1st Year Report
Pure Functional Methods in Agent-Based Modelling & Simulation

Jonathan Thaler
jonathan.thaler@nottingham.ac.uk

April 24, 2017

Abstract

A succinct and concise summary (250 words maximum) of the report contents and presented on a single page.

So far specifying Agent-Based Models and implementing them as an Agent-Based Simulation (ABS) is done using object-oriented methods like UML and object-oriented programming languages like Java. The reason for this is that until now the concept of an agent was always understood to be very close to, if not equals to - which it is not - the concept of an object. Therefore, the reasoning goes, object-oriented methods and languages should fit themselves naturally to specify and implement agent-based simulations. In this PhD Thesis we fundamentally challenge this assumption by investigating how Agent-Based Models and Simulations can be specified and implemented using pure functional methods and programming in Haskell and what the benefits are. We will show that the implicit assumption that an Agent is *about equal to* an Object is not correct and leads to many implicit assumptions in object-oriented implementations of Agent-Based Simulation (ABS). When implementing ABS in Haskell these implicit assumptions become explicit and challenge the fundamental assumptions about ABS and Agents. We present these implicit assumption in an explicit way by approaching it through programming, type-theory and category-theory to further deepen the concepts and methods in the field of Agent-Based Modelling & Simulation. We also think that the major benefit of implementing ABS in Haskell is the potential for an unprecedented approach of formal validation & verification of an Agent-Based Model and its implementation. Due to the declarative nature of pure functional programming in Haskell it is possible to implement an EDSL for ABS which ideally results in code which looks like specification thus closing the gap between specification and implementation because the specification is already the code. For validation we want to pursue testing through QuickCheck.

In this report I discuss the research conducted to far, present the open problems together with an in-depth literature review. An outline for the research of the following 2 years is given and the aims.

1 Introduction

TODO: Discuss the key aspects of the PhD project such as: background material, context, motivation, aims and objectives. Include a succinct and concise account of the progress and achievements during this stage of the PhD.

Direction of Research: ABS mit FRP durch FrABS. SugarScape und Agent_Zero implementieren BEGRueNDEN wieso FP in ABS. Robustness, Testbarkeit mit Quickcheck, Verification, Validierung, Code = Spezifikation

1.1 Problems of ABS in OO

- Objects don't compose - implicit state, change through effectful computations - blurring of fundamental difference between agent and object: an agent is a metaphor, it is much more than an object. an object is: a uniquely identifieable compound of functions (=methods) and data

1.2 Problems of ABS in general

Specification: how is my model specified? Verification: does my implementation really match my specification? Validation: how to connect the results to the hypothesis? are the emergent properties the ones anticipated? if it is completely different why? note: we always MUST HAVE a hypothesis regarding the outcome of the simulation, otherwise we leave the path of scientific discovery. But we must admit that sometimes it is extremely hard to anticipate *emergent patterns*. But anyway there must be *some* hypothesis regarding the dynamics of the simulation.

1.3 Functional approach to Agent-Based Modelling & Simulation

Because we left the path of OO and want to develop a completely different method we have fundamentally two problems to solve in our functional method: 1. Specifying Agent-Based Models 2. Implementing the Agent-Based Models and running the ABS

2 Conducted Research

TODO: Describe the research work carried out during this stage of the PhD and the outcomes. A literature review must be included. Then, as appropriate according to the PhD project, this section can also include theoretical and/or experimental methods, presentation and discussion of results, etc. In the case that papers have been submitted or published within the year of the review, this section can be shorter and focused on discussing the outcomes from those papers within the wider context of the PhD programme of study (papers to be included in the appendix).

Literature Review - ask other students: Tuong, Olusola, Ivan for their 1st year reviews - trichter: mit den 3 themen beginnen und dann runterbrechen und ins detail gehen, bis der gap gefunden wurde

the really unique thing which is ONLY possible in pure functional progranning is composition of concurrency

2.1 Category Theory

include paper on arrows my hughes

apply category theory to agent-based simulation: how can a ABS system itself be represented in category theory and can we represent models in this category theory as well?

ADOM: Agent Domain of Monads: https://www.haskell.org/communities/11-2006/html/report.html

develop category theory behind FrABS: look into monads, arrows

3 Conclusions

TODO: Provide a succinct account of the conclusions from the report, stating clearly the research questions that have been identified during this stage of the PhD and the progress so far towards addressing those questions.

4 Future Work Plan

TODO: A future work plan that is consistent with the progress to date, stating clearly the research question(s) to be addressed during the next year of the PhD.

TODO: gantt chart!

4.1 TODOs

out of this i will build the gantt chart for the next 12 months+

4.1.1 Category Theory

develop category theory behind FrABS: look into monads, arrows

category theory foundations (monads, arrows)

4.1.2 Implementation and Software-Engineering

implement chapter 4 of sugarscape implement chapter 5 of sugarscape use monadic or arrowized programming for structuribg the siftware implement schelling segregation in recursiveABS and report results

4.1.3 Verification and Validation

look into QuickCheck to test and verificate FrABS. start with SIRS (quickcheck, isabelle, agda?), recursive simulation

4.1.4 Papers

paper 2: recursive ABS paper 3: FrABS - Towards pure functional programming in ABS paper 4: Towards category theory in ABS paper 5: verification and validation in ABS with pure functional programming

4.1.5 Reading

read "Writing For Computer Science" read "Agent_Zero" read "category theory for the sciences"

4.2 Concept of an Agent

an agent is not an object but when implementing ABS in oo then it is tempting to treat an agent like that. when implementing it in a pure functional language like haskell, this temptation cannot arise which creates a different view on agents.

5 Appendix

Material that is complementary to the main body of the report can be included in an appendix. For externally sponsored students, if a report has been submitted to the sponsor during the year of the review, the report should be included in the appendix (a copy of the report can be supplied by the PGR coordinator). The appendix should include a list of training courses (including dates, duration, etc.) taken by the student during the year and other relevant research activities such as given seminars, attendance and presentations to conferences, etc. The appendix could also include material that is supplementary to the main body of the report such as: description of data sets, detailed experimental results, papers that have been submitted or published, etc.