A Quantitative Metric to Evaluate the Satisfaction of Stylized Facts

Jan Niklas Martin
Institute for Visual and Analytic Computing, University of Rostock, Germany
Pia Wilsdorf
Institute for Visual and Analytic Computing, University of Rostock, Germany
Adelinde M. Uhrmacher
Institute for Visual and Analytic Computing, University of Rostock, Germany

Abstract

Stylized facts offer a way for validating and calibrating simulation models in data-scarce applications. SF-DSL formalizes such facts for automated model checking against simulation output. This paper extends SF-DSL with quantitative semantics to compute robustness and penalty scores, enabling optimization-based calibration. A case study using an epidemiological model demonstrates how these metrics support the identification of parameter configurations that best satisfy behavioral requirements.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics; General and reference → Metrics; General and reference → Reliability; General and reference → Validation

Keywords and phrases quantitative metrics, stylized facts, LTL, model calibration

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Supplementary Material Code: https://git.informatik.uni-rostock.de/mosi/dsl-for-stylized-facts/-/tree/Calibration?ref_type=heads/

Funding This work was funded by the German Research Foundation (DFG) grant no. 534708586.

1 Introduction

Over the last decades, modeling and simulation have become indispensable tools for analyzing, understanding, and predicting the behavior of complex systems. Successfully building a simulation model suited for these tasks typically involves successive steps of model refinement, validation, and calibration. Both validation and calibration heavily rely on data to ensure that the simulation results reflect real-world behavior [9, 19]. However, in many application fields, there is a lack of real-world data, e.g., due to the absence of historical data, high data collection costs, or privacy concerns [10]. But without proper validation and calibration, models risk producing misleading results, which can lead to incorrect conclusions [20].

Thus, in applications such as economics [7, 16], demography [21], or epidemiology [11], a common approach has become the use of so-called stylized facts [14]. Stylized facts summarize empirical findings and established domain-specific knowledge, often in natural language. E.g., a recently established fact about the dynamics of COVID-19 is that a “higher share of elderly in the county is correlated with higher death rates” [11]. In [22], a domain-specific language (DSL) was developed that enables the formalization of stylized facts (here referred to as SF-DSL). Its syntax stays close to natural language expressions but enriches them with clearly defined mathematical operators.

Based on the formalization and a custom model checker, stylized facts can be automatically evaluated on simulation model outputs [22]. Typically, a set of stylized facts is defined for a simulation model, representing its behavioral requirements. If all (or at least an important subset) of these facts are fulfilled, the model can be considered as validated.

In contrast, when calibrating simulation models, selected parameters or structures of the simulation model are adjusted. The goal is to identify the parameter or structural configuration that ensures that the simulation outputs fulfill the requirements of the model in the best possible way. In order to enable effective, automated calibration with stylized facts, a quantitative measure is required that can be optimized accordingly.

This paper therefore aims to enhance SF-DSL with a quantitative semantics that enables the model checker to not only produce a Boolean decision but a measure of robustness that expresses the degree of fulfillment as well as a penalty value that can steer a calibration algorithm towards the optimal (robust, ideally non-penalized) model parameters or structure. A case study with a classic epidemiological model illustrates the usage of SF-DSL to express model requirements and the calculation of the quantitative metrics for different assignments of the rate parameters.

2 SF-DSL's Formalization of Stylized Facts

SF-DSL is a domain-specific language that enables the formalization of stylized facts [22, 13]. It allows describing properties of time series, the relation between multiple time series, as well as temporal trends within and across time series. SF-DSL contains common operators known from temporal logic, which have been extended to support relative time intervals (see Listing 1) which allow for more specific temporal expressions, not to be confused with time intervals as in metric interval temporal logic (MITL) [12]. The language also defines unary and binary arithmetic operators on time series that return another time series, e.g., the element-wise *+*, *-*, ***, */*, as well as the *squared* operator. In addition, univariate and bivariate functions are included that result in a scalar value, such as *mean*, *standard deviation*, *autocorrelation*, and *cross-correlation*. Moreover, predicates are defined, such as *is heavy-tailed* and *is close to zero*, returning a Boolean value. These operators, functions, and predicates can be used to specify desirable properties of the simulation output. The following expression combines some of the previously described operators to formalize the condition for being a post-transitional country, thereby relating the time series of life expectancy, crude birth rate (CBR), and fertility (See Listing 1) as described in [4]. In this fact, the condition e_2 ("fertility is falling...") is expected to become true within the relative time interval of 5 to 20 time units after condition e_1:

Listing 1 Exemplary stylized fact, which was formalized with SF-DSL

```
(life_expectancy > 50 and CBR < 0.03) after (fertility is falling for 5 years or
CBR is falling for 5 years) within 5 to 20 years;
```

SF-DSL ships with a model checker, which allows for the automatic evaluation of stylized facts for given simulation output data. Previously, SF-DSL was used during model development to validate model versions against specified model requirements [22]. The applicability of SF-DSL in model development would be further enhanced if the stylized facts could also be used for model calibration, as discussed in the following section.

3 Expansion of SF-DSL with a Quantitative Interpretation

To be able to use the model checker for model calibration, a distance measure must be calculated to indicate how closely an expression is satisfied. Accordingly, all operators contained in SF-DSL must be evaluated not only qualitatively but also quantitatively.

3.1 Related Work

The notion of robustness as a definition of distance between a trajectory of a system and a temporal-logic behavioral property in signal temporal logic (STL) has been defined in the context of system verification [5], and allows for distinguishing between marginal and more robust satisfaction/violation of a property. The robustness degree can be measured as the distance between the system behavior and the boundary of the set of all behaviors that satisfy the property. It relies on ratio scale variables. This robustness is more positive as the behavior lies deeper inside the satisfying set, and more negative the further it is outside of it. The point-wise robustness can be calculated on a time series, e.g., for the inequality $x > 5$, by evaluating the difference $x - 5$, resulting in a robustness value of -3 for $x = 2$ at a given time t. When a temporal operator and interval are specified for which a formula is expected to hold, point-wise robustness cannot capture how long in this interval the property was satisfied or violated. Thus, left and right time robustness can be calculated to capture how far back/forward in time we can shift from a time t in the interval so that the truth value remains the same [5]. The overall robustness value could then be the longest duration for which the formula was satisfied/violated in the interval. Alternative formulations of time robustness do not use the longest duration but the maximum (or minimum, depending on the logic operator used) point-wise robustness that occurred in that interval [15].

In [2] the robustness definition was extended for stochastic models. For a given formula, they calculate a distribution of robustness degrees over all possible system trajectories. To receive a single robustness score, the expected value can be taken. In [3] this approach was applied and implemented for smoothed model checking and parameter identification with MITL formulas. In the following, we refer to their implementation, when mentioning robustness calculation for MITL.

3.2 Distance Measures for SF-DSL Operators

We adopt the robustness definitions by [2, 3], and slightly adapt them to the needs of SF-DSL where necessary. In addition, we design new distance metrics for the operators not yet covered. In the following, $\mathbf{x}$ describes a multivariate time series, x_k is the k^{th} time series from $\mathbf{x}$ and Index() return all indices of a given time series or list.

As described in Section 3.1, the robustness ρ of a formula is based on distances. For expressions such as $v > c$ the robustness is $\rho(v < c) = c - v$. In the case of the expression v *is positive* c corresponds to 0. If v < c is to apply, the subtrahend and minuend are swapped. If a value range is specified, as in v *is between* c_1 *and* c_2, robustness is calculated using $\rho(c_1 < v < c_2) = min(v - c_1, c_2 - v)$. Here c, c_1 and c_2 are constant thresholds and v is a scalar value. Similarly, distances can be calculated when evaluating functions on the time series, such as $mean(x_k) < c$ or the cross-correlation $CC(x_k,x_l) < c$. There, the function value is used in the difference calculation instead of v. Other expressions for describing a time series, specific to SF-DSL, use the same definition of distance. For example, if a time series is described as *exponentially distributed*, a p-value is calculated using a goodness-of-fit algorithm [6]. The robustness is determined by the distance between the p-value and a in SF-DSL pre-defined significance level. Other examples are the predicates *is growing* or *is falling*. There, the distance is calculated from the calculated slope of the time series and the *growing*/*falling* threshold defined in SF-DSL.

Given the robustness values for "simple expressions", we can adopt the quantitative semantics for the temporal logic operators *and*, *or*, *not*, *Always* ($\Box$) and *Finally* ($\Diamond$) as defined in the implementation of the robustness calculation in [3]. The robustness of the *and*/*or* operators is the minimum/maximum robustness of the two connected expressions. The *not* operator changes the sign of the robustness of the negated expression. The robustness of an expression *Always* φ (where φ may be one of the expressions described above) is the minimum point-wise robustness over the entire time series. Reversely, the robustness of *Finally* φ can be calculated by taking the maximum point-wise robustness.

$$\rho(\varphi_1 \wedge \varphi_2, \mathbf{x}, t) = \min(\rho(\varphi_1, \mathbf{x}, t), \rho(\varphi_2, \mathbf{x}, t)) \qquad \rho(\Box\varphi, \mathbf{x}, t) = \min_{i \in \text{Index}(\boldsymbol{x})}(\rho(\varphi, \mathbf{x}, i))$$

$$\rho(\varphi_1 \vee \varphi_2, \mathbf{x}, t) = \max(\rho(\varphi_1, \mathbf{x}, t), \rho(\varphi_2, \mathbf{x}, t)) \qquad \rho(\Diamond\varphi, \mathbf{x}, t) = \max_{i \in \text{Index}(\boldsymbol{x})}(\rho(\varphi, \mathbf{x}, i))$$

$$\rho(\neg\varphi, \mathbf{x}, t) = -\rho(\varphi, \mathbf{x}, t)$$

The evaluations of expressions using the quantifiers *Forall* ($\forall$)/*Exists* ($\exists$) *v in* $\{v_1,\dots,v_n\}$ $\varphi(v)$ are similar to the robustness calculations of *Always*/*Finally*.

$$\rho(\forall v \textit{ in } \{v_1,\dots,v_n\}\ \varphi(v), \mathbf{x}, t) = \min_{v \in \{v_1,\dots,v_n\}}(\rho(\varphi(v), \mathbf{x}, t))$$

$$\rho(\exists v \textit{ in } \{v_1,\dots,v_n\}\ \varphi(v), \mathbf{x}, t) = \max_{v \in \{v_1,\dots,v_n\}}(\rho(\varphi(v), \mathbf{x}, t))$$

In SF-DSL, temporal logic operators can optionally be supplemented with a relative time interval $[t', t'']$ which, depending on the operator, can mean a tightening or relaxation of the requirement. In the case of the expression φ_1 *precedes* φ_2 *within 1 to 5 time steps*, the time interval describes the time window in which φ_1 had to be fulfilled before the first occurrence of φ_2 (at index i^*). In this case, φ_1 would therefore have to be fulfilled at least once within the 5 time steps before the first occurrence of φ_2. Here, robustness is defined as:

$$\rho(\varphi_1 \textit{ precedes } \varphi_2 \textit{ within } t' \textit{ to } t'', \mathbf{x}, t) = \max(\min_{\substack{i \in \text{Index}(\boldsymbol{x}) \\ i \leq i^*}}(\rho(\varphi_2, \mathbf{x}, i), \quad \max_{j \in [i-t'', i-t']}(\rho(\varphi_1, \mathbf{x}, j))))$$

Operators of SF-DSL, whose quantitative semantics were not yet considered in the definitions described in 3.1, are the ranked quantifiers. They allow relating simulation results depending on parameter values of operators, for example, the smaller/larger a parameter's value v_i *in* $v = \{v_1,\dots,v_n\}$ is, the smaller/larger the result f_i of applying operator f to v_i and the simulation output is. For example, v_i could be a lag and f the calculation of an autocorrelation where the lag is applied. The calculation of the robustness values in all cases is shown in Table 1.

Table 1 Formulas for calculating the robustness of the ranked quantifiers.

	... the larger f_i	... the smaller f_i
the larger v_i ...	$\min_{i,j \in \text{Index}(v); j=i+1}(f_j - f_i)$	$-\max_{i,j \in \text{Index}(v); j=i+1}(f_j - f_i)$
the smaller v_i ...	$\min_{i,j \in \text{Index}(v); j=i+1}(f_i - f_j)$	$-\max_{i,j \in \text{Index}(v); j=i+1}(f_i - f_j)$

The calculation of robustness values if a time series x_k *is monotonically increasing/decreasing* is handled similarly. Here, instead of the distances between the calculated values for f, as with the ranked quantifiers (f_j - f_i or f_i - f_j), the differences between the values of the described time series itself are used instead. In addition, if a time series is described as strictly monotonically increasing/decreasing, the number of repeated values in the time series is counted and set in relation to the length of the time series.

Since our goal is to calibrate models with respect to stylized facts, we require a single metric that we can optimize. As for calibration, we are interested specifically in the cases where a property is violated, we define a second quantitative measure, i.e., the penalty $\psi = \min(\rho(\varphi, \mathbf{x}, t), 0)$. The penalty thus refers to the negative robustness values as introduced by [2]. Penalties have been used, e.g., by [15] to guide optimization algorithms, or in reinforcement learning to reduce the reward an agent receives for its action [8]. Calibration then aims to minimize the absolute value of the penalty, which has the advantage that an optimizer will focus on the expressions not yet fulfilled instead of further improving already fulfilled ones. As a downside, the result of the optimization will be a parametrization that is only weakly robust.

Contrary to the robustness definition of the *and* operator, we decided to calculate its penalty not as a minimum, but as the sum of the penalties of the two sub-expressions. This ensures that not only the lower-scoring sub-expression is taken into account in the calibration.

3.3 Normalization

When calibrating a simulation model based on stylized facts, all sub-expressions of the facts and their respective robustness values need to be evaluated and combined. Since the range between the observed variables over time can vary significantly, a normalization is needed to ensure a fair treatment of each sub-expression. However, this normalization procedure must not be based on the (minimum and maximum) values of the time series itself, as these are influenced by the varying model parameters during the calibration process – otherwise, the evaluation metric for the model parameters would depend on the model parameters it evaluates. In our current implementation, we assume that the ranges of observed variables are predefined so as to allow for normalization of distance values.

4 Case Study

To evaluate the robustness and penalty metrics, a small case study is carried out using an application to a simple simulation model. For this purpose, the susceptible-infected-recovered (SIR) model, implemented in the agent-based modeling language ML3 [18], is examined with four different parameter configurations and for two stylized facts.

4.1 SIR Model & Experiment Specification

The agents in this implementation of the SIR model [17] belong to one of the three basic types: susceptible, infected, and recovered. The model has two parameters: r_i describes the rate at which susceptible persons become infected when in contact with an infected neighbor in their network, r_r is the recovery rate. There is a transition rule from susceptible to infected (see Listing 2) and one from infected to recovered (see Listing 3).

Listing 2 ML3 transition rule from susceptible to infected

```
Person
| ego.status = "susceptible"
@ r_i * ego.network.filter(alter.status = "infected").size()
-> ego.status := "infected"
```

Listing 3 ML3 transition rule from infected to recovered

```
Person
| ego.status = "infected"
@ r_r
-> ego.status := "recovered"
```

The model forms a Population-based Continuous-time Markov Chain (PCTMC). There is a total constant population of 1000. Initially, 50 people are infected, and 950 are susceptible. The network is randomly generated using the Barabási-Albert algorithm [1]. Each person has at least five neighbors. Each execution of the simulation model is terminated after the 200th time step. One time step in the model corresponds to one day.

4.2 Stylized Facts

All parameter configurations are evaluated on the basis of two stylized facts. Since this is an abstract example, arbitrary stylized facts were chosen, which are influenced by the two model parameters (r_r & r_i). The first stylized fact (see Listing 4) states that the number of infected persons is expected to peak between 400 and 500 individuals.

Listing 4 Stylized fact 1: Peak of infected people

```
max(infected) is between 400 and 500;
```

The second stylized fact (see Listing 5) consists of two conjuncted sub-expressions. The first one states that the cross-correlation between the number of infected and recovered should be less than -0.85. The second sub-expression uses the temporal logic operator *precedes*. The second sub-expression states that after the last time step at which the number of infected has risen for at least 25 time steps in a row, the number of infected must start to fall within the next two time steps for at least 25 time steps.

Listing 5 Stylized fact 2: Cross-correlation between susceptible and infected and temporal relationship between increase and decrease in the number of infected

```
CC(susceptible, recovered) < -0.85 and (infected is growing for 25 days) precedes
(infected is falling for 25 days) within 1 to 2 days;
```

4.3 Results

For each parameter configuration, 100 simulation replications were carried out, and the average time series was produced (see Figure 1). The robustness and the penalty were calculated for the resulting average time series for both stylized facts (see Table 2). For the second stylized fact, these values were also calculated for the two sub-expressions to illustrate how the two metrics differ. Since all time series are bounded to the user-defined interval between 0 and 1000 (the total population), the distances could be normalized accordingly.

Table 2 Robustness and penalty evaluation for tested parameter configurations and the stylized facts. The sub-expressions of Stylized fact 2 are also evaluated in isolation to show the differences in calculation between robustness and penalty. The cross-correlation comparison is referenced as e_1, the *precedes* expression is referenced as e_2, and the conjunction of both is referenced as $e_1 \wedge e_2$.

	Stylized fact 1		Stylized fact 2					
	ρ	ψ	ρ			ψ		
			e_1	e_2	$e_1 \wedge e_2$	e_1	e_2	$e_1 \wedge e_2$
r_i=0.05; r_r=0.05	-0.114	-0.114	-0.043	0.707	-0.043	-0.043	0	-0.043
r_i=0.05; r_r=0.03	-0.222	-0.222	-0.111	0.818	-0.111	-0.111	0	-0.111
r_i=0.03; r_r=0.05	0.014	0	0.012	0.533	0.012	0	0	0
r_i=0.01; r_r=0.01	-0.114	-0.114	-0.010	-0.019	-0.019	-0.010	-0.019	-0.029

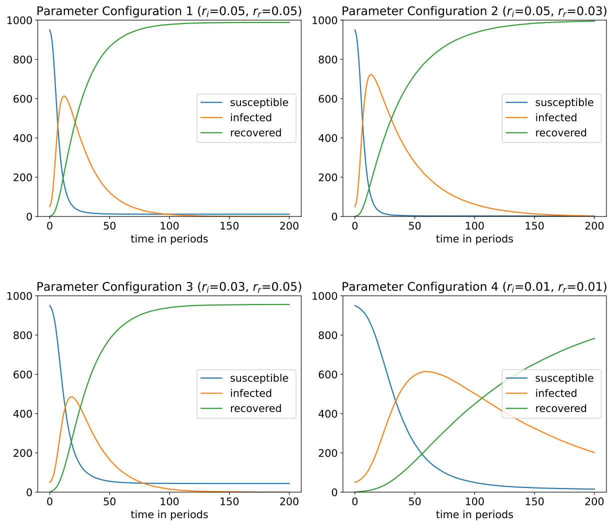

Figure 1 Results of Simulation runs using the respective parameter configuration. r_i is the rate of infection and r_r is the rate or recovery.

As can be seen in the result data, the penalty corresponds in most cases to the minimum of zero and the robustness. However, there are also scenarios in which this does not apply, for example, if both sub-expressions of the *and* operator are not satisfied. This can be seen in the evaluation of Stylized Fact 2 with the parameter configuration $r_i = 0.01; r_r = 0.01$. Here, both penalties of the sub-expressions are added together instead of simply taking the minimum of the two values, as is the case when calculating robustness. The e_2 sub-expression is not evaluated as fulfilled in this parameter configuration because the slopes before and after the peak of the *infected* curve are not steep enough to be classified accordingly. Therefore, the condition is not fulfilled in the given time interval of one to two days. This is a direct consequence of the low parameter values, which lead to a slower curve development overall.

Since parameter configuration 3 has a penalty of 0 for both stylized facts, it may be identified as an optimum in the calibration process.

5 Conclusion

We introduced a quantification of our language, SF-DSL, a language developed to support the specification of stylized facts. The quantification allows us to use the language not only for validation, as we did for an economic simulation study [16, 22], but also for calibration purposes. As a proof of concept, we applied the language and its quantitative semantics to evaluate parameter configurations of a small epidemiological model. The quantification of SF-DSL presents, to our knowledge, the first quantification of LTL operators with relative time intervals. Whereas for many operators we build on earlier work on quantifying logics, such as *Forall* or *not*, adaptations and extensions were required for some of the operators, such as *Precedes* or *and*. To assess the possible impact on simulation studies, we plan to apply the approach within a realistic epidemiological simulation study. This application will give us the possibility not only to test our design decisions when creating the language, but in particular also to analyse the accessibility of the quantitative semantics and thus overall usability with modelers and epidemiologists alike.

References

1 Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002. doi:10.1103/RevModPhys.74.47.

2 Ezio Bartocci, Luca Bortolussi, Laura Nenzi, and Guido Sanguinetti. On the robustness of temporal properties for stochastic models. In *International Workshop on Hybrid Systems Biology (HSB)*, pages 3–19, 2013. doi:10.4204/EPTCS.125.

3 Luca Bortolussi, Dimitrios Milios, and Guido Sanguinetti. U-check: Model checking and parameter synthesis under uncertainty. In Javier Campos and Boudewijn R. Haverkort, editors, *Quantitative Evaluation of Systems*, pages 89–104, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-22264-6_6.

4 Matteo Cervellati and Uwe Sunde. Life expectancy and economic growth: the role of the demographic transition. *Journal of economic growth*, 16:99–133, 2011. doi:10.1007/s10887-011-9065-2.

5 Alexandre Donzé and Oded Maler. Robust Satisfaction of Temporal Logic over Real-Valued Signals. In Krishnendu Chatterjee and Thomas A. Henzinger, editors, *Formal Modeling and Analysis of Timed Systems*, Lecture Notes in Computer Science, pages 92–106, Berlin, Heidelberg, 2010. Springer. doi:10.1007/978-3-642-15297-9_9.

6 Diane L. Evans, John H. Drew, and Lawrence M. Leemis. The Distribution of the Kolmogorov–Smirnov, Cramer–von Mises, and Anderson–Darling Test Statistics for Exponential Populations with Estimated Parameters. In Andrew G. Glen and Lawrence M. Leemis, editors, *Computational Probability Applications*, pages 165–190. Springer International Publishing, Cham, 2017. doi:10.1007/978-3-319-43317-2_13.

7 Cars H. Hommes. Modeling the stylized facts in finance through simple nonlinear adaptive systems. *Proceedings of the National Academy of Sciences*, 99(suppl_3):7221–7228, May 2002. doi:10.1073/pnas.082080399.

8 Parv Kapoor, Anand Balakrishnan, and Jyotirmoy V. Deshmukh. Model-based reinforcement learning from signal temporal logic specifications, 2020. URL: https://arxiv.org/abs/2011.04950, arXiv:2011.04950.

9 Jack P.C. Kleijnen. Verification and validation of simulation models. *European Journal of Operational Research*, 82(1):145–162, 1995. URL: https://www.sciencedirect.com/science/article/pii/0377221794000166, doi:10.1016/0377-2217(94)00016-6.

10 Jack PC Kleijnen. Validation of simulation, with and without real data. Discussion Paper 1998-22, Tilburg University, Center for Economic Research, 1998. URL: https://EconPapers.repec.org/RePEc:tiu:tiucen:40e44c6d-3f65-4f8f-8e30-cf9761b60eb8.

11 Christopher R Knittel and Bora Ozaltun. What does and does not correlate with covid-19 death rates. Working Paper 27391, National Bureau of Economic Research, June 2020. URL: http://www.nber.org/papers/w27391, doi:10.3386/w27391.

12 Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004. doi:10.1007/978-3-540-30206-3_12.

13 Jan Niklas Martin. Specifying and communicating the semantics of stylized facts. Master's thesis, University of Rostock, 2025. URL: http://eprints.mosi.informatik.uni-rostock.de/827/.

14 Matthias Meyer. How to Use and Derive Stylized Facts for Validating Simulation Models. In Claus Beisbart and Nicole J. Saam, editors, *Computer Simulation Validation: Fundamental Concepts, Methodological Frameworks, and Philosophical Perspectives*, pages 383–403. Springer International Publishing, Cham, 2019. doi:10.1007/978-3-319-70766-2_16.

15 Eshan D Mitra, Ryan Suderman, Joshua Colvin, Alexander Ionkov, Andrew Hu, Herbert M Sauro, Richard G Posner, and William S Hlavacek. Pybionetfit and the biological property specification language. *IScience*, 19:1012–1036, 2019. doi:10.1016/j.isci.2019.08.045.

16 Florian Peters, Doris Neuberger, Oliver Reinhardt, and Adelinde Uhrmacher. A basic macroeconomic agent-based model for analyzing monetary regime shifts. *PLOS ONE*, 17(12):1–39, 12 2022. doi:10.1371/journal.pone.0277615.

17 Oliver Reinhardt and Adelinde M. Uhrmacher. An efficient simulation algorithm for continuous-time agent-based linked lives models. In *Proceedings of the 50th Annual Simulation Symposium*, ANSS '17, San Diego, CA, USA, 2017. Society for Computer Simulation International.

18 Oliver Reinhardt, Tom Warnke, and Adelinde M. Uhrmacher. A language for agent-based discrete-event modeling and simulation of linked lives. *ACM Trans. Model. Comput. Simul.*, 32(1), January 2022. doi:10.1145/3486634.

19 Robert G. Sargent. Simulation model validation. In Tuncer I. Ören, Bernard P. Zeigler, and Maurice S. Elzas, editors, *Simulation and Model-Based Methodologies: An Integrative View*, pages 537–555. Springer Berlin Heidelberg, Berlin, Heidelberg, 1984. doi:10.1007/978-3-642-82144-8_19.

20 Robert G. Sargent. Verification and validation of simulation models: An advanced tutorial. In *2020 Winter Simulation Conference (WSC)*, pages 16–29, 2020. doi:10.1109/WSC48552.2020.9384052.

21 Jeffrey G. Williamson. Growth, distribution, and demography: Some lessons from history. *Explorations in Economic History*, 35(3):241–271, 1998. URL: https://www.sciencedirect.com/science/article/pii/S001449839890701X, doi:10.1006/exeh.1998.0701.

22 Pia Wilsdorf, Marian Zuska, Philipp Andelfinger, Adelinde M. Uhrmacher, and Florian Peters. Validation without data - formalizing stylized facts of time series. In *2023 Winter Simulation Conference (WSC)*, pages 2674–2685, 2023. doi:10.1109/WSC60868.2023.10408388.