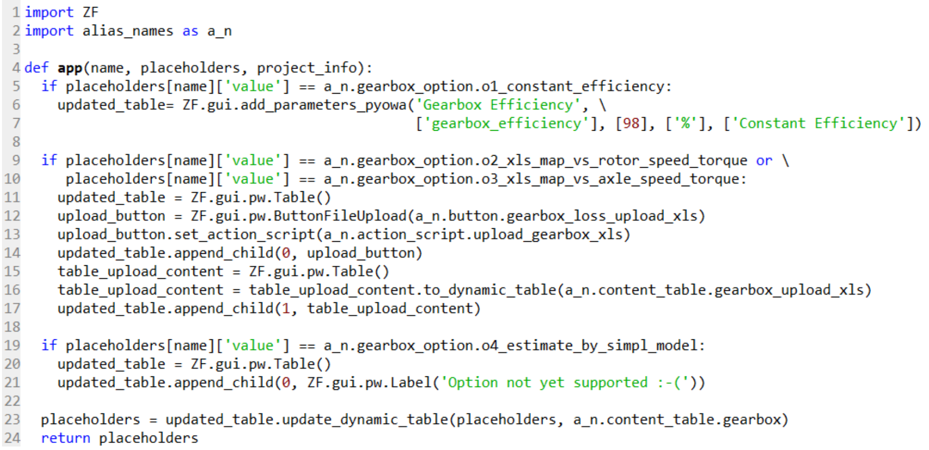
```python
##########################################################################################################
#                                                            Tab 5: Gearbox                                                          #
##########################################################################################################
tab5 = ZF.gui.pw.Table()
tab5_head = ZF.gui.pw.Table()
tab5a = ZF.gui.add_radiobuttons_pyowa('Choose Gearbox Data Source:', a_n.get_names(a_n.gearbox_option), \
                                       a_n.action_script.change_gearbox)
tab5_head.append_child(0, tab5a)
labels, values, units, names = ZF.core.read_parameters(a_n.settings_file.gearbox_must_have_param)
tab5b = ZF.gui.add_parameters_pyowa('Must-Have Gearbox Parameters:', labels, values, units, names)
tab5b.append_child(tab5b.get_num_rows(), ZF.gui.pw.Label('*For re-scaling losses of an excel map, use 0 if not needed'))
tab5_head.append_child(0, tab5b)
tab5.append_child(0, tab5_head)
tab5_dyn = ZF.gui.pw.Table()
tab5_dyn.append_child(0, ZF.gui.pw.Label('Wait for selection...'))
tab5_dyn = tab5_dyn.to_dynamic_table(a_n.content_table.gearbox)
tab5.append_child(tab5.get_num_rows(), tab5_dyn)
tab_bar.append_child('Gearbox', tab5)
```

Output json file:



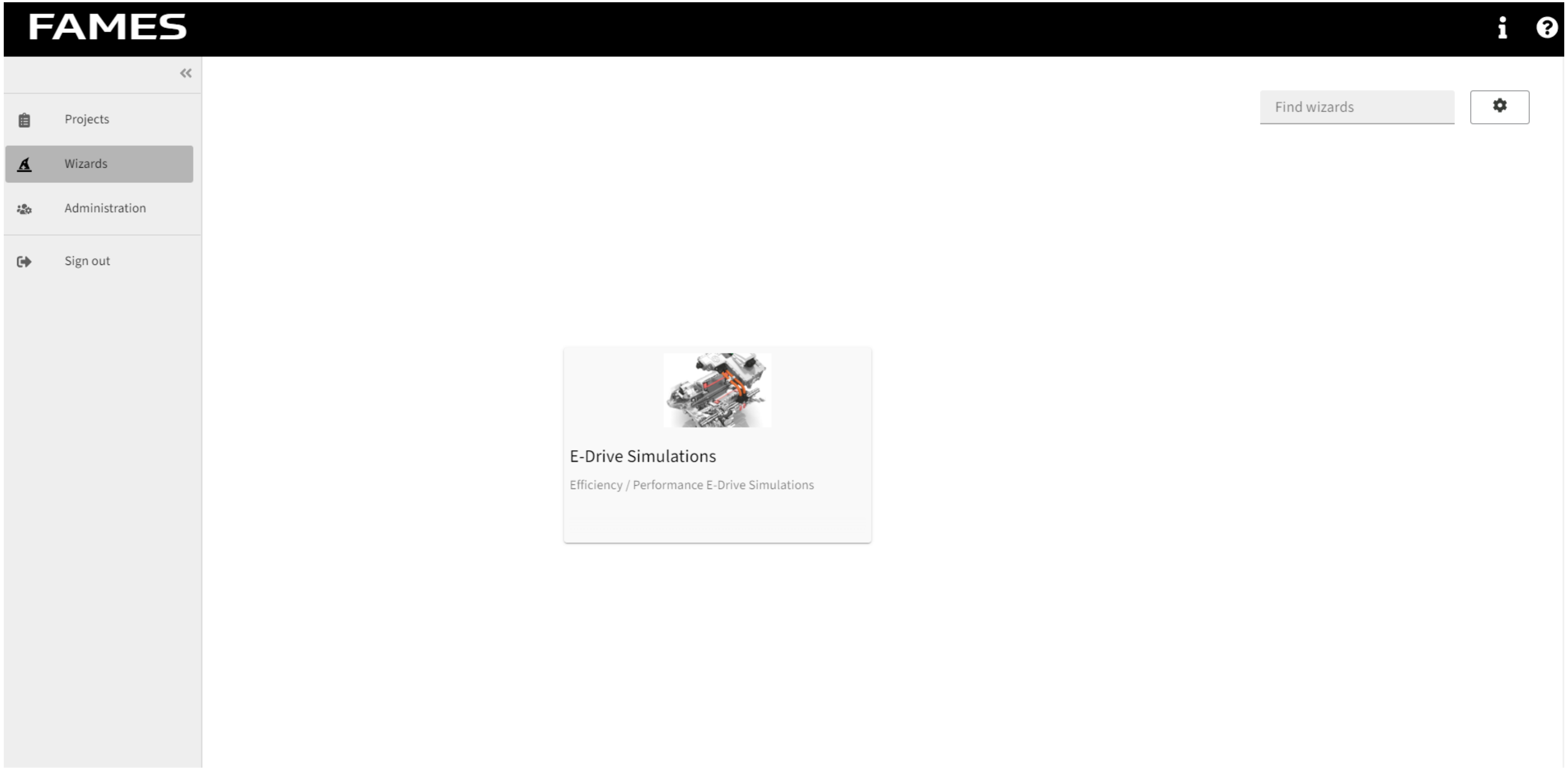
```json
            }
    },
    "childs": [
        [
            {
                "id": "label_16",
                "type": "label",
                "data": {
                    "text": "Choose Gearbox Data Source:",
                    "style": {
                        "fontSize": "16px",
                        "fontWeight": "bold"
                    }
                }
            }
        ],
        [
            {
                "id": "table_26",
                "type": "table",
                "data": {},
                "childs": [
                    [
                        {
                            "id": "radio_button_4",
                            "type": "radio_button",
                            "data": {},
                            "placeholder": "Choose_Gearbox_Data_Source"
```

Callback on change gearbox loss source:



```python
import ZF
import alias_names as a_n

def app(name, placeholders, project_info):
  if placeholders[name]['value'] == a_n.gearbox_option.o1_constant_efficiency:
    updated_table= ZF.gui.add_parameters_pyowa('Gearbox Efficiency', \
                                               ['gearbox_efficiency'], [98], ['%'], ['Constant Efficiency'])

  if placeholders[name]['value'] == a_n.gearbox_option.o2_xls_map_vs_rotor_speed_torque or \
     placeholders[name]['value'] == a_n.gearbox_option.o3_xls_map_vs_axle_speed_torque:
    updated_table = ZF.gui.pw.Table()
    upload_button = ZF.gui.pw.ButtonFileUpload(a_n.button.gearbox_loss_upload_xls)
    upload_button.set_action_script(a_n.action_script.upload_gearbox_xls)
    updated_table.append_child(0, upload_button)
    table_upload_content = ZF.gui.pw.Table()
    table_upload_content = table_upload_content.to_dynamic_table(a_n.content_table.gearbox_upload_xls)
    updated_table.append_child(1, table_upload_content)

  if placeholders[name]['value'] == a_n.gearbox_option.o4_estimate_by_simpl_model:
    updated_table = ZF.gui.pw.Table()
    updated_table.append_child(0, ZF.gui.pw.Label('Option not yet supported :-('))

  placeholders = updated_table.update_dynamic_table(placeholders, a_n.content_table.gearbox)
  return placeholders
```

Live Demo



FAMES

Projects

Wizards

Administration

Sign out

Find wizards

E-Drive Simulations

Efficiency / Performance E-Drive Simulations

Open Discussion

- Further comfort functions welcome (e.g. native integration of a webbased optimization setup).
- Performance improvements (due to numerous write, read and batch operations the start of the actual workflow is relatively long).
- More extensive admin functionalities.
- How to reach an enterprise level (Ansys Dynardo webservice roadmap)?

And last but not least…

…special thanks to Rene Kallmeyer and Stefan Marth for their proactive support!

Ansys

WOST

WORKSHOP