PhD Thesis

Foundations of Pure Functional Agent-Based Simulation

Jonathan Thaler (4276122)
jonathan.thaler@nottingham.ac.uk

supervised by
Dr. Peer-Olaf Siebers
Dr. Thorsten Altenkirch

April 23, 2018

Abstract

TODO

Contents

0.1 Introduction

first year report, FpOOPAbs report,

Part I: General Concepts

FpOOPAbs report, abs defined, pure functional programming defined, artiterating paper, towards pure functional paper

Chapter 1

Part II: Pure functional ABS

concepts of time- and event-driven approach in haskell (see robinson book and pidd book) FrABS report, pure functional epidemics paper additional research on event-driven approach in haskell: unscheduling events in functional style easy: rollback to previous state is easy but memory costly. look into that in the thesis. clarify easy rolling back of system: can capture the whole state at a given point which allows reverting the system to a state in case of cancelling of an event applicability of UML and peers Framework to pure functional ABS add section on recursive ABS parallelise using cloud haskell?

agent-interactions 1) data-flow: for continuous ABS systems where data takes 1 dt to appear at the target agent and does not persist e.g. the target agent can check it or not but the data received in the current step will be gone in the next. Use-case: implementing continuous time-dependent ABS e.g. SIR model 2) events: an agent can send to an arbitrary other agent an event which happens at a given time in the future: when the event happens this means the target agent is executed with the information about the receiving event. Use-Case: implementing discrete event simulation ABS e.g. a bank, very useful when the model is specified in terms of events and not in a continuous fashion 3) transactions: method-call emulation, which takes no time at all and can involve an arbitrary but finite number of steps between agents. Use-case: trading between agents where they must to come to terms within the same time-step but where the negotiation process takes multiple steps between the agents.

Chapter 2

Part III: Dependent types in ABS

3rd paper

Chapter 3

Part IV: Verification

Chapter 4

Conclusions