Towards User-Centred Validation and Calibration of Agent-Based Models

Etienne Tack[1,2] [a], Gilles Enée[2] [b], Thomas Gaillard[3],
Jean-Marie Fotsing[2] and Frédéric Flouvat[4] [c]
[1]*INSIGHT, Noumea, New-Caledonia*
[2]*Institute of Exact and Applied Sciences (ISEA), University of New Caledonia (UNC), Noumea, New-Caledonia*
[3]*Ecosophy, Noumea, New-Caledonia*
[4]*Aix Marseille Univ, CNRS, LIS, Marseille, France*
e.tack@insight.nc, {gilles.enee,jean-marie.fotsing}@unc.nc,
thomas.gaillard@ecosophy.nc, frederic.flouvat@univ-amu.fr

Keywords: Agent-Based Modelling, Validation, Calibration, User-Centred.

Abstract: This paper describes a path to a user-centred approach for calibration and validation of agent-based models, particularly for spatially explicit models. Including the end-user in these critic modelling steps, we hope for better models that converge more easily toward reality. Using experts' knowledge, validation measures and feedback links to model parameters can be established. However, experts are not necessarily proficient in computer science. Tools should be created to help the transmission of their knowledge. With this paper, complying with a user-centred approach, we suggest using user-defined validation measures and a visual programming language to let the experts adjust themselves the behaviour rules of the agents.

1 INTRODUCTION

Multi-Agent Systems (MAS) are used to model dynamic systems and their environment. In contrast to a centralised artificial intelligence approach, MAS—also called agent-based models—are a decentralised "bottom-up" approach that allow modellers to solve problems by splitting knowledge and complexity into multiple entities called agents. An agent is an entity representing an object that evolves in an environment and can interact with it and other agents to perform any kind of tasks following behaviour rules (Ferber, 1999; Wooldridge and Jennings, 1995). Thus, the main objectives of MAS, among others, are to bring out a collective intelligence resulting of a sum of individual interactions, and to study the dynamics of complex systems which sometimes seem chaotic. Using such tools, modellers are able to do hypothesis and confirm them through simulation by analysing the results. The multi-agent model is not necessarily the end, sometimes it just helps to understand and find an easier way to model the studied subject.

MAS is not only a data driven approach, data can be used but is not mandatory. However, MAS is in practice a user-driven approach. The place of users in the agent based modelling is often very important. Definition of agents and interactions are mainly done by experts. One of the most difficult points here is the definition of agents' behaviour rules. The experts have a global knowledge of the different objects (agents or environment) that model the system. However, it is much more difficult for them to finely define their interactions, and the rules that govern them. Definition of these rules are done iteratively by experts based on results of many simulations, which is time-consuming and complex. The user requires being able to clearly identify missing or incorrect rules and, more generally, to evaluate whether a simulation result is correct or not. It is not trivial as it may depend on several factors as well as the users. Moreover, the objective of a simulation is generally not to have an exact projection of the behaviour of a system, but to have a realistic projection of it. Even if data exists, it is therefore not enough to compare it to simulation results. It is necessary to identify features which make it possible to check more generally the plausibility of the simulation in relation to data, and measure it. In such a modelling

[a] https://orcid.org/0000-0003-4131-1449
[b] https://orcid.org/0000-0002-0140-5291
[c] https://orcid.org/0000-0001-7288-0498

process, validation and calibration are thus closely linked to each other, and to the user.

For example, a typical application of MAS is urban growth simulation (Jokar Arsanjani et al., 2013; González-Méndez et al., 2021). The main objective is to model development of a city, through construction, modification and destruction of its buildings. Agents can be people, households or builders. The environment is spatially explicit and constructed from GIS data (Geographic Information System). It can include road networks, transportation networks, and amenities (e.g. schools, supermarkets, hospitals, police stations). Agents modify, construct or destruct buildings according to behaviour rules specified by a specialist in urban planning. Experts know some of these rules, but defining finely these rules, their parameters and triggering thresholds is not trivial. For example, building supermarkets may depend on population density in a district, but it is not easy to identify the corresponding threshold (and location). Moreover, experts may miss some of them, and they may not know some others, such as implicit rules linked to socio-cultural considerations. Identifying and tweaking these behaviours rules is difficult. Experts need methods and tools to help them identify, fine-tune and validate their rules iteratively.

In this paper, we first present and discuss different methodologies for MAS modelling. We show that these approaches offer a general framework, but their integration of validation and calibration remains limited. Next, we present a user-centred view of this validation and calibration process, and illustrate our proposition on a use case dealing with a spatial explicit model. Finally, we summarise the benefits of our propositions for the MAS community.

2 METHODOLOGY FOR MAS MODELLING

Agent-based models involve a lot of inputs (e.g. data, expert knowledge, ...) and processes (e.g. agents behaviour rules, environment definition, ...). A clear modelling approach is required for the reproducibility of agent-based experiments, especially for spatially explicit models aiming to reproduce human behaviours.

To facilitate the definition of agent based systems, the MAS research community has proposed several methodologies. The last methodology developed by the community is ODD (Overview / Design Concepts / Details) (Grimm et al., 2006, 2010). It is a descriptive framework which has been adopted as a protocol to describe and share agent-based models.

2.1 ODD: a Protocol to Standardise Model Definition

Before ODD, model descriptions were often hard to read and incomplete (Grimm et al., 2006). ODD offer a structure to help modellers not to forget anything that can be useful when reimplementing an agent-based model. Reimplementation is facilitated by a more transparent description of the model, therefore the results can be reproduced by peers more easily.

In the ODD acronym, the "O" stands for "Overview". This section covers general information about the model (purpose, state variables and scales, process overview and scheduling). As mentioned in (Grimm et al., 2006), "after reading the overview it should be possible to write, in an object-oriented programming language, the skeleton of a program that implements the described model".

The first "D" is for "Design Concepts". It is a description of the general concepts of the model. It covers agents' interactions: whether the agents should take into account the future states of the model in their reasoning, the emergence of collective dynamics resulting from local interactions, and the stochasticity of the model.

Finally, the last section—"Details"—describes the initialisation of the model, the required input data and the submodels (e.g. weather, expert models, ...).

Depending on the kind of model that needs to be described, ODD can still be insufficient, especially when human behaviour is involved. ODD+D (Müller et al., 2013) is an extension of ODD that focuses on human decision-making aspects. It has the same structure, but "adaptations were created to allow the protocol to be extended to human decision-making" (Crooks, 2018). Changes are located in the design concept category and implementation details (c.f. grey boxes in Figure 1). For further details, Crooks (2018) suggest seeing examples of the implementation of this protocol in Pires and Crooks (2017) and Orsi and Geneletti (6 11).

2.2 Validation and Calibration

The purpose of validation is to measure the distance between reality and simulations results. For simulation models, whose aiming to reproduce the reality, results needs to be validated. As shown in Figure 2, the model generates data through simulation. This data is compared to the observed data, and an error ("fitness" or distance) is calculated. If the model is not good enough, this information is used to adjust the model parameters (e.g. submodels' parameters or agents' rules), and run the same loop again. It is the

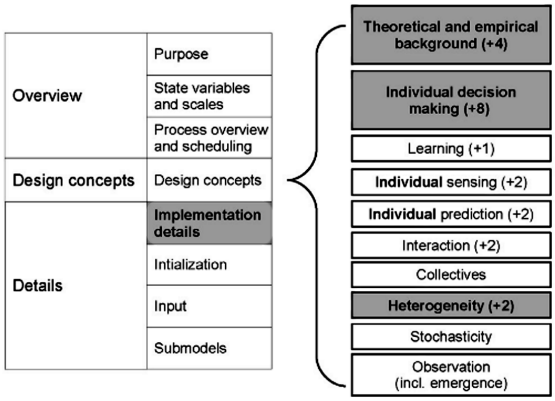

Figure 1: The structure of the ODD+D protocol. Grey boxes indicate new elements compared to the ODD protocol, and the number in brackets shows the added questions (Müller et al., 2013).

calibration step.

The first question that comes in such process (fig. 2) is: what do we rely on for calibration? Wilensky and Rand (2015) talk about "microvalidation" when the calibration is done on the agent attributes individually and "macrovalidation" when it is done by analysing the global dynamics of the system.

However, the model can be overfitted, i.e. it is too specialised to the observed data. Such as in classification approaches, the solution is to split the observation data into two datasets: one for calibration (training data) and the other for validation (validation data). Usually the partitions' distribution is between 75%/25% and 80%/20% (respectively for training/validation). If the observed dataset is small, Crooks (2018) suggest to use cross-validation, which train and test the model multiple times using random train and test partitions. However, it is not always possible in a user-centred approach because it is particularly time-consuming for the experts. The experts could help, correcting the model when the data is not covering local minima, but such interventions must remain highly targeted.

Calibration of a model based on natural processes or human behaviour can be tricky. It depends on lots of explicit and implicit parameters that are in interaction. These parameters may have lots of possible values, and even an infinite number of values for some. Therefore, the parameter search space can be very colossal.

Crooks (2018) introduces two kinds of calibration: "qualitative"and 'quantitative". On the one hand, with a "qualitative" calibration, the model is adjusted until it looks correct for the modeller (i.e. "face-validation"). Statistics and measures can be computed to help the modeller in this process (i.e. if the measurements are accurate. Then, the model is also suspected to be accurate).

On the other hand, with a "quantitative" calibration, the model is adjusted using measures that quantify the gap between simulation and reality. Railsback and Grimm (2019) define a methodology for quantitative calibration with the following steps:

1. Choose a few uncertain and important parameters.
2. Choose what kind of solution is wanted, a range of potential solutions, or an optimum value.
3. If time is important in the model, use a measure that integrates differences over time (e.g. mean difference, maximum error, RMSE, ...).
4. Choose calibration patterns based on which observation data exist.
5. Explore the model's parameter search space to find a satisfactory solution (detail below).
6. Analyse the calibration results, and identify the solution that best represent the reality.

Within this calibration methodology, the authors recommend several technical solutions. They suggest exploring the parameter search space, i.e. generate all parameter combinations until the difference with observation data is small enough (statistics, model error evaluation, ...). A sensitivity analysis can also be done for the more sensitive parameters (i.e. the parameters that influence the most in the model). Its function is to operate several simulations and to observe the outcomes of the simulations while changing the values of the parameters. It is usually applied with a "one factor at a time" approach. Sensitivity analysis is the most widely used method for testing the stability of a simulation (Crooks, 2018). An uncertainty analysis can also be useful as well. In such approach, the model is executed many times (e.g. 1000 runs) to construct a frequency distribution of simulation output measures. These distributions enable to capture variation in simulation results for each parameter. Next, they are compared to probability distributions (i.e. Gaussian, log-normal, ...) constructed using the means and standard deviations of the observed data. Finally, Railsback and Grimm (2019) also suggest using a heuristic search. This approach has the same objective as parameter space search, but instead of exploring all possibilities, it uses a heuristic to reduce the search space (s.t. genetic algorithms (Reuillon et al., 2013), gradient descent or gradient climbing, reinforcement learning, ...).

Hence, the whole process of validation and calibration is not trivial. Whatever is the method followed, the experts involved in the calibration process need simplified interactions and semi-automation to raise parameters leading to an invalid model.

Figure 2: The process of validation and calibration (Crooks, 2018)

2.3 Limitation of Current Methodologies

Although ODD and its extensions allow a clear structure, it is not yet universally used. For some complex models, the protocol is time-consuming or too restrictive to write (Crooks, 2018). In addition, it does not frame the details about calibration, validation and description of the agent behaviour rules, even though it requires a description for these. This limit leads to a more difficult calibration and validation, especially for complex and innovative models. The lack of calibration and validation approaches is still an open issue for agent-based models (Lee et al., 2015; Heppenstall et al., 2021).

Currently, calibration and validation solutions are chosen on a case-by-case basis, as these steps heavily require experts of the domain. Moreover, domain experts are not necessarily proficient to code the rules themselves, this can limit the explainability of the model, and thus, make the implementation of a multi-agent model more complicated. A more user-centred approach is needed.

3 USER-CENTRED VALIDATION AND CALIBRATION

In a multidisciplinary context, experts from many scientific domains (e.g. computer science, geography, sociology, ...) have to collaborate together to model a system. Thus, the models, rules and methods must be understandable by all so that the results are accepted, especially for non-computer scientists which are the users (i.e. field experts). It is the acceptability problem that is at the centre of many works lately.

In this context, we propose a user-centred view of calibration and validation, as these are the modelling phases that require the most knowledge from field experts. We believe that by involving more of the experts in these steps, the acceptability and the explainability of results will be improved.

To do this, we propose to integrate more the experts in the definition of the validation measures, and in the calibration of the behaviour rules, in relation with these measures. We illustrate this on an application related to urban growth, i.e. spatially explicit measures to evaluate multi-agents simulations in a GIS context. The agents are representing physically existing geographical objects (buildings) and have a shape that may evolve over time.

Measure	**Detail**
Surface area	Let a the area of a polygon
Shape index	$p/(2*\sqrt{a*\pi})$ with p the perimeter of a polygon
Morton index	$a/(\frac{\pi}{2}*\max_{d\in D} d^2)$ with D the set of all the possible diagonal lengths of a polygon
Perimeter to surface ratio	p/a
Number of objects	"Explicit"
Hausdorff Distance	$d_\infty(C,D) := \sup_{x\in\mathbb{R}^n} \lvert d_C(x) - d_D(x)\rvert$, with two object shapes $C,D \subset \mathbb{R}^n$ closed and non-empty (Rockafellar and Wets, 2009)
Kernel Density Difference	Difference of the kernel density of the generated data against the validation data (fig. 3)

Table 1: Example of spatial validation measures defined by an expert for urban growth simulation

3.1 User-defined Validation Measures

For each application, the experts identify a list of domain specific measures that will be used to validate the model. These measures are used to evaluate the distance between simulation and reality. Several validation measures may be identified, and combined, to provide an overview of the realism of the simulation. If field data is available, these measures can be used to process a distance between simulation results and real data.

In our case study related to urban growth simulation, these measures must describe the buildings generated by the model. They can be compared to real buildings observed in satellite images, for example. The table 1 lists several spatial measures identified by our expert (a geographer). The table 2 shows whether these measures can be applied at a micro-level (e.g. morphological measures applied for each individual objects) or at a macro-level (i.e. the whole area). Note that micro-level measures are aggregated to generate macro-level indicators (e.g. by averaging).

Most of these measures are processed for each object individually first. Only the Hausdorff distance and the Kernel density difference require considering several objects. The Hausdorff distance evaluates the

Measure	Level	
	Micro	Macro
Surface area	X	⋆
Shape index	X	⋆
Morton index	X	⋆
Perimeter to surface ratio	X	⋆
Number of objects		X
Hausdorff distance		X
Kernel density difference		X

⋆ Aggregations (i.e. mean value, standard deviation, minimum and maximum values, distribution by value, ...)

Table 2: Validation Measures Levels

distance between two building shapes. It can be used to measure the minimum distance between a building and its neighbours. It can also be aggregated to evaluate more globally the distances between buildings (using the mean, the average, etc.). The Kernel density is also processed at a macro-level, and enables to generate a density map (heat map). It evaluates the density of buildings in the studied area. All these indicators can be processed on simulated buildings and buildings extracted from images using remote sensing. If images are available, it is thus possible to measure the differences between the simulation and reality, and this for each simulation step for which data is available. This enables experts to study simulated buildings from different points of view. Generally, this analysis is done at a macro-level by experts.

These results must be presented to experts in user-friendly and customisable dashboards. For example, the Figure 3 illustrates a visualisation of the kernel density difference. It is obtained by subtracting between the kernel density map of simulated buildings and the one obtained from field data. It is important to let the experts customise their indicators and charts. The developed tool must therefore allow the expert to build dashboards intuitively. It is also important to provide experts a global view of all these indicators. For this, we use for example a "spider" chart for its simplicity (see Figure 4).

Beside GIS-centred measures, the experts must be able also to follow other non-spatial measures such as measures based on socio-economic variables, utility function values, behavioural rule triggering history, interactions history, ...

3.2 An Intuitive and Guided Visual Calibration

By giving the appropriate validation information to the expert, the goal is to help him make the model

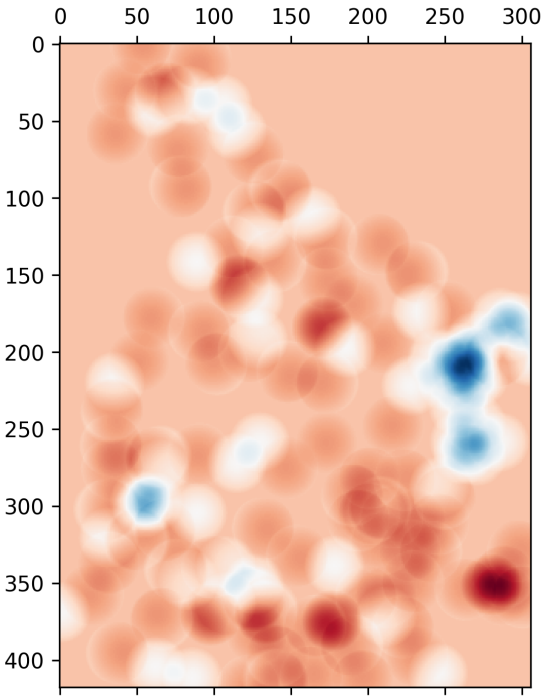

Figure 3: Kernel Density Difference (red areas represent a lack of buildings in the simulation and blue areas shows the contrary). Here is an example for our first case study, an informal settlement situated in Fiji.

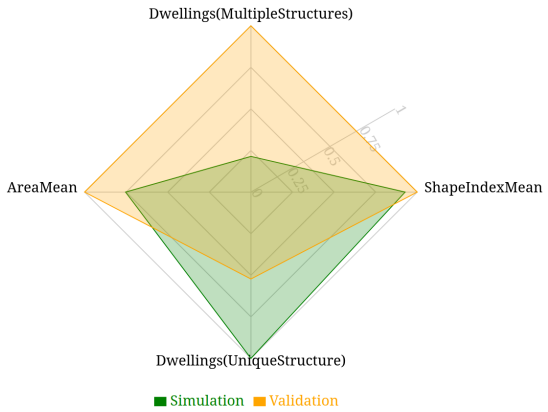

Figure 4: Comparative "Spider" Chart (values are normalised for better visualisation, which can be disabled).

converge towards reality. Based on the validation dashboards, the expert can explore the parameter space. He can modify certain parameters of the sub-models and adjust the agents' behaviours, while gradually checking that the results of the simulation converge.

As mentioned before, a difficult point in agent based modelling is the calibration of the behaviour rules of agents. Even if validation information is provided, it can be time-consuming to identify and adjust, at each execution, the rules that need to be modified.

To deal with this problem, we suggest identifying and store the links between validation measures and agents' rules, when defining the validation measures.

Experts define behaviour rules when modelling. They know their impact on the simulation (at least locally). For example, they know that some agent rules create buildings and that others extend buildings. Thus, they can store this information when defining validation measures. Once these links have been stored at design time, it is easy at runtime to display in the validation dashboards the rules impacted by specific measures. Thus, the experts can directly adjust the right rules when they identify in the dashboards an indicator that is too far from reality.

Going further in the implication of the user, it is important to provide an intuitive, and sufficiently rich, language to describe the behaviour rules of the agents. A visual language like scratch, which is designed to teach children coding, can be a solution for experts to be more autonomous in the calibration process. For example, such an approach has been followed in the RePast platform, through the Repast Simphony Statecharts (Ozik et al., 2015). However, it is not a full programming language. It is just a finite state machine that can set agent's statuses, and thus trigger behaviours written in others languages (e.g. ReLogo, Java, ...).

Two types of visual programming do exist. Instructions can either be represented with blocks (like Scratch) or with nodes in a graphical representation (like flowcharts in node programming languages). For example, Figure 5 illustrates a behaviour rule "look for food" described in the NetLogo language to simulate ant foraging (Wilensky, 1997). The same rule is translated in the Scratch language in Figure 6.

```
to look-for-food  ;; turtle procedure
  if food > 0
  [ set color organge + 1  ;; pick up food
    set food food - 1  ;; reduce food source
    rt 180 ;; and turn around
    stop ]
  ;; go in the direction where the chemical
  ;; smell is strongest
  if (chemical >= 0.05) and (chemical < 2)
  [ uphill-chemical ]
end
```

Figure 5: NetLogo code for the "look for food" procedure Wilensky (1997)

Of course, such manual modification of behaviour rules still time-consuming and difficult. To alleviate this, it is possible to use machine learning approaches. For example, genetic algorithms have been used at the model level to set parameter thresholds (Reuillon et al., 2013). Reinforcement learning have also been used a lot in MAS (Buşoniu et al., 2010; Zhang et al., 2019). They are used to finding good parameters for submodels, or to build a behaviour model for agents. However, a problem of such approach is the interpretability of the generated model. It is far from behaviour rules defined by experts (which are often "if ... then ... else" rules). Another problem for some applications is the quantity and the quality of available field data used to train algorithms. For example, methods based on deep learning need a lot of data (and are black boxes for experts). In any case, even if learning algorithms are used, the expert must be fully integrated into the calibration process to guarantee the acceptability of the results. He must be able to understand the generated models and to guide the learning algorithms thanks to his domain knowledge. This is an important challenge for future agent learning methods.

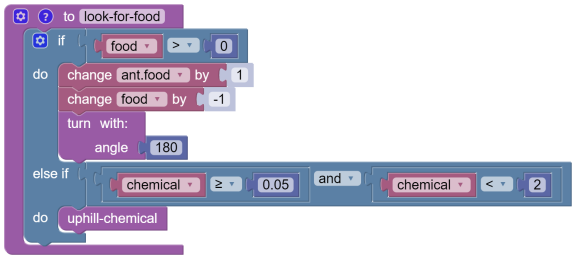

Figure 6: The "look for food" procedure written with a visual programming language (for instance, Blockly which has more or less the same functionalities as Scratch)

4 CONCLUSIONS

In this paper, we present a user-centred approach for calibration and validation of agent-based models. By including the end user in most of the modelling phases, especially calibration and validation, we are hoping for better explainability of simulation results and a smoother transmission of expert knowledge. We illustrate this position with an application dealing with urban growth. Such an application involves complex, spatially explicit models of human behaviour, and experts who have some knowledge of the underlying mechanisms, but who want interpretable models and acceptable results.

In order to achieve this goal, we think it is important to involve more the experts in the model design. They must be able to define behaviour rules themselves using a visual programming language. They must also be able to define their validation measures and to navigate easily between behaviour rules and validation results. Machine learning methods can help to automate and simplifying the calibration process, but chosen methods must produce interpretable models, such that experts can trust simulation results. We are currently implementing such an approach with a geographer in the setting of informal settlement growth modelling (e.g. slums), and the first results are very encouraging.

ACKNOWLEDGEMENTS

We specifically thank the University of South-Pacific (USP) for participating in data acquisition for an informal settlement in Fiji.

This work was financed by the Pacific Islands Universities Research Network (PIURN) and the French Ministry for Foreign Affairs (Pacific Fund). It is now supported by the French National Research Agency (ANR SITI project).

REFERENCES

Buşoniu, L., Babuška, R., and Schutter, B. D. (2010). Multi-agent reinforcement learning: An overview. In *Innovations in Multi-Agent Systems and Applications - 1*, pages 183–221. Springer Berlin Heidelberg.

Crooks, A. (2018). *Agent-Based Modelling and Geographical Information Systems: A Practical Primer*. Spatial Analysis and GIS. SAGE Publications.

Ferber, J. (1999). *Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman.

González-Méndez, M., Olaya, C., Fasolino, I., Grimaldi, M., and Obregón, N. (2021). Agent-Based Modeling for Urban Development Planning based on Human Needs. Conceptual Basis and Model Formulation. *Land Use Policy*, 101:105110.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jørgensen, C., Mooij, W. M., Müller, B., Pe'er, G., Piou, C., Railsback, S. F., Robbins, A. M., Robbins, M. M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R. A., Vabø, R., Visser, U., and DeAngelis, D. L. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 198(1-2):115–126.

Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., and Railsback, S. F. (2010). The ODD protocol: A review and first update. *Ecological Modelling*, 221(23):2760–2768.

Heppenstall, A., Crooks, A., Malleson, N., Manley, E., Ge, J., and Batty, M. (2021). Future Developments in Geographical Agent-Based Models: Challenges and Opportunities. *Geographical Analysis*, 53(1):76–91.

Jokar Arsanjani, J., Helbich, M., and de Noronha Vaz, E. (2013). Spatiotemporal simulation of urban growth patterns using agent-based modeling: The case of Tehran. *Cities*, 32:33–42.

Lee, J.-S., Filatova, T., Ligmann-Zielinska, A., Hassani-Mahmooei, B., Stonedahl, F., Lorscheid, I., Voinov, A., Polhill, G., Sun, Z., and Parker, D. C. (2015). The Complexities of Agent-Based Modeling Output Analysis. *Journal of Artificial Societies and Social Simulation*, 18(4):4.

Müller, B., Bohn, F., Dreßler, G., Groeneveld, J., Klassert, C., Martin, R., Schlüter, M., Schulze, J., Weise, H., and Schwarz, N. (2013). Describing human decisions in agent-based models – ODD + D, an extension of the ODD protocol. *Environmental Modelling & Software*, 48:37–48.

Orsi, F. and Geneletti, D. (2016-11). Transportation as a protected area management tool: An agent-based model to assess the effect of travel mode choices on hiker movements. *Computers, Environment and Urban Systems*, 60:12–22.

Ozik, J., Collier, N., Combs, T., Macal, C. M., and North, M. (2015). Repast simphony statecharts. *Journal of Artificial Societies and Social Simulation*, 18(3).

Pires, B. and Crooks, A. T. (2017). Modeling the emergence of riots: A geosimulation approach. *Computers, Environment and Urban Systems*, 61:66–80.

Railsback, S. F. and Grimm, V. (2019). *Agent-Based and Individual-Based Modeling A Practical Introduction, Second Edition*. Princeton University Press.

Reuillon, R., Leclaire, M., and Rey-Coyrehourcq, S. (2013). OpenMOLE, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems*, 29(8):1981–1990.

Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational Analysis*. Springer Science & Business Media.

Wilensky, U. (1997). NetLogo Ants model. http://ccl.northwestern.edu/netlogo/models/Ants. [Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL].

Wilensky, U. and Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. The MIT Press.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152.

Zhang, K., Yang, Z., and Başar, T. (2019). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384.