Thesis proposal - two pager

Stop reinventing the wheel:
A formal ABM framework for infinitely stackable behavior.

Agent-based modeling has been around for decades. In that period, thousands of models were developed. However, aside from basic built-in functions of popular ABM libraries, all models have been created mostly in isolation, from scratch.

In the meantime, many psychological, sociological and economical papers have been published which describe all kinds of complex behaviors. Many have been implemented, but with each implementation there are underlying assumptions, which can differ from implementation to implementation. This is not necessarily bad, since each model might have different requirements and thus will cut different corners by applying Ockham's razor on their specific problem. However, it makes these behaviors conceptually incompatible with each other and poses an implementation risk and limits the maximum attainable scope for any project with a given (time) budget.

Interesting, often emergent, behavior of ABMs rises from complexity generated by relatively simple agents. However, in the field of policy analysis very often humans are modeled. Humans are multidimensional decision makers, that not only can be a socially and culturally very heterogeneous group, but also make decisions on different levels (like system 1 and system 2 thinking). So to get interesting but representative behavior for humans, a certain amount of complexity can be needed.

Implementing these complex behaviors from scratch is always a lot of work, especially if you want to properly test, validate and document those. Then, often the behavior only works under some specific circumstances with some specific data. Furthermore, the implementation itself can be done in many languages, using many constructs, and many definitions of what means what. Those implemented behaviors are very unlikely to work with each other, without major work.

Conceptually, some work has been done to solve this problem. The most recent effort is the effort by Berger et al (2023). In this paper, a detailed conceptual framework is proposed for reusable building blocks. However, this research only works on a conceptual level, without considering the formal implementations. This will most likely create the problems discussed above: Different languages, constructs, interface levels and disagreement about variables.

Therefore, it would be beneficial to have a formal ABM framework for behaviors that are guaranteed to be compatible with each other. Each component of this building block structure should be fully specified and formalized.

The prime directive here is that instead of all behaviors having to talk to each other, all communication happens through modifications of the agent's own state, other agents' states and the environment state. This reduces the number of possible communications lines, and thus integration pain.
To properly propose a formalized ABM language, the following components are needed:

- A programming paradigm to efficiently, scalably and easily integrate different building blocks into each other, without causing incompatibilities.
- A formal dictionary of ABM states, mapping names to units, data formats and definitions, so that each building block agrees what each state and variable is.
- A formal specification of how a building block should be implemented, documented, tested, validated and maintained.
- (optionally) infrastructure to review, test and validate building blocks as they are added.
- (optionally) a proof of concept proving the components above.

Research question
From this, we can derive a main research question:

> What design patterns and architectural approaches are optimal for creating a modular and interoperable ABM framework of behavioral components in Python.

The research could be split in two parts, with each three subquestions:

Part 1: Formalizing a framework

A. How can a standardized set of agent types, data formats and variable definitions be established to ensure uniformity and clarity in behavioral components for ABM?
B. What strategies and frameworks are effective for testing the functionality, interoperability, and performance of the ABM framework in diverse modeling scenarios?
C. How can the building blocks be designed to be universally applicable across a diverse range of agent types, while conforming to broad predefined temporal and spatial categorization schemes?

Part 2: Building, testing and integrating the framework

D. How can this framework of building blocks be integrated in existing ABM libraries like the Mesa?
E. How can open-source development practices and community engagement be leveraged to enhance the robustness, adoption, and continuous improvement of the framework?
F. What methodology should be adopted to demonstrate the practical application of the ABM framework in policy analysis, and how can its impact on simplifying the modeling process be evaluated?

A structured list of questions and potential directions

Part 1: Formalizing a framework

A. How can a standardized set of agent types, data formats and variable definitions be established to ensure uniformity and clarity in behavioral components for ABM?
    a. How can we standardize agent types?
        i. Do we need nestedness?
        ii. Do we need domain specific agent types?
        iii. How strict should we manage agent types?
    b. Data formats
        i. Should data formats be linked to agent types?
        ii. If so,
B. What strategies and frameworks are effective for testing the functionality, interoperability, and performance of the ABM framework in diverse modeling scenarios?
C. How can the building blocks be designed to be universally applicable across a diverse range of agent types, while conforming to broad predefined temporal and spatial categorization schemes?

Part 2: Building, testing and integrating the framework

D. How can this framework of building blocks be integrated in existing ABM libraries like the Mesa?
E. How can open-source development practices and community engagement be leveraged to enhance the robustness, adoption, and continuous improvement of the framework?
F. What methodology should be adopted to demonstrate the practical application of the ABM framework in policy analysis, and how can its impact on simplifying the modeling process be evaluated?

Definitions

- How do we synchronize the meaning (of agents, variables and states)?
- How do we hierarchically inventorize and create distinct subdomains?
- How do we reach consensus on the above?
- How do we allow adjustments to existing code without breaking everything?

All the open source and governance questions → Future research

Goals:

- Prove that it can be done (technically).
- Give the ABM world something to shoot on
  (someone just needs to build the wrong thing)

Needed:

- A really strong definition of a behavior
    - Conceptually: What goes in it?

    - Formally: How exactly does it go in it?
- A code structure or paradigm to integrate low-level behaviors into high-level behavior into agents.
    - How can multiple low-level behaviors be integrated in a single, coherent high-level behavior?
        - Mixins? Strategy patterns?
    - How can behaviors be integrated into agents?
- A mechanism to synchronize the meaning (of agents, variables and states)
    - All information is spread through agent states and models
    - How do we agree what means what?
    - Technically, how can we enforce data types and meanings?
    - How can we store and distribute those definitions?

TODO
- Read up about model-driven architecture. Collect relevant ABM papers.
- Check how they solved similar problems in other modeling fields
    - Discrete event - Omar
- Determine levels to solve this (math to impl) and choose one, preferably about the programming language
- Think about non-obvious behavior conflicts and how to prevent them
- Start building small things
    - Think about composition and composition order while doing that
- Update this proposal if necessary

Literature

- **Model Driven Architecture**
  - Kind-of relevant: The paper "Application of Model Driven Techniques for Agent-Based Simulation" by Rubén Fuentes-Fernández and colleagues focuses on enhancing agent-based modeling (ABM) for social systems simulation using software engineering methods, specifically model-driven methodologies. They propose using INGENIAS, a model-driven approach, to create more efficient and understandable models for non-experts, facilitating multidisciplinary collaboration. The paper demonstrates this methodology through a case study on urban dynamics, emphasizing the benefits of using domain-specific modeling languages and model transformations for agent-based social simulations. This approach addresses common challenges in ABM, such as the difficulty in understanding and validating complex models, by improving communication between experts from different fields and enhancing model transparency and reproducibility. https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/978-3-642-12384-9_11
  - Not my problem. This paper by Candelaria Sansores and Juan Pavón discusses the replication of multi-agent based simulation (MABS) systems using a Model Driven Architecture approach. The authors highlight the importance of replication in MABS for improving reliability and understanding of the system, while acknowledging the challenges posed by the lack of integrated environments supporting the entire research process from modeling to implementation and analysis. Their proposed solution focuses on high-level conceptual modeling to drive the simulation development, with transformation tools to facilitate simulation implementation across various platforms. This approach allows users to concentrate on conceptual modeling, while the implementation code is generated automatically, thereby enhancing the simulation development process and aiding in the replication and validation of simulation models. https://www.researchgate.net/profile/Juan-Pavon/publication/220887190_Agent-Based_Simulation_Replication_A_Model_Driven_Architecture_Approach/links/0fcfd50cac5e6efc0c000000/Agent-Based-Simulation-Replication-A-Model-Driven-Architecture-Approach.pdf
  - Not my problem. The paper "Agent-Based Simulation Replication: A Model Driven Architecture Approach" by Candelaria Sansores and Juan Pavón focuses on improving the replication process in multi-agent based simulation (MABS) systems. It addresses the challenge of the lack of integrated environments supporting the entire research process from conceptual modeling to simulation implementation and analysis. The authors propose a high-level conceptual modeling abstraction for simulation development, using transformation tools to facilitate simulation implementation across different platforms. This approach emphasizes modeling-driven simulation development, allowing users to focus on conceptual modeling while automatically generating implementation code. The paper also highlights the importance of replication in verifying the reliability of simulation results and contributing to a deeper understanding of the modeled systems. https://www.researchgate.net/publication/220887190_Agent-Based_Simulation_Replication_A_Model_Driven_Architecture_Approach