

1. What is the main claim of the paper? Why is this an important contribution to the autonomous agents and multi-agent systems literature?

We are convinced that code testing (verification) of agent-based simulation (ABS) is important, especially for models designed to support far-reaching policy decisions, where invalid results could have severe consequences for property and people. Still, it seems that research on code testing ABS is dramatically underrepresented as to the best of our knowledge only two papers (see point 3) deal with that matter so far. Part of the problem is that we think that unit testing, which is the established method of code testing in general, is not very well suited for that task due to ABS exploratory, generative and, most important, stochastic nature. Unit testing in general is simply not able to deal with this explorative process and does not support stochastic / random / probabilistic testing.

This paper claims that property-based testing could be a remedy for the lack of code testing in ABS as it maps naturally to testing ABS. The reason is that it allows to express model specifications and invariants directly in code and test them through automated and randomised test data generation. Further, the paper claims that in the context of ABS, property-based testing is strictly more powerful than unit testing as it allows to run thousands of test cases automatically instead of constructing each manually and because it is able to encode probabilities, something unit testing is not capable of in general.

Thus we claim that by incorporating property-based testing into simulation development, confidence in the correctness of an implementation is likely to increase dramatically, something of fundamental importance for ABS in general and for models supporting far-reaching policy decision in particular.

2. What is the evidence you provide to support your claim? Be precise.

To support our claims we are using an agent-based SIR model as use case and demonstrate how property-based testing works on a technical level to encode informal specifications into formal ones in code as property tests. Most importantly it is shown how probabilities can be expressed and tested using statistically robust verification. The result are only 2 property tests but which cover the specification of the part of an agent, and when are run, are the equivalent of thousands of unit tests. We claim that these demonstrations are strong evidence for our claims made in the point above.

3. What papers by other authors make the most closely related contributions, and how is your paper related to them? Provide journal references whenever possible. Don't rely only on a conference proceedings.

We are the authors of the conceptual paper "Thaler, J., Siebers, P.O.: Show Me Your Properties! The Potential Of Property-Based Testing In Agent-Based Simulation. SummerSimulation Conference 2019, Berlin.". This submission builds on the concepts developed in that conceptual paper but extends it substantially in technical terms and adds the dimension of statistical robust verification with property-based testing.

The paper "Collier, N., Ozik, J.: Test-driven agent-based simulation development. In: 2013 WinterSimulations Conference (WSC), pp. 1551–1559 (2013). DOI 10.1109/WSC.2013.6721538" is closely related to the topic and is the only other reference we could find on code testing in ABS on a technical level.

4. Have you published parts of your paper before, for instance in a conference? If so, give details of your previous paper(s) and a precise statement detailing how your paper provides a significant contribution beyond the previous paper(s).

We have published the conceptual paper mentioned in the previous point (Thaler et al). It discusses property-based testing in ABS from a rather conceptual and high-level perspective, using pseudo code without going much into technical detail. It did not mention the power of statistically robust verification available with property-based testing. This submission expands the concepts substantially, introduces the new use case of technical specification testing and focuses primarily on discussing and using property-based testing on a technical level. More importantly, it discusses how probability and hypothesis testing can be implemented with property-based testing, something not discussed in the conceptual paper.