

The Art of Iterating: Update-Strategies in Agent-Based Simulations

Jonathan THALER

January 19, 2017

Abstract

When developing a model for an Agent-Based Simulation (ABS) it is of very importance to select the right update-strategy for the agents to produce the desired results. In this paper we develop a systematic treatment of all general update-strategies in ABS and discuss their philosophical interpretation and meaning. Further we discuss the suitability of the very three different programming languages Java, Scala with Actors and Haskell to implement each of the update-strategies. Thus this papers contribution is the development of a general terminology of update-strategies and their implementation issues in various kinds of programming languages.

1 Introduction

In the paper of [3] the authors showed that the results of the simulation of the classic prisoners-dilemma on a 2D-grid reported in [4] depends on a a very specific strategy of iterating this simulation and show that the beautiful patterns as reported by [4] will not form when selecting a different iteration-strategy. Although the authors discussed philosophical aspects of choosing one strategy over the other, they lacked to generalize their observation. We will do so in the central message of our paper by stressing that when doing Agent-Based Simulation & Modelling (ABM/S) *it is of most importance to select the right iteration-strategy which reflects and supports the corresponding semantics of the model*. We find that this awareness is yet still under-represented in the literature of ABM/S and most important of all is lacking a systematic treatment. Thus our contribution in this paper is to provide such a systematic treatment by

- Presenting all the general iteration-strategies which are possible in an ABM/S.
- Developing a systematic terminology of talking about them.
- Giving the philosophical interpretation and meaning of each of them.

- Comparing the 3 programming languages Java, Haskell and Scala in regard of their suitability to implement each of these strategies.

Besides the systematic treatment of all the general iteration-strategies the paper presents another novelty which is its inclusion of the pure functional declarative language Haskell in the comparison. This language has so far been neglected by the ABM/S community which is dominated by object-oriented (OO) programming languages like Java thus the usage of Haskell presents a real, original novelty in this paper.

2 Problem

1 Page

- describe in more technical detail what the introduction tells.
- [3] and <https://www.openabm.org/book/33102/54-importance-sequence-updating> only mentions synchronous and asynchronous updates but this is not precise enough and lacks subcategories

3 Update-Strategies

2 Pages

This is all programming-language agnostic

- A terminology and classification of all the possible iteration-strategies presented as a list
- short discussion separate paragraph for each
 - Abstract implementation of the strategies
 - Philosophical meaning and interpretation
 - Advice for selecting it

4 Implementation Approaches

5 Pages

This is now very programming-language specific

- Mapping the strategies to 3 programming-languages: Java, Scala with Actors, Haskell
- Comparing the programming languages in regard of their suitability to implement each of these strategies
- Screen-shots of results of the same simulation-model with all the strategies

5 Related Research

- [3]
- [1]
- [2]

6 Conclusion

- Selecting the correct Iteration-Strategy is of most importance and must match the model semantics
- Java: best for non-parallel, non-concurrent strategy
- Scala with Actors: best for concurrency
- Surprise: Haskell can faithfully implement all strategies equally well, something not anticipated in the beginning

7 Further Research

- Reasoning in Haskell about the Model & Simulation
- Develop a small modelling-language which is close to the Haskell-Version of modelling agents therefore specification and implementation match

References

- [1] A, M. L., B, R. D., AND B, K. R. *A Framework for Megascale Agent Based Model Simulations on the GPU*. 2008.
- [2] BOTTA, N., MANDEL, A., AND IONESCU, C. Time in discrete agent-based models of socio-economic systems. Documents de travail du Centre d'Economie de la Sorbonne 10076, Université Panthéon-Sorbonne (Paris 1), Centre d'Economie de la Sorbonne, 2010.
- [3] HUBERMAN, B. A., AND GLANCE, N. S. Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences* 90, 16 (Aug. 1993), 7716–7718.
- [4] NOWAK, M. A., AND MAY, R. M. Evolutionary games and spatial chaos. *Nature* 359, 6398 (Oct. 1992), 826–829.