Optimization of multi-carrier energy systems using an FMI-based co-simulation approach

Edmund Widl, AIT Austrian Institute of Technology, Energy Department, Vienna, Austria

Traditional simulation tools and models are typically focusing on only one respective energy domain. They are thus not capable of properly describing multi-carrier energy systems in detail (including their controls), which is an important prerequisite for a suitable design process and optimized operation. Tool coupling approaches (co-simulation) provide a promising alternative, facilitating the detailed assessment and optimization of the interactions between the various domains.

A prototype implementation of a modular and flexible framework is presented, which uses a co-simulation approach to enable a detailed analysis and optimization process. It relies on established methods and tools where available and extends the state-of-the-art where necessary. Furthermore, the applicability of the Functional Mock-up Interface (FMI) specification within this context is demonstrated, which facilitates modularity and flexibility with regard to the utilized models and tools.

1. THE FUNCTIONAL MOCK-UP INTERFACE (FMI)

- The **Functional Mock-up Interface** (FMI) [1] defines a generic but powerful API and model description scheme to encapsulate and link models and simulation tools
- FMI is a non-proprietary, industrial-strength specification, developed by both academia and industry

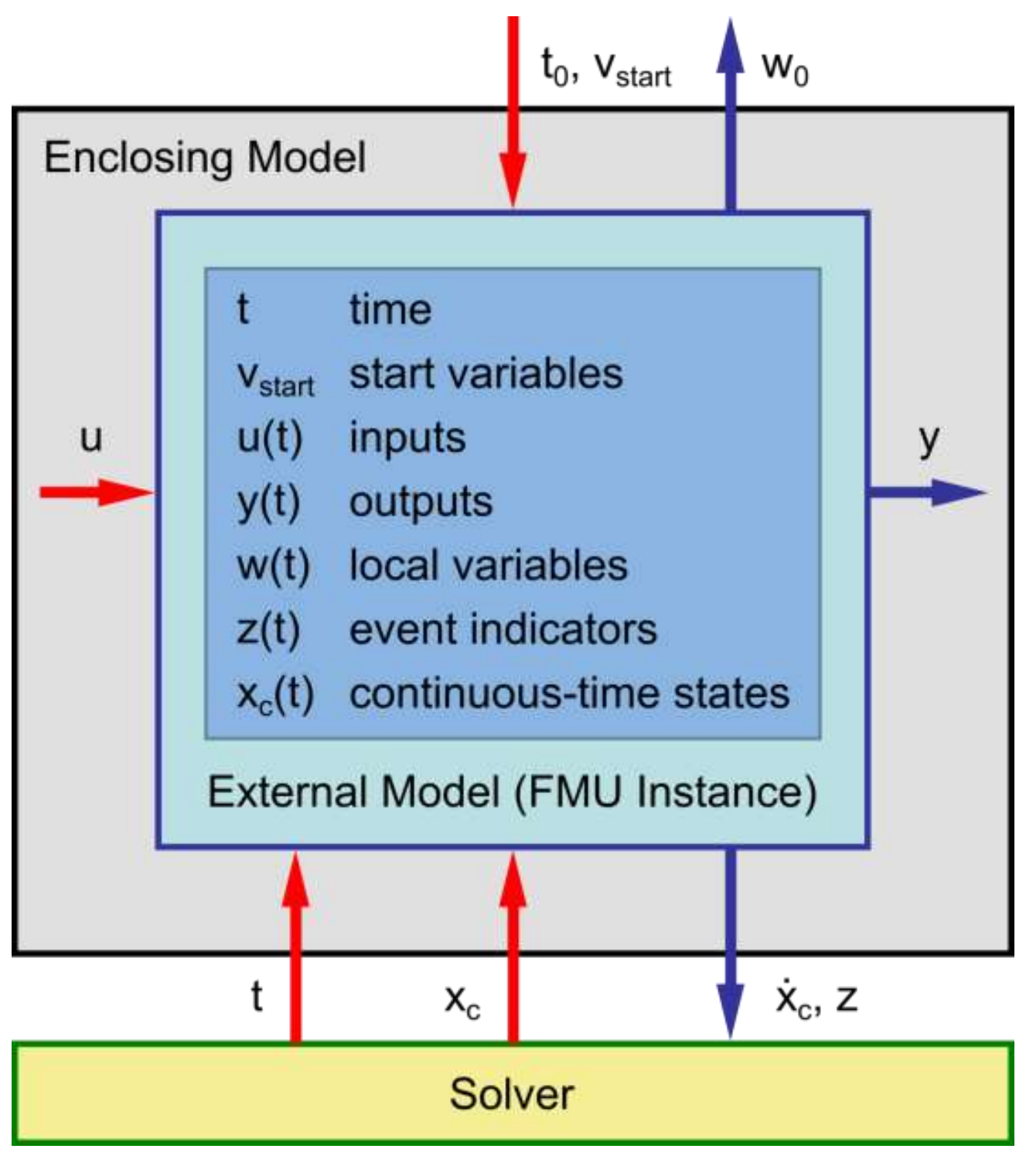

Figure 1: Data flow between the environment and an FMU for Model Exchange

- FMI-compliant simulation components are called **Functional Mock-up Units** (FMU)
- FMI for **Model Exchange**:
  - standardized access to model equations
  - models described by differential, algebraic and discrete equations
  - solved with integrators provided by embedding environment
- FMI for **Co-Simulation**:
  - stand-alone black-box simulation components
  - data exchange restricted to discrete communication points
  - system model is solved by internal solver

3. OPTIMIZATION OF CO-SIMULATION MODELS

Application of co-simulation in the context of design optimization is challenging:

- **no closed (semi-)analytical representation** of the overall system is available
- therefore **no closed (semi-)analytical representation of objective functions** or its derivatives
- prevents the straightforward deployment of many optimization algorithms
- **use instead metaheuristics** that rely solely on the evaluation of the objective function, which can be easily achieved using co-simulation

5. EXAMPLE APPLICATION: HYBRID THERMAL-ELECTRICAL NETWORK

System layout:

- either a **boiler** or a **heat pump** (HP) feeds into a **thermal buffer**, which is connected to the **thermal loads**
- main source of electricity is the **external grid**, but there is also a **PV system** and a **battery** available

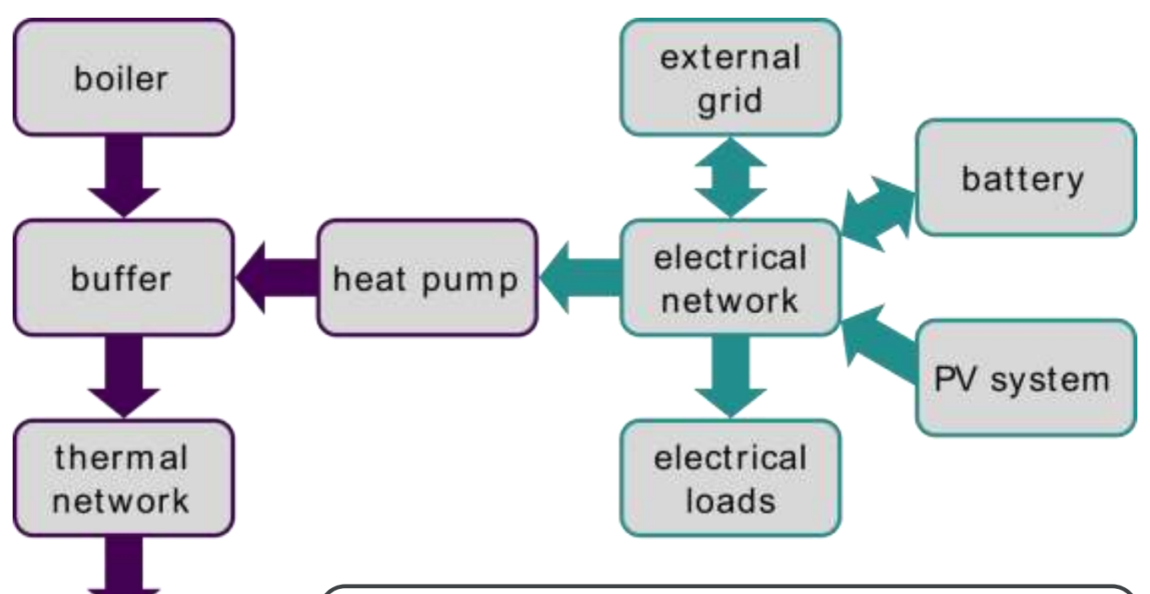

Figure 4: Schematic view of the system with arrows indicating the energy flows.

The boiler, the battery, and the HP are operated with the help of an **energy management system** (EMS):

- **goal**: use local electricity generation from renewable energy sources to **operate the HP** and **reduce the utilization of the boiler**
- in case of PV overproduction or when the battery is sufficiently charged, the EMS prioritizes the HP over the boiler
- charge the battery in case of PV overproduction but no need to operate the HP or enough PV overproduction to have a surplus even if the HP is running

The EMS and the physical system have been **modeled independently**, then **exported as FMUs** and coupled in a **co-simulation model** according to Figure 2.

2. CO-SIMULATION OF MULTI-DOMAIN ENERGY SYSTEMS

The **FUMOLA co-simulation** environment has been used:

- developed on top of the **Ptolemy II** [2] simulation environment
- specifically designed to **support** the **features** offered by the **FMI** specification
- utilizing the **FMI++ Library** for handling FMI-based co-simulation components

FUMOLA: http://fumola.sourceforge.net/
The FMI++ Library: http://fmipp.sourceforge.net/

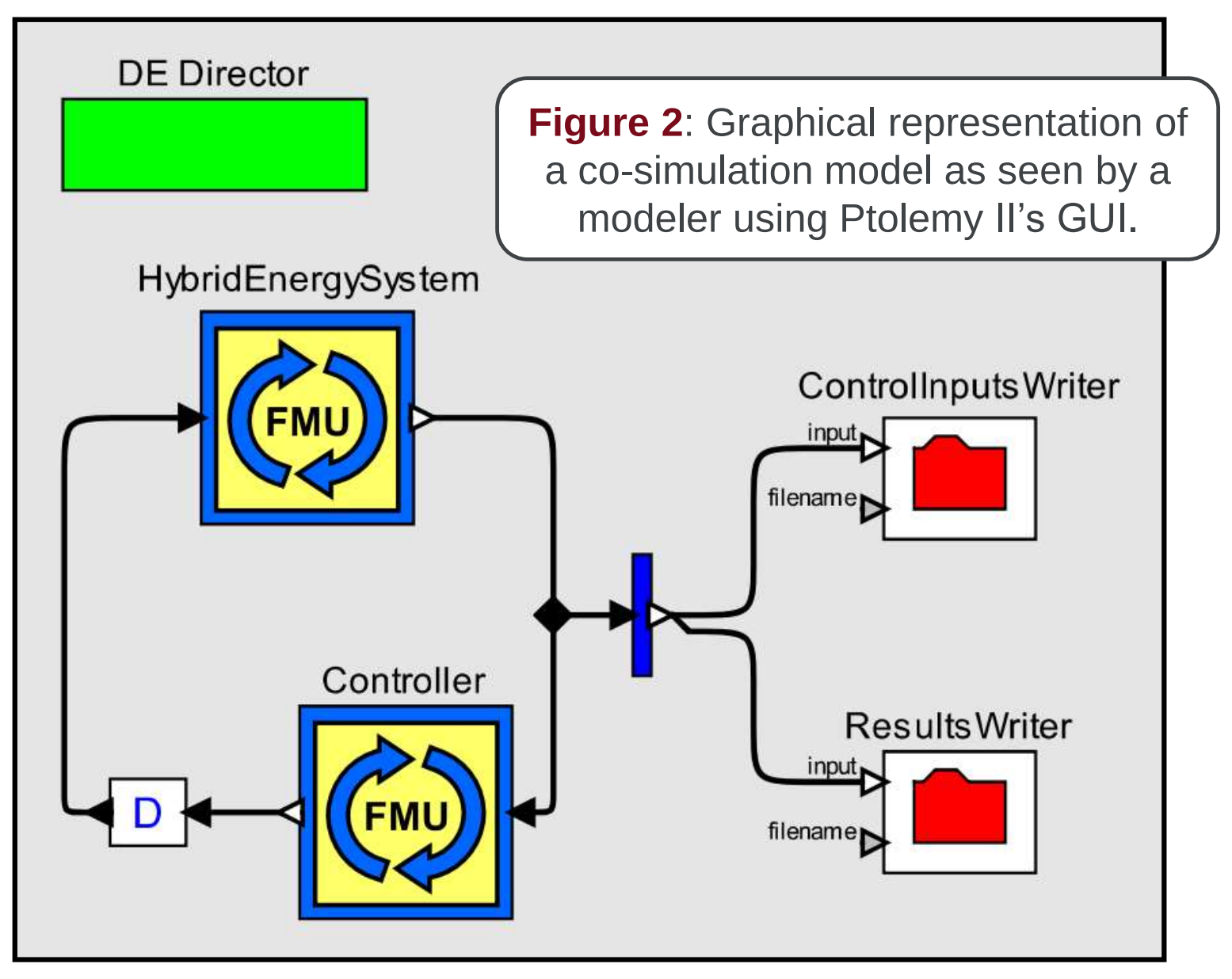

Figure 2: Graphical representation of a co-simulation model as seen by a modeler using Ptolemy II's GUI.

4. THE DIFFERENTIAL EVOLUTION METHOD

- a **metaheuristics** that treats optimization problems as black boxes **without the need of computing derivatives** [3]
- optimizes a problem by maintaining a **population of candidate solutions**
- creates new candidate solutions by combining existing ones according to a simple procedure
- at each iteration, the candidate solution associated to the **smallest value for the objective function** is kept
- implementation used here is an **object-oriented adaptation** of openly available MATLAB code, containing the algorithm in its full functionality

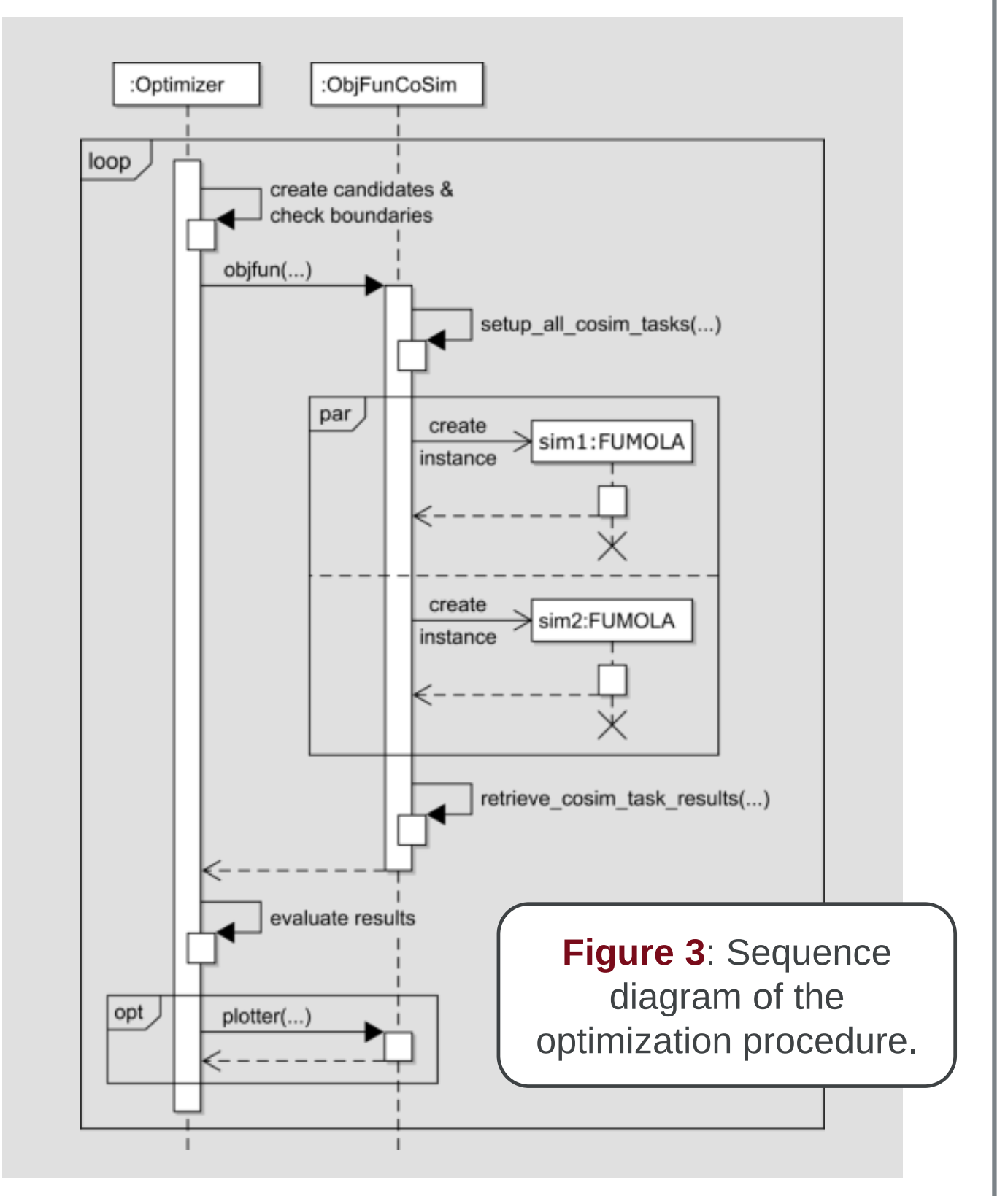

Figure 3: Sequence diagram of the optimization procedure.

Differential Evolution homepage (incl. link to MATLAB code): http://www1.icsi.berkeley.edu/~storn/code.html

6. EXAMPLE APPLICATION: OPTIMIZATION RESULTS

- **degrees of freedom** for optimization procedure:
  - **heat pump size** P_{hp}: electrical power consumption when turned on
  - **battery size** E_{bat}: electrical energy stored when fully charged
- **optimization function**: $\frac{E_{boiler}}{\varepsilon_{bat}} \longrightarrow \min$

  with E_{boiler}, the total thermal energy produced by the boiler, and f_{bat}, a measure for the battery's utilization

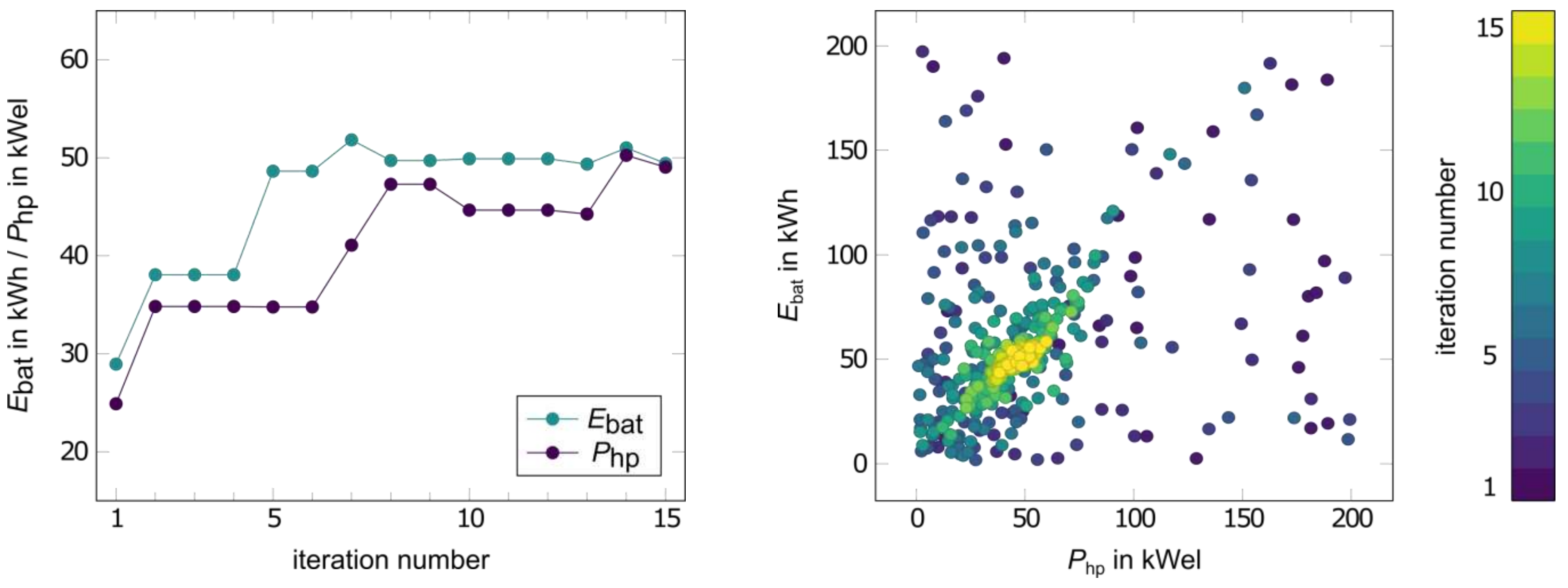

Figure 5: Example of optimization parameter evolution (left) and evolution of population distribution in the parameter plane (right).

REFERENCES

1. T. Blochwitz, et al.: *The Functional Mockup Interface for Tool independent Exchange of Simulation Models*. In: Proceedings of the 8th International Modelica Conference (2011)
2. J. Eker, et al.: *Taming heterogeneity – the Ptolemy approach*. Proceedings of the IEEE **91**(1), 127-144 (2003)
3. R. Storn and K. Price: *Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*. Journal of Global Optimization, **11**(4), 341-359 (1997)