PhD Project-Plan

Jonathan Thaler

January 17, 2017

This document gives detailed information about the structuring of the research undertaken in this PhD.

1 Years

The whole PhD lasts for 3 years, 36 Months, from October 2016 to September 2019 and thus I will structure it according to 3 years where each year will be a major milestone - which is also intended by the Computer School.

1.1 1st Year: Groundwork

In this year I will learn basics and develop and research the methodology I will use for the main work in the 2nd year. Also I want to write a paper about this work and try to publish it on a Conference or Journal. The time-frame will be set by the date of the 1st year annual oral report which will happen beginning of July thus there are about 6 Months time (counting including January '17 and leaving out July '17). These are the things I want / need to achieve this year:

- Prototyping in Haskell, Scala and Java
- Study Actor-Model theory: Hewitt, Greif, Clinger, Agha
- Get into reasoning about programs
- Basics of Economics [2], [3]
- Basics of ACE: Tesfatsion
- Implement PureAgents Library
- Write & publish paper
- Write 1st year report
- Prototype ACE using Akka and Haskell

Important and mile-stones:

April / May	Finished and submit Paper
June (Mid)	Finished writing 1st year report
July	Oral annual report

1.1.1 January

Research

- investigate and increase low performance of Haskell implementations
- implement environment-access in SEQ & PAR of Haskell
- implement concurrent semantic in Haskell

Paper

- Rework abstract
- 30min presentation of content

Reading

- Book "Gleichzeitige Ungleichzeitigkeiten"
- Complex Systems Stuff
- Find semantics of simulation papers

1.1.2 February

Research

- Embed SEQ & PAR in Yampa
- Experimenting

Paper Write first very basic draft

Reading

- Re-Read existing papers on functional programming and simulations
- Semantics of simulation papers

1.1.3 March

Research

- Implement SimTime Actors in Akka: have global simulation-time instead of local real-time
- Reasoning about Haskell-Implementations
- Continue Experimenting

Reading

- Read Semantics of Actors (Greif & Clinger)
- Semantics of simulation papers

Paper Write second refined draft

1.1.4 April

Research

- No programming: code-freeze
- continue experimenting

Paper Do refinements until final version

Reading

- Continue reading Semantics of Actors (Greif & Clinger)
- Semantics of simulation papers

1.1.5 May

Paper Submit to selected Conference/Journal (may also be at the end of April)

Reading Read papers not read so far but include in 1st year report

Writing Start writing 1st year report

1.1.6 June

Research Do 1st Year oral exam

Reading Read papers not read so far but include in 1st year report

Writing Finalize 1st year report

1.1.7 July to September

Research Prototyping ACE using my Haskell-Library and Akka: bilateral trading, EDSL for ACE and qualitative modelling, reasoning in ACE

Reading

- ACE basics papers (Tesfatsion)
- ACE models for trading/bartering (e.g. Gintis)

1.1.8 October

2nd year starts

1.2 2nd Year: Main Work

Applying 1st year results, methods and experiences to develop and write main paper to be published in a journal in 3rd year thus in 2nd year the main work and implementation will be done. The idea is to start from Ionescus Framework [**?**] and build on his paper.

- Implement Ionescous framework using the methodology developed in 1st year
- Generalize implementation to market models
- Learn Agda and dependent types
- Dig deeper into equilibrium theory
- Get into market-microstructure: [**?**], [1]
- Dig into *emergent properties* of systems. Can they be formalized?
- Get into basics of Category-Theory [**?**] [4]
- Get into basics of Type-Theory (found good lectures on Youtube)

1.3 3rd Year: Finalizing, Publishing & Writing

I plan to be finished - or nearly finished - at the end of the 3rd year. In this year I will finalize the work of the 2nd year, publish the my main journal paper (and optional fun-papers if possible) and will write down the thesis.
To have a bit of distraction and to prevent myself to become too locked in in writing on the thesis I will also work on my optional fun-papers (see below) and hope to at least finish them and maybe publish them - at least I want to present them to 2-3 audiences (e.g. FP Lunch) to test the reaction (especially the Genesis-Paper).

- Finalize research of 2nd year
- Publish journal paper
- Write thesis
- Work on fun-papers

2 Papers

This is the list of papers I have in my mind and includes both work mandatory for the PhD and optional ones. The latter ones are just fun/philosophical-papers, not directly related to my PhD but somehow tangential with the very basic direction - they are intended to be worked on in my free-time and to free my head when wrestling too hard with my PhDs main work.

2.1 Influence of Simulation-Semantics on Dynamics of Agent-Based Simulations.

Type: Groundwork
Target: Conference/Journal
Requirement: Mandatory

The first paper which describes how one can implement ABM/S in Haskell and compares the implementation and results to Java and Akka. A major focus are update-strategies, parallelism, reproducibility, reasoning and comparability between the various implementations. Actors: The Future in Agent-Based Simulation & Modelling? Although the actor-model is quite old (beginning of the 70s) it seems to have a revival both in Erlang in the 90s and now in the Framework Akka (based on Scala). It is one way of organizing highly parallel (and optionally distributed) applications. Also the actor-model is very close to the agent-metaphor where the latter one was strongly inspired by the former one. Thus It would be very interesting to look closer into how the Actor-Model can be utilized to ABM/S as it seems that this has not been properly done yet. This paper will establish my methodology in using Haskell / pure functional programming in the 2nd year main work.

2.2 Pure Functional ACE (Catchy title yet to be defined)

Type: Main PhD Work
Target: Journal
Requirement: Mandatory

Is the main work of the PhD and targeted at publication in a Journal. The exact topic and content will be clarified at the beginning of the 2nd year. Mainly it will describe how to implement Ionescus Framework of Gintis trading model and extend it to a more general Market-Model. It will also give an outlook on implementing it using dependent types.

2.3 Pure by Nature: A Library for pure Agent-Based Simulation & Modelling in Haskell

Type: Extension
Target: Conference

Requirement: Optional

This paper describes the ideas and theory behind the implementation of my ABM/S library "PureAgents" in Haskell.

2.4 Time in Games: a Tron Light-Cycle Game in Dunai

Type: Fun
Target: Conference
Requirement: Optional

This paper describes the 2D light-cycle game inspired by the movie Tron implemented in Dunai. It allows to turn back time.

2.5 Pure Functional Islamic Design

Type: Fun
Target: Conference
Requirement: Optional

Inspired by the paper "Functional Geometry" by Peter Henderson I had the idea to come up with a EDSL for declaratively describing pictures of islamic design which are then rendered using the gloss-library. From its focus totally unrelated to the PhD topic but still a great opportunity to learn Haskell, to learn to think functional, to learn to design my own EDSL - thus it may be a great paper to pursue even if I won't finish or produce something publishable.

2.6 The Genesis According to Computer-Science: Reality as Simulation of Free Will

Type: Philosophy
Target: ?
Requirement: Optional

I've always been interested in a deeper meaning behind things so I want to look into the philosophy and future of simulation: why do we simulate, what can we derive from simulations, what does it say that we humans simulate, what will the future of simulation be?
I claim that our ability to "simulate" in our mind separates our intelligence from those of the animals and that this is a unique property of humans. Also i think the future of simulation will be that humankind will do its own creation/live (artifical life, conciousness) which allows to accurately simulate a given setting - this of course could have ethical implications.

3 TODO-List as of January 17, 2017

1. Write John Garibaldi a Request for the School in April: Supervisors are OK with it but how can we finance it?
2. Arrange Meeting with Thorsten
3. Start Writing the paper
   - Watch Youtube-Video
   - 1 core message:
   - not too complex (Peer-Olaf)
   - Define Structure of Paper & send to Peer: Headings with each having 2 bullet-points
   - Consider Splitting the paper: influence of semantics / reasoning in functional ABM/S
   - Real Abstract
   - Title which sparks interest
   - Aim of Paper
   - what is the message of the paper? when implementing / running a simulation the semantics of the model MUST be considered otherwise the outcome could be completely different
   - what is the novelty of the paper? comparison with pure functional approaches
   - what is the contribution of the paper?
   -
4. Ideas to capture in the paper
   - If one wants to write global in case of parallel this is regarded as a systematic error as this is not logical as it would imply an ordering thus we requiring different semantics: SEQ or CONCURRENT. Thus we would have to make the Environment in case of Par local to an agent which is the same as moving it into the agents state =¿ we choose another approach: pass in an environment which cant be changed by the agents (no environment in return type) but only by the simulation-iterator after an iteration. =¿ dynamic WildFire-Model does not work with PAR
   - note that the difference between SEQ and PAR in Haskell is in the end a 'fold' over the agents in the case of SEQ and a 'map' in the case of PAR

- haskell: dont have objects with methods which can call between each other but we need some way of representing agents. this is done using a struct type with a behaviour function and messaging mechanisms. important: agents are not carried arround but messages are sent to a receiver identified by an id. This is also a big difference to java where don't really need this kind of abstraction due to the use of objects and their 'messaging'. messaging mechanisms have up- and downsides, elaborate on it.
- reason for patterns: heroes try to stay 50 % in between and have selected 2 cowards which themselves are at the borders opposite. need much more cowards than heroes 75/25.
- parallel more natural in haskell
- sequential more natural in java
- concurrent difficult in both, using stm in haskell it becomes very natural, STM available in java too. actors are an even better approach but is having problems with time in simulation
- question: do emergent patterns break down / global dynamics change completely in some ABM/S when changing sim-semantic? which kind of ABM could show this behaviour? which properties are responsible for it? Answer: Yes they do but only under given circumstances: discrete simulations with dependence on each other. continuous not so easy.
- do we find a continuous simulation in which it breaks down under given circumstances? Yes: Heroes & Cowards can lead to specific patterns as shown by the creators

5. SEQ: two possibilities: time stays constant or time advances with every agent

6. FINALIZE AND FREEZE the literature I use for my paper (see next items READ): its all on my desk plus some prints to do (organize papers i need to read: ACE & Economics, complex systems & simulation, functional programming, abm & actor model)

    - READ: Gleichzeitige Ungleichzeitigkeiten
    - READ: Complex Systems 'Cellular Automata' & 'Turing Machines'
    - READ: Functional programming & Simulation papers (on desk)
    - READ: Semantics of Actors (Hewitt, Greif & Clinger)
    - READ: philosophical papers of simulation
    - READ: functional programming papers
    - SEARCH: for further papers looking into my direction (when the difference between SEQ and PAR matters / matters not)

7. Conference: ESSA (checkout website), Informs Proceedings Winter Simulation

8. Bring PureAgents to Yampa

   - embed PureAgentsPar and Seq in Yampa: PureAgentsYampa so we can leverage the power of the EDSL, SFs, continuations,... of Yampa/Dunai.
   - implement agent monad: PureAgentsMonadic. but what is an AgentMonad? build a monad to chain actions of the agent and always run inside an agent-monad
   - embed PureAgentsMonadic in Dunai
   - implement wait blocking for a message so far. utilize yampas event mechanism?

9. Push PureAgentsYampa by considering the next step: implementing an economics example at different levels of complexity e.g. Ionescus implementation of Gintis

10. Always ask the question: what is the difference to the OO approach?

4 Future-List as of January 17, 2017

1. Topics & Issues of Haskell Implementation

   - performance unacceptable: 1000 in haskell vs 100.000 in java is a shame on haskell, more should be possible. investigate using profiling both of CPU and memory: `http://keera.co.uk/blog/2014/10/15/from-60-fps-to-500/`. strictness & tail-recursion!
   - look into QuickCheck and HPC
   - problem: so far only agents with same static messagetypes, environment and states, can communicate: the agents are homogenous. how can we implement hetereogenous agents in this library?
   - implement a general-purpose rendering-frontend with gloss but let the simulation be driven by Yampa/SimulationBackend instead of frontend (not real-time)
   - implement PheroTrails: agents move on a 2d grid in a 8-neighbourhood and leave pheromone-trails which decay over time. the agents select the neighbourhood cell they move in the next step randomly according to the amount of pheromones present: the more pheromones are present the more likely they will move to that cell - note that this is relative: the pheromones of all neighbour cells are added up and normalized!

2. Simulation Model Ideas

- IDEA: abm/s of a go game
- IDEA: what about an ABM/S of karma and rebirth? add to genesis paper
- IDEA: what about ABM/S generating sound? could be a perfect example for Yampa due to its signal functions. the sound is the result of interactions of agents which try to generate harmonies and agents trying to create dissonance
- IDEA: what about abm/s creating drawings/art? 2d continuous and each agents path is drawn

References

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

[2] Bowles, S., Edwards, R., and Roosevelt, F. *Understanding Capitalism: Competition, Command, and Change*, 3 edition ed. Oxford University Press, New York, Mar. 2005.

[3] Kirman, A. *Complex Economics: Individual and Collective Rationality.* Routledge, London ; New York, NY, July 2010.

[4] Spivak, D. I. *Category Theory for the Sciences*, 1 ed. Mit Press Ltd, Cambridge, Massachusetts, Nov. 2014.