Research Proposal

Functional methods in agent-based modelling & simulation.

Jonathan Thaler

November 5, 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. Thus 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,...).

- Easy reproducibility
- Reasoning about convergence
- EDSL
- Qualitative modelling with quantitative results

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

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`
? Agent-based modeling and economic theory: where do we stand? - Ballot, Mandel, Vignes
? Agent-based Computational Economics. A Short Introduction - Richiardi
? Agent-based computational economics: a constructive approach to economic theory - tesfatsion
? Introduction to computer science and economic theory - blume, easley, kleinberg
? 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. **?**
- Debt . TODO cite
- Multiagent Systems Wooldridge. TODO cite
- Multiagent Systems Weiss. TODO cite
- Actor model Agha. TODO cite
- A computable universe. TODO cite
- The nature of computation. TODO cite
- Economics and Computation by Parkes and Seuken `http://economicsandcomputation.org/`
- Functional Compiler Design and Internals.

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.
- Get into category theory. Why: deeper understanding of Haskell type theory and computation.
- Get into theoretical basics of agent-based simulation and get to know more types of agent-based models. Why: To know the requirements my EDSL/framework has to cover.
- Understand theory of out-of-equilibrium / non-walrasian models: TODO (various Gintis & Mandel Papers)
- Understand Market Micro-structure: **?**, **?** Part II: Chapter 8-12.
- Do literature research on dynamics of equilibrium: **?**

4.4 Research on Papers (see the papers pdf)

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