Research Proposal

Functional methods in agent-based modelling & simulation.

Jonathan Thaler

November 7, 2016

Institution: University of Nottingham
Department: School of Computer Science
Project: PhD Programme 2016-2019
Supervisor: Dr. Peer-Olaf Siebers
Co-Supersvisor: Dr. Thorsten Altenkirch

Abstract

TODO: Consider a provocative abstract and an interesting title

Agent-Based Modelling and Simulation (ABM/S) is still a young discipline and the dominant approach to it is using object-oriented methods. This thesis goes into the opposite direction and asks how ABM/S can be mapped to and implemented using functional methods and what one gains from doing so. To the best knowledge of the author, so far no proper treatment of ABM/S in this field exists but a few papers which only scratch the surface. The author argues that approaching ABM/S using functional methods offers a wealth of new powerful tools and methods. The most obvious one is that when using pure functional computation reasoning about the correctness and about total and partial correctness of the simulation becomes possible. Also pure functional approaches allow the design of an embedded domain specific language (EDSL) in which then the models can be formulated by domain-experts. The strongest point in using EDSL is that ideally the distinction between specification and implementation disappears: the model specification is then already the code of the simulation-program. This allows to rule out a serious class of errors where specification and implementation does not match, which is especially a big problem in scientific computing thus making functional methods in ABM/S especially suitable for scientific computing. The application will be in the field of agent-based computational economics (ACE) where the primary goal will be to compare functional and non-functional methods for developing ACE simulations and to identify in which scenarios pure functional methods shine and where their limits are.

TODO: Contributions should appear in the conclusion, introduction and abstract

1 Introduction

I noticed that it is pretty hard to convince an agent-based economics specialist who is not a computer scientist about a pure functional approach. My conjecture is that the implementation technique and method does not matter much to them because they have very little knowledge about programming and are almost always self-taught - they don't know about software-engineering, nothing about proper software-design and architecture, nothing about software-maintenance, nothing about unit-testing,... In the end they just "hack" the simulation in whatever language they are able to: C++, Visual Basic, Java or toolboxes like Netlogo. For them it is all about to *get things done somehow* and not to get things done the right way or in a beautiful way - the way and the method doesn't matter, its just a necessary evil which needs to be done. Thus if functional programming could make their lives easier, then they will definitely welcome it. But functional programming is, i think, harder to learn and harder to understand - so one needs to provide an abstraction through EDSL. So I REALLY need to come up with convincing arguments why to use pure functional approaches in ACE THEY can understand, otherwise I will be lost and not heard (not published,...).

What ACE economists care for:

- Very: Qualitative modelling with quantitative results
- Yes: Easy reproducibility
- Likely: Reasoning about convergence?
- Likely: EDSL

My contributions are: pure functional framework, functional agent-model for market-simulations, EDSL for market-simulations, qualitative / implicit modelling with quanitative results, reasoning in my framework about convergence

IDEA: could I develop non-causal modelling (models are expressed in terms of non-directed equations, modelled in signal-relations) to allow for qualitative modelling for the agent-based economists? See hybrid modelling paper of Yampa. **THIS WOULD BE A HUGE NOVEL CONTRIBUTION TO ACE ESPECIALLY WHEN COMBINED WITH AN EDSL AND PROVIDING FULL REFERENTIAL TRANSPARENCY TO KEEP THE ABILITY TO REASON ABOUT CONVERGENCE**. This should be covered in the "EDSL"-paper.
TODO: maybe i should really focus only on market models? otherwise too much?

central novelty of my PhD: model specification = runnable code. possible through EDSL. but only in specific subfield of ACE: market-models. need a functional description of the model, then translate it to model specification in

EDSL and then run it to see dynamics. But: model specification moves closer to functional programming languages.

another novelty approach: model specification through qualitative instead of quantiative approaches. is this possible?

WHY FUNCTIONAL? "because its the ultimate approach to scientific computing": fewer bugs due to mutable state (why? is thos shown obkectively by someone?), shorter (again as above, productivity), more expressive and closer to math, EDSL, EDSL=model=simulation, better parallelising due to referental transparency, reasoning

scientific results need to be reproduced, especially when they have high impact. a more formal approach of specifying the model and the simulation (model=simulation) could lead to easier sharing and easier reporduction without ambigouites

pure functional agent-model & theory, EDSL framework in Haskell for ACE

1. Which kind of problem do we have?
2. What aim is there? Solving the problem?
3. How the aim is achieved by enumerating VERY CLEAR objectives.
4. What the impact one expects (hypothesis) and what it is (after results).

Note: It is not in the interest of the researcher to develop new economic theories but to research the use of functional methods (programming and specification) in agent-based computational economics (ACE).
NOTE: Get the readers attention early in the introduction: motivation, significance, originality and novelty.

1.1 Methods

Methods need to be selected to implement the simulations. Special emphasis will be put on functional ones which will then be compared to established methods in the field of ABM/S and ACE.

1.2 Scenarios

To apply and test functional methods in ACE, four scenarios of ACE are selected and then the methods applied and compared with each other to see how each of them perform in comparison. The 4 selected scenarios represent a selection of the challenges posed in ACE: from very abstract ones to very operational ones.

1.3 Comparison

Each of the selected scenarios is then implemented using the selected methods where each solution is then compared against the following criteria:

1. suitability for scientific computation
2. robustness
3. error-sources
4. testability
5. stability
6. extendability
7. size of code
8. maintainability
9. time taken for development
10. verification & correctness
11. replications & parallelism
12. EDSL

This will then allow to compare the different methods against each other and to show under which circumstances functional methods shine and when they should not be used.

2 Literature Research

2.1 TODO: Yampa

read papers of yampa and games and summarize in paper for "pure functional agents"

2.2 TODO: ACE

read ACE introduction papers, summarize in this research-proposal
look into computable economics book: `http://www.e-elgar.com/shop/computable-economics`
TODO: the reading should pull out the essence of what types of ACE there are and what features each type has (continuous/discrete time, complex agent communication, equilibriua, networks amongst agents,...)
NOTE: I REALLY need to work out what is special in ACE? what is the unique property of ACE AS compared to other ABM/S? Conjecture: equilibrium of dynamics is the central aspect. `http://www2.econ.iastate.edu/tesfatsi/ace.htm`

Ballot et al. (2015) Agent-based modeling and economic theory: where do we stand? - Ballot, Mandel, Vignes
Richiardi (2007) Agent-based Computational Economics. A Short Introduction - Richiardi
Tesfatsion (2006) Agent-based computational economics: a constructive approach to economic theory - tesfatsion
Blume et al. (2015) Introduction to computer science and economic theory - blume, easley, kleinberg
Tesfatsion (2002) agent-based computational economics - tesfatsion

2.3 TODO: functional agent-models

look into functional agents in wooldridge 2.5/6 and find primary papers and read them

2.4 TODO: market-models

find all agent-based market models (gintis, gode and sunder,...), read gintis equilibrium stuff and summarize in the paper: EDSL for agent-based market-simulations

2.5 TODO: actor- & agent-models

read all of the actor-related stuff and summarize in the paper: actor model in ABM/S

3 Methods

3.1 Method 1: Haskell

This is the main functional method this thesis wants to investigate as it is the purest functional programming language of all the methods.

3.2 Method 2: Scala & Actors (Akka)

This method was selected because Scala is an object-oriented functional programming language and has a powerful library included which implements the actor-model. Because actors and agents are closely related this is an obvious method to follow.

3.3 Method 3: AnyLogic / NetLogo / Repast

These tools are state-of-the-art in ABM/S and ACE and are included to show how one can perform scenarios (see below) with these tools.

3.4 Method 4: Java

Java is the state-of-the-art programming language in ABM/S and ACE and is thus included as well as a benchmark against such a state-of-the-art.

4 Goals/TODOs 1st Year

4.1 Practical

- Implement SIRS in AnyLogic - DONE
- Implement Wildfire in Akka - DONE
- Implement Wildfire in Yampa.
- Implement Market-models in Yampa, Akka, Java and AnyLogic.

4.2 Reading

- Understanding Capitalism. Bowles et al. (2005)
- Microeconomic Theory. TODO cite
- Market Microstructure. TODO cite
- A computable universe. TODO cite
- The nature of computation. TODO cite
- Economics and Computation by Parkes and Seuken `http://economicsandcomputation.org/`

4.3 Studying

- Get into basics of economics and equilibrium theory. Why: because i need to understand basics to understand the models better and to talk and sell my models better to economists.
- Understand theory of out-of-equilibrium / non-walrasian models: TODO (various Gintis & Mandel Papers)
- Understand Market Micro-structure: Lehalle and Laruelle (2013), Baker et al. (2013) Part II: Chapter 8-12.

4.4 Research on Papers (see the papers pdf)

- The use of Actor-Model in ABM/S.
- Implementing pure functional agents.
- An EDSL for simulating qualitative agent-based market-models.

References

(2008). *Emergent Macroeconomics*. Springer Milan, Milano.

Aldridge, I. (2009). *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*. John Wiley and Sons. Google-Books-ID: fEXKZAAVu4oC.

Baker, H. K., Kiymaz, H., Alan, N. S., Bildik, R., and Schwartz, R. (2013). Market Microstructure in Emerging and Developed Markets. *Business Faculty Book Gallery*.

Ballot, G., Mandel, A., and Vignes, A. (2015). Agent-based modeling and economic theory: where do we stand? *Journal of Economic Interaction and Coordination*, 10(2):199–220.

Bauer, T., Erwig, M., Fern, A., and Pinto, J. (2011). Adaptation-Based Programming in Haskell. *Electronic Proceedings in Theoretical Computer Science*, 66:1–23. arXiv: 1109.0774.

Blume, L., Easley, D., Kleinberg, J., Kleinberg, R., and Tardos, . (2015). Introduction to computer science and economic theory. *Journal of Economic Theory*, 156(C):1–13.

Botta, N., Mandel, A., Ionescu, C., Hofmann, M., Lincke, D., Schupp, S., and Jaeger, C. (2011). A functional framework for agent-based models of exchange. *Applied Mathematics and Computation*, 218(8):4025 – 4040.

Bowles, S., Edwards, R., and Roosevelt, F. (2005). *Understanding Capitalism: Competition, Command, and Change*. Oxford University Press, New York, 3 edition edition.

Budish, E., Cramton, P., and Shim, J. (2015). Editor's choice the high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621.

Charness, G., Oprea, R., and Friedman, D. (2012). Continuous time and communication in a public-goods experiment. University of california at santa barbara, economics working paper series, Department of Economics, UC Santa Barbara.

Di Stefano, A. and Santoro, C. (2005). Using the erlang language for multi-agent systems implementation. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IAT '05, pages 679–685, Washington, DC, USA. IEEE Computer Society.

Di Stefano, A. and Santoro, C. (2007). exat: an experimental tool for programming multi-agent systems in erlang. Technical report.

Gaffeo, E., Delli Gatti, D., Desiderio, S., and Gallegati, M. (2008). Adaptive microfoundations for emergent macroeconomics. *Eastern Economic Journal*, 34(4):441–463.

Gaffeo, E. and Gatti, D. (2008). *Emergent Macroeconomics An Agent-Based Approach to Business Fluctuations*. Springer Milan, Milano.

Gintis, H. (2007). The Dynamics of General Equilibrium. SSRN Scholarly Paper ID 1017117, Social Science Research Network, Rochester, NY.

Glasserman, P. (2015). Contagion in financial networks. Economics Series Working Papers 764, University of Oxford, Department of Economics.

Gode, D. and Sunder, S. (1991). Allocative efficiency of markets with zero intelligence (z1) traders: Market as a partial substitute for individual rationality. Gsia working papers, Carnegie Mellon University, Tepper School of Business.

Herbert, G. (2006). The Emergence of a Price System from Decentralized Bilateral Exchange. *The B.E. Journal of Theoretical Economics*, 6(1):1–15.

Jankovic, P. and Such, O. (2007). Functional programming and discrete simulation. Technical report.

Lehalle, C.-A. and Laruelle, S., editors (2013). *Market Microstructure in Practice*. World Scientific Publishing Co. Pte. Ltd.

Nilsson, H. and Perez, I. (2014). Declarative game programming: Distilled tutorial. In *Proceedings of the 16th International Symposium on Principles and Practice of Declarative Programming*, PPDP '14, pages 159–160, New York, NY, USA. ACM.

Nilsson, H., Peterson, J., and Hudak, P. (2003). *Functional Hybrid Modeling*, pages 376–390. Springer Berlin Heidelberg, Berlin, Heidelberg.

Parker, J. and Epstein, J. M. (2011). A distributed platform for global-scale agent-based models of disease transmission. *ACM Trans. Model. Comput. Simul.*, 22(1):2:1–2:25.

Poggi, A. (2015). Agent based modeling and simulation with actomos. In *WOA*.

Richiardi, M. (2007). Agent-based computational economics. a short introduction. LABORatorio R. Revelli Working Papers Series 69, LABORatorio R. Revelli, Centre for Employment Studies.

Schneider, O., Dutchyn, C., and Osgood, N. (2012). Towards frabjous: A two-level system for functional reactive agent-based epidemic simulation. In *Proceedings of the 2Nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, pages 785–790, New York, NY, USA. ACM.

Sulzmann, M. and Lam, E. (2007). Specifying and controlling agents in haskell. Technical report.

Tesfatsion, L. (2002). Agent-based computational economics. Computational economics, EconWPA.

Tesfatsion, L. (2006). Agent-based computational economics: A constructive approach to economic theory. In Tesfatsion, L. and Judd, K. L., editors, *Handbook of Computational Economics*, volume 2, chapter 16, pages 831–880. Elsevier, 1 edition.

Varela, C., Abalde, C., Castro, L., and Gulías, J. (2004). On modelling agent systems with erlang. In *Proceedings of the 2004 ACM SIGPLAN Workshop on Erlang*, ERLANG '04, pages 65–70, New York, NY, USA. ACM.

Vendrov, I., Dutchyn, C., and Osgood, N. D. (2014). *Frabjous: A Declarative Domain-Specific Language for Agent-Based Modeling*, pages 385–392. Springer International Publishing, Cham.