The Journal of Open Source Software

FAME-Core: An open Framework for distributed Agent-based Modelling of Energy systems

Christoph Schimeczek [1¶], **Marc Deissenroth-Uhrig** [1,2], **Ulrich Frey** [1], **Benjamin Fuchs** [1], **A. Achraf El Ghazi** [1], **Manuel Wetzel** [1], and **Kristina Nienhaus** [1]

1 German Aerospace Center (DLR), Institute of Networked Energy Systems, Curiestr. 4, 70563 Stuttgart, Germany **2** School of Engineering, University of Applied Sciences Saarbrücken, Goebenstr. 40, 66117 Saarbrücken, Germany ¶ Corresponding author

DOI: 10.21105/joss.05087

Software

- Review
- Repository
- Archive

Editor: Frauke Wiese

Reviewers:

- @xtruan
- @pgranato

Submitted: 23 September 2022
Published: 12 April 2023

Summary

In this paper we introduce FAME-Core, a Java library supporting development and execution of agent-based simulation models (ABM).
It is a main component of the open **F**ramework for distributed **A**gent-based **M**odels of **E**nergy systems. FAME-Core offers rapid development and fast simulation execution capabilities while reducing required programming skills and efforts for creating and running complex ABM.

Within the domain of energy systems, ABMs are employed to investigate the dynamics of these systems and their emerging variables, e.g., electricity prices. Designing such models requires in-depth domain knowledge, e.g., of relevant actors, their interactions, and decision-making processes. In addition, enhanced programming skills are necessary to ensure numeric integrity and fast execution of the models' implementations. However, if models require parallelisation to keep computation times within acceptable limits, highly specialised programming skills are required and program complexity rises tremendously. FAME-Core removes the need for these specialised programming skills by providing parallelisation capability out of the box. It makes the code base of ABMs less complex by managing common tasks like reading inputs, gathering outputs, and scheduling of agents. Thus, FAME-Core eases the burden of ABM development and maintenance.

FAME-Core is one of two main components of FAME, a software suite to support creators and users of ABM simulations. Hence, the FAME repository[1] is split into several components. Each component addresses one aspect within FAME's model development and usage workflow, see Figure 1. Of these, FAME-Core and FAME-Io, see (Nitsch et al., 2023), are the most important ones. While the first is a Java library supporting development and execute ABM simulations, the latter is a Python package[2] to configure complex simulations and manage files associated with FAME. Other FAME components, like FAME-Mpi, FAME-Protobuf, and FAME-Prepare are supporting tools used by either FAME-Io or FAME-Core. In addition, FAME-Gui – a graphical user interface to FAME-Io – is currently under development. Lastly, the demonstration model FAME-Demo introduces the most important features of FAME-Core and FAME-Io and their combined workflow.

[1] https://gitlab.com/fame-framework/
[2] https://pypi.org/project/fameio/

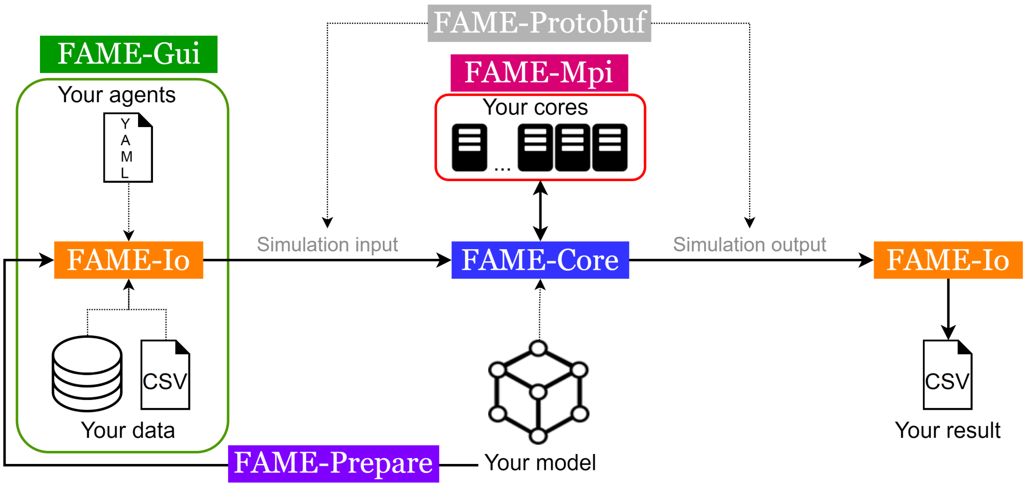

Figure 1: Components of FAME and their interactions.

Statement of need

In the field of energy systems analysis (ESA), optimisation-based models are by far the most common (Hoekstra et al., 2017). However, this type of modelling is not ideally suited to capture real-world market imperfections, as was pointed out by, e.g., (Trutnevyte, 2016). The social-planner approach employed in optimisation models often disregards, e.g., restricted foresight, complex (and maybe non-rational) actor behaviour, and market distortions due to policy instruments. Thus, results from such models may significantly deviate from outcomes of ABM simulations that consider such aspects, see (Torralba-Díaz et al., 2020).

Yet, ABMs are not new to the field of ESA. Many research questions were addressed with this modelling technique, see, e.g., (Ringler et al., 2016; Zhou et al., 2007). However, most models are not actively developed over a longer period. One possible cause might be the high effort of keeping models up to date to an ever-evolving energy system with diverse actors ranging from power plant owners over policymakers to household customers. A framework to design ABM in ESA that reduces coding overhead could help with both keeping existing models up-to date more easily and creating new models more quickly.
Such a framework, however, must be flexible enough to consider highly diverse actors that, e.g., decide on different time scales, ranging from seconds (e.g., grid-stabilisation) to decades (e.g., investment decisions) and must also consider that some decisions might be addressed with computationally cheap rule sets, whereas other decisions might require complex and demanding numerical algorithms (e.g., dynamic programming).

Many platforms, languages, and libraries already exist for the development of ABM, see the CoMSES Network[3]. We assessed 42 candidate ABM frameworks for the purpose of modelling energy systems. However, it was difficult to find frameworks that are both actively developed and open source, support diverse actors of inhomogeneous numerical complexity, and are able to execute code in parallel for high performance computing applications. An additional requirement was that the framework should use Java as programming language since it offers high performance. The most promising candidates regarding these criteria were:

- Akka[4]: a powerful and concurrent actor-model framework,
- DEVS-Suite[5]: a modelling framework for discrete event simulations and cellular automata,

[3] https://www.comses.net/
[4] https://akka.io/
[5] https://acims.asu.edu/devs-suite/

- Jade[6]: a wide-spread peer-to-peer platform for agent-based modelling,
- Jadex[7]: a framework to create applications using a service-oriented architecture,
- JIAC[8]: a multi-agent framework for distributed applications,
- MASS[9]: a parallel computing library for multi-agent simulations,
- OMS[10]: a simulation framework for connected components, and
- Repast[11]: a mature and well-known ABM suite.

A close inspection of these most promising candidates, however, revealed some drawbacks. None of the frameworks supports agent scheduling with discrete events (except for DEVS-suite). This scheduling approach might be significantly faster compared to a linear time progression if agents act on different time scales. DEVS-Suite, however, offered almost no documentation on its functionalities and its examples included but a selection of screenshots. We deemed Akka inappropriate for scientific modelling, as it does not guarantee the delivery of messages between agents. Repast offered parallelisation only with the C++ language, which is not wide-spread in the ESA community - Python support was added only recently. Jadex was licensed under GPL v3.0, and was thus not suitable for our research institute. MASS required spatial assignment of agents and organised parallelisation around their spatial distribution. For market models in ESA like (Schimeczek et al., 2023), this seemed unfitting as their agents are often unrelated to spatial coordinates. The linear data-flow driven design of OMS seemed not suited for depicting autonomous acting agents and their complex interactions. JIAC's syntax for defining agents is complex, the XML configuration files are bloated, and it relies on physical time intervals for agent scheduling. In conclusion, we deemed Jade the best candidate. However, it is designed around the FIPA[12] specifications which, after a close examination, seemed to not benefit the design of scientific models in ESA, but to add additional complexity to the model code.

We therefore developed FAME to feature discrete-event scheduling and guaranteed message delivery without the need of spatial assignment for agents. In addition, we aimed at supporting heavy-duty simulations with slim configuration files and simple model code under a non-copy left license. Furthermore, we wanted to provide easy setup procedures and extensive documentation. In our view, this combination of features could facilitate the development of new ABM in ESA and improve their longevity and interoperability.

In essence, FAME addresses management of data input, simulation output, scheduling of agent actions, and agent-to-agent communication. Java model code created with FAME-Core defines the structure of input and output files in YAML and CSV format and enable FAME-Io and FAME-Gui to easily configure such models.
FAME supports flexible scheduling of agents, i.e. the order of agent actions can be configured instead of hard-coded. Actions are either triggered via configured contracts that represent repeated actions, or alternatively, upon receiving of messages.

All FAME-Core models are ready for parallel execution by default and can be run either in single-core or multicore mode. Hence, model developers do not need to consider any aspects of parallelisation when using FAME. FAME's multicore mode is based on the Message Passing Interface[13] and automatically coordinates agent scheduling and message exchange across computing nodes. It enables fast execution of computationally intense simulation models featuring inhomogeneous agents with regard to time and complexity. For small to medium-sized simulations, however, single-core execution is recommended to avoid parallelisation overhead.

[6] https://jade.tilab.com
[7] https://github.com/actoron/jadex
[8] https://www.jiac.de/
[9] http://depts.washington.edu/dslab/MASS
[10] https://alm.engr.colostate.edu/cb/project/oms
[11] https://repast.github.io
[12] http://www.fipa.org/
[13] https://www.mpi-forum.org/

Use Case

FAME was designed to support modelling of energy systems. However, it is not restricted to this domain and may be employed for ABM simulations of other domains with similarly structured problems. Albeit FAME-Core is intended to drive numerically complex simulations, its rapid-prototyping features also support designing simpler or less demanding models.

For a first experience with FAME-Core, we recommend to inspect the code of the FAME-Demo[14] project. It may serve as a convenient starting point for any new model developments.

A full-fledged example of a FAME-Core based simulation is AMIRIS (Deissenroth et al., 2017; Schimeczek et al., 2023). It was ported to FAME in 2020. This did not only reduce the required lines of code in AMIRIS, but also opened new configuration options to the model, which weren't possible before. FAME-Core proved to be numerically efficient introducing only minor overhead in comparison to its previous implementation. In single-core mode, FAME-Core executes the AMIRIS model with low overhead – performing a simulation of the German wholesale electricity market for one year in hourly resolution on a desktop computer within about 20 seconds. In multicore mode, FAME demonstrated high parallelisation efficiency for a setup of 16 computationally heavy agents: Computation wall time was roughly proportional to $1/n$ as $n \in 1, 2, 4, 8$ cores were utilised.

Acknowledgements

This project was funded by the German Aerospace Center (DLR).

References

Deissenroth, M., Klein, M., Nienhaus, K., & Reeg, M. (2017). Assessing the Plurality of Actors and Policy Interactions: Agent-Based Modelling of Renewable Energy Market Integration. *Complexity*, *2017*, 1–24. https://doi.org/10.1155/2017/7494313

Hoekstra, A., Steinbuch, M., & Verbong, G. (2017). Creating agent-based energy transition management models that can uncover profitable pathways to climate change mitigation. *Complexity*, *2017*, 1–23. https://doi.org/10.1155/2017/1967645

Nitsch, F., Schimeczek, C., Frey, U., & Fuchs, B. (2023). FAME-Io: Configuration tools for complex agent-based simulations. *Journal of Open Source Software*, *under review*. https://doi.org/10.21105/joss.04958

Ringler, P., Keles, D., & Fichtner, W. (2016). Agent-based modelling and simulation of smart electricity grids and markets – A literature review. *Renewable and Sustainable Energy Reviews*, *57*, 205–215. https://doi.org/10.1016/j.rser.2015.12.169

Schimeczek, C., Nienhaus, K., Frey, U., Sperber, E., Sarfarazi, S., Nitsch, F., Kochems, J., & El Ghazi, A. A. (2023). AMIRIS: Agent-based market model for the investigation of renewable and integrated energy systems. *Journal of Open Source Software*. https://doi.org/10.21105/joss.05041

Torralba-Díaz, L., Schimeczek, C., Reeg, M., Savvidis, G., Deissenroth-Uhrig, M., Guthoff, F., Fleischer, B., & Hufendiek, K. (2020). Identification of the Efficiency Gap by Coupling a Fundamental Electricity Market Model and an Agent-Based Simulation Model. *Energies*, *13*(15), 3920. https://doi.org/10.3390/en13153920

Trutnevyte, E. (2016). Does cost optimization approximate the real-world energy transition? *Energy*, *106*, 182–193. https://doi.org/10.1016/j.energy.2016.03.038

[14] https://gitlab.com/fame-framework/fame-demo

Zhou, Z., Chan, W. K., & Chow, J. H. (2007). Agent-based simulation of electricity markets: A survey of tools. *Artificial Intelligence Review*, *28*(4), 305–342. https://doi.org/10.1007/s10462-009-9105-x