PhD Project-Plan

Jonathan Thaler

March 13, 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. The year will be guided by the principal question of "How can Agent-Based Simulation be done using pure functional programming?".
Important mile-stones:

31st March 2017	Finished and submit Paper
June (Mid) 2017	Finished writing 1st year report
End of June / Begin of July 2017	Oral annual report

1.1.1 Major Activities

- Develop FrABS library: March 2017 - May 2017
- Refine FrABS and publish as library on Hackage: July 2017 - September 2017
- Write FrABS paper: July 2017 - September 2017
- Study decentralized bartering: July 2017 - September 2017

1.1.2 March 2017

- experiment with MetaABS (also implement it in the EDSL) to see if it is making any sense to follow this road
- Programming: bring ABS to FRP in Yampa: FrABS, implement SugarScape
- 31st March: Finalize and submit 'Art of Iterating'-Paper to SSC 2017
- Formulate Research Questions
- Draft Project Plan

1.1.3 April 2017

- 9th - 13th: Midland Graduate School in Leicester
- Programming: bring ABS to FRP in Yampa: FrABS, implement SugarScape
- Refine Research Questions
- Refine Project Plan

1.1.4 May 2017

- Start writing 1st year report
- Programming: bring ABS to FRP in Yampa: FrABS, implement SugarScape
- Finalize Research Questions
- Finalize Project Plan

1.1.5 June 2017

- 4th - 18th: 2 weeks holiday on Amrum (planned last year already)
- Finalize 1st year report
- Prepare for 1st Year oral exam
- End (or beginning of July) 1st Year oral exam

1.1.6 July 2017

- Start writing on FrABS Paper: show how the rules of SugarScape can be formalized in FrABS =¿ specification equals code
- Study decentralized bilateral trading/bartering
- Refine FrABS library

1.1.7 August 2017

- Work on FrABS Paper: Formalize the Rules in the EDSL of FrABS
- Study decentralized bilateral trading/bartering
- Refine FrABS library

1.1.8 September 2017

- Finalize FrABS Paper
- Study decentralized bilateral trading/bartering
- Refine FrABS library: put on hackage

1.1.9 October 2017

2nd year starts

1.2 2nd Year: Main Work

Applying 1st year results, methods and experiences to work in the Research-Questions. The second year is investigating the question of "What are the benefits of doing ABS in pure functional programming?".
Important mile-stones:

October	Finished FrABS Paper
?	Submit FrABS Paper
?	Finished and submit MetaABS Paper

1.2.1 Major Points for research

- EDSL which is both specification- and implementation- language
- MetaABS which allows to do recursive simulation
- Reasoning about dynamics and emergent properties of ABS

1.2.2 October 2017

Start of 2nd year

1.2.3 December 2017

- 22nd December 2017 - 7th January 2018: 2 weeks xmas holidays

1.2.4 January 2018

- 22nd December 2017 - 7th January 2018: 2 weeks xmas holidays

1.2.5 April 2018

1 week holiday

1.2.6 August 2018

2 weeks holidays

1.2.7 October 2018

3rd year starts

1.3 3rd Year: Finalizing, Publishing & Writing

The third year serves to refine, finish and publish the research of the 2nd year (ideally a journal paper) and then to write the final 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 philosophical paper 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.
Important mile-stones:

30th September 2019	End of official PhD

1.3.1 October 2018

3rd year starts

1.3.2 October 2018 - December 2018

Finalize research of 2nd year

1.3.3 22nd December 2018 - 6th January 2019

2 weeks xmas holidays

1.3.4 January 2019 - March 2019

Publish journal paper of 2nd year, define structure of PhD Thesis

1.3.5 April 2019

1 week holiday

1.3.6 April 2019 - August 2019

Writing thesis

1.3.7 September 2019

Submitting final thesis, 2 weeks of holidays

1.3.8 October 2019

End of official PhD

1.4 4th Year: Pending Period

It is very hard to finish ALL within the 3rd year and it is very likely that I will enter the 4th year - pending period. I plan on spending in this period not more than till Christmas as I have no funding in this time and I want to finish within time.
Important mile-stones:

1st October 2019	Start of pending period
30th September 2020	End of pending period

1.4.1 October

- Prepare for viva

1.4.2 November

- Viva

1.4.3 December

- Refine Thesis (incorporate minor changes)

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

Type: Groundwork
Target: Conference
Requirement: Mandatory

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 properties, derive the possible update-strategies in ABS and discuss their interpretation and semantics something which is still lacking in the literature on ABS. Further we investigate the suitability of the three very different programming languages Java, Haskell and Scala with Actors to implement each of the update-strategies. Thus this papers contribution is the development of a new, general terminology of update-strategies and their implementation comparison in various kinds of programming languages.

2.2 Specification equals Code: An EDSL for pure functional Agent-Based Modelling & Simulation

Type: Main PhD Work
Target: Conference/Journal
Requirement: Mandatory

In our previous work on update-strategies in Agent-Based Modelling & Simulation (ABM/S) we showed that Haskell is a very attractive alternative to existing object-oriented approaches but our presented approach was too limited and we hypothesized that embedding it within a functional reactive framework like Yampa would leverage it to be able to build much more complex models. In this paper we investigate whether this hypothesis is true by testing if our approach can easily be transferred to Yampa, what we really gain from it and if more complex models can become reality. Also we ask in this paper if the declarative power of the pure functional language can be utilized to write specifications for ABS models which can be directly translated to our Haskell implementation. As a proof-of-concept we implement the famous SugarScape model to proof that FrABS is able to implement also very complicated agent-based models and to see if we can formalize the rules of the model into our EDSL.

2.3 MetaABS: Recursive Agent-Based Simulation

Type: Groundwork / New Idea
Target: Conference
Requirement: Mandatory

2.4 Reasoning about dynamics and emergent properties in Agent-Based Simulations

Type: Main PhD Work
Target: Journal
Requirement: Mandatory

Is the main work of the PhD and targeted at publication in a Journal.

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

Type: Philosophy
Target: ?
Requirement: Optional

All sciences have their genesis-model which are basically explanations of how the world did come into existence, what the reason for existence is and who or what God is. Unfortunately computer science has none so far, so the aim of this paper is to set out to develop such genesis-model from a computer-science perspective. The model is motivated from the perspective that the world and humankind is a simulation to see free will in action in a sand-boxed environment as opposed to the afterlife / after-world or just the Beyond, which is an outer level of simulation and itself again a simulation as will be shown in subsequent sections. This paper addresses important fundamental questions of belief and religion and tries to explain them using this model - which it does surprisingly well. Thus this paper has a keen aim: it wants to amalgamate religious concepts with concepts of theoretical computer science. We build on the discussion of the *simulation argument* proposed by [1] which states that we *may* live in a simulation, created by transhumans (TODO: explain). Although we neither can proof or disproof the *simulation argument*, we find the idea of existence being a simulation highly intriguing as it finally allows us to *investigate how existence can be understood from the perspective of computer science in a scientific way.* This was so far not possible due to the lack of a scientific context, which is now given through the *simulation argument.* The obvious question which raises, is then *why are we simulated?*. Bostrom does not give any reason for it in his paper as this obviously touches on ideological and religious ground. Our main hypothesis is that if we are living in a simulation, then it is to simulate Free

Will. Further we hypothesize that the emergent properties of this simulation are the emergence of ideologies, brought forward by Free Will.

3 TODO-List as of March 13, 2017

1. Refine research questions
2. Refine project-plan
3. work on FrABS
4. experiment with MetaABS

References

[1] Bostrom, N. Are We Living in a Computer Simulation? *The Philosophical Quarterly 53*, 211 (2003), 243–255.