AgentsX.jl — An Extended Julia Framework for Exploring Urban and Social Systems ⋆

Rajith Vidanaarachchi[1,2][0000−0003−1645−0084], Jason Thompson[2], Branislava Godic[2][0000−0003−2976−7689], and Rod McClure[1]

[1] Faculty of Medicine and Health, The University of New England, Australia
[2] Transport, Health, and Urban Design (THUD) Research Lab, The University of Melbourne, Australia
{rajith.vidanaarachchi, jason.thompson, b.godic}@unimelb.edu.au

Abstract. Agent-Based Modelling (ABM) is applied successfully in various use cases, including but not limited to economic modelling, socio-behavioural modelling, ecological modelling, public health, and urban design. We draw attention to the emerging ABM platform Agents.jl, written in Julia — an accessible, high-level programming language. We identify Agents.jl as a promising package for interfacing, customising and extending for specialised uses. We present the key design ideas for a proposed extension and interface to the Agents.jl framework for ABM for urban and social systems simulation — AgentsX.jl — that provides greater flexibility of agent definitions for urban and social researchers. Our primary motivation is to formalise ABM design through a "Code as the Model" approach, reducing barriers to documentation and increasing reproducibility. Our proposed design entails a structured means of defining an ABM based on layers modelled after spheres of influence, clearer constructs to coding an ABM as interfaces to Agents.jl, and an insightful visualisation toolkit that uses dimension reduction techniques.

Keywords: Agent-Based Modelling · Urban Simulation · Social Simulation · Julia · Agents.jl

1 Introduction

Agent-Based Models (ABMs) are useful tools for modelling urban and social systems. ABMs are often used in an interdisciplinary manner to inform large-scale policy decisions, to understand complex behaviours, and to aid in planning and designing within socio-technical systems. ABMs at their core involve the creation of an artificial society containing 'agents', which can represent any decision-making entity that acts along with a set of 'rules' or 'behaviours' defined by the modeller and reveals emergent phenomena [1]. ABMs are useful for understanding urban systems as they allow the 'agents' or the entities within the model to have a unique set of characteristics which leads them to be located

⋆ Supported by The University of New England and The University of Melbourne

in a simulated 'neighbourhood'. Each neighbourhood can have its own characteristics which influence the agents' interactions with their environment, with emergent consequences on factors such as the health and financial status of the agent [2]. There are extensive examples of applications of ABMs in the context of urban systems, including transport, residential choice, urban growth and expansion, urban food access, urban planning and health, and urban systems generally [3–8].

1.1 Background

There are a host of Agent-Based Modelling platforms available, including Netlogo [9], Swarm [10], MASON [11], Repast [12], Mesa [13], and the newly minted Agents.jl [14]. Among these, we identify Agents.jl as a platform with much potential (for review, see [15]). It has a simple back-end, is fast, and the underlying language — Julia — is a well-supported general programming language.

As ABMs are often used interdisciplinarily, it is crucial that a human-readable definition accompanies them. A proper definition of an ABM has two roles. Firstly, it allows for a better understanding of the model by users. Secondly, it increases reproducibility. We identify two noteworthy approaches to these issues. First is the ODD (Overview, Design Concepts, Details) by [16], which is the formal documentation method in the ABM field. Second is the "Code as the Model" approach argued for in [17]. Both have documented strengths and weaknesses [18]. The latter of which may be overcome using this extension.

Formal methods are often employed where computational methods are applied in critical systems. When ABMs are used in urban and social simulations that affect policy decisions, it is not unreasonable to expect a certain degree of formalisation. [19–21] present attempts to formalise ABMs.

1.2 Motivation and Contributions

In this paper, we introduce AgentsX.jl, which is an extension to an ABM framework based on the Agents.jl platform. The framework's design reflects an effort to formalise ABMs and improve the ease of human readability. We achieve these goals through the extension and interfacing of the Agents.jl platform and present a framework specifically suitable for urban and social simulation.

We are influenced by the arguments presented in [17] that code is the ultimate definition of a computational model. However, we believe that code needs to be structured in a manner that serves as a robust, readable definition. Hence, we provide infrastructure to generate ABMs that would represent unambiguous models. We also recognise the importance of visual documentation as mentioned in ODD [16]. We, therefore, also provide infrastructure for model developers to generate simple automated visual representations of ABMs, reducing barriers to visual documentation.

A complementary contribution that stems from the layered structure we introduce is the visualisation of Agent movement and behaviour beyond the traditional spatial domain, which allows analysis and actuation in spatial and alternate (e.g., social) domains, as visualised in Fig. 2.

2 Methods

This paper discusses the design and the initial development of work-in-progress that is available at github.com/rajithv/AgentsX.jl. This work is an actualisation of a conceptual framework that seeks to formalise ABMs. The Julia language implementation of the framework eases barriers into coding a robust Agent-Based Model through facilitating formalisation through code generators.

We present AgentsX.jl as an extension and an interface to the Agents.jl platform that would cater specifically to urban and social simulation paradigms. Primary reasons to keep the work distinct from the base Agents.jl package is due to the specialisation into social simulation support that would be unnecessarily complicated for simulations solely dedicated to the spatial domain that is already well supported through Agents.jl.

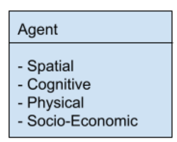

Fig. 1. A layered agent

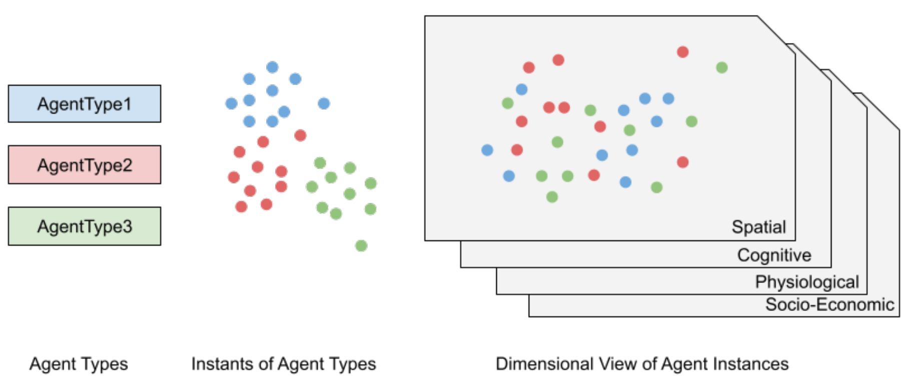

Fig. 2. Multiple agent types provide instances to the ABM that inhere independently in different dimensions

As part of our extension, we propose a structure that separates the agent definition in attributes and actions through clearly defined source code files. To formalise the definition of attributes, we introduce a new abstract agent type — LayeredAgent — that recognises the heterogeneity of agent attributes. A standard example is an agent with Spatial, Cognitive, Physiological and Socio-Economic attributes as seen in Fig. 1. Each layer is modelled as a parametric layer, enabling agents to be visualised through any subset of layers, not just in the spatial domain, as is the case with traditional visualisations. We formalise the actions by separation of responsibilities in the source code from the attributive definition of the agent and by defining the scope of the actions as intra- or inter-layer actions taking place between agents and the environment.

We facilitate this formalised structure through a code generator similar to that of PkgTemplates.jl package generator and use Mustache.jl based templates in its implementation.

```
create_agent_template(;agent_class, num_agents, space,
    random_seed = 250, layers = Nothing, actions = Nothing)
```

We provide two main interfaces to the existing Agents.jl architecture. Firstly, we provide a generalised step function and defer the definition of an agent's 'step' to the definition of the agent itself through an ordered list of actions or a function that describes a sequence of actions. Secondly, we provide a simpler construct to adopt multi-agent models by overriding the AgentBasedModel() function.

```
ABM(agents::Array{<:AbstractAgent}, args...; kwargs...)
```

We propose the following rules for layers and actions — and enforce the same in AgentsX.jl — to ensure a robust ecosystem for ABMs.

Layers

1. All agents share a single environment.
2. Agents in a model can be of different types.
3. Agents have layers describing different internal domains.
4. Different agent types have different subsets of layers.
5. The environment has a set of corresponding layers that is equivalent to the superset of the union of layers of each agent type.
6. Layers are described through parameters.

Actions

1. The environment is omniscient.
2. Interactions could be within the agent (self-interactions), between multiple agents, or between the agent and the environment.
3. Agent-agent interactions must be facilitated through the environment.
4. Interactions could be inter- or intra-layer interactions.

3 Conclusion

The design of the AgentsX.jl framework allows the formalisation of Agent-Based Models in a programming language that is fast, easy to use and future-proof. The proposed structure, together with the code generator, would allow the reproduction of ABMs with minimal ambiguity. Moreover, the layered agent design would improve the overall design of agents with respect to clarity of agent behaviour, data collection, isolation of agent domains, experimentation, and analysis.

We suggest that the proposed interfaces to Agents.jl would facilitate coding practices that contribute to the "Code as the Model" approach and enhance clarity, communication, and reproducibility for users. The layer-based reduced dimension visualisations would allow modellers to look beyond the spatial arrangement of agents in single domains, allowing more sophisticated agent interactions in other (any) parameterised agent domains.

3.1 Future Work

The current design of the Agents.jl and AgentsX.jl, as well as the design principles of the Julia language, allows modular improvements resulting in ambitious possibilities for development and future work. These may include a centralised representation of agents that can be converted to human readable documentation as well as computer-readable code with minimal effort. Another direction relates to perception and actuation interfaces for interactions. This is based on the idea that the actual environmental conditions are perceived by the agents with an individual bias, resulting in variations of perception. Similarly, actions taken by the agents will have an intention-actuation gap, resulting in variations of actuation.

We note that in the most recent Agents.jl publication [15] it is suggested that the multi-agent simulation architecture may be upheaved in the future. We believe the work presented and the proposed future works can positively influence the modelling of complex social behaviours within urban and social systems.

References

1. Bonabeau, E.: Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences 99(suppl_3), 7280–7287 (5 2002), https://pnas.org/doi/full/10.1073/pnas.082080899
2. Auchincloss, A.H., Diez Roux, A.V.: A new tool for epidemiology: The usefulness of dynamic-agent models in understanding place effects on health (7 2008), https://pubmed.ncbi.nlm.nih.gov/18480064/
3. Thompson, J., Wijnands, J.S., Savino, G., Lawrence, B., Stevenson, M.: Estimating the safety benefit of separated cycling infrastructure adjusted for behavioral adaptation among drivers; an application of agent-based modelling. Transportation Research Part F: Traffic Psychology and Behaviour 49, 18–28 (8 2017)

4. Zou, Y., Torrens, P.M., Ghanem, R.G., Kevrekidis, I.G.: Accelerating agent-based computation of complex urban systems. International Journal of Geographical Information Science 26(10), 1917–1937 (10 2012), https://www.tandfonline.com/doi/abs/10.1080/13658816.2012.669891
5. Widener, M.J., Metcalf, S.S., Bar-Yam, Y.: Agent-based modeling of policies to improve urban food access for low-income populations. Applied Geography 40, 1–10 (6 2013)
6. Guzy, M.R., Smith, C.L., Bolte, J.P., Hulse, D.W., Gregory, S.V.: Policy Research Using Agent-Based Modeling to Assess Future Impacts of Urban Expansion into Farmlands and Forests. Ecology and Society 13(1), art37 (2008), http://www.ecologyandsociety.org/vol13/iss1/art37/
7. Jokar Arsanjani, J., Helbich, M., de Noronha Vaz, E.: Spatiotemporal simulation of urban growth patterns using agent-based modeling: The case of Tehran. Cities 32, 33–42 (6 2013)
8. Huang, Q., Parker, D.C., Filatova, T., Sun, S.: A review of urban residential choice models using agent-based modeling. Environ. Plann. B Plann. Des. 41(4), 661–689 (8 2014), https://journals.sagepub.com/doi/abs/10.1068/b120043p
9. Tisue, S., Wilensky, U.: Netlogo: A simple environment for modeling complexity. Conference on Complex Systems pp. 1–10 (2004), http://ccl.sesp.northwestern.edu/papers/netlogo-iccs2004.pdf
10. Iba, H.: Agent-Based Modeling and Simulation with Swarm. Chapman and Hall/CRC (6 2013), https://www.taylorfrancis.com/books/9781466562400
11. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: MASON: A Multiagent Simulation Environment. Simulation 81(7), 517–527 (2005)
12. North, M.J., Collier, N.T., Ozik, J., Tatara, E.R., Macal, C.M., Bragen, M., Sydelko, P.: Complex adaptive systems modeling with Repast Simphony. Complex Adaptive Systems Modeling 1(1), 3 (12 2013), https://link.springer.com/10.1186/2194-3206-1-3
13. Masad, D., Kazil, J.: Mesa: An Agent-Based Modeling Framework. In: Proceedings of the 14th Python in Science Conference. pp. 51–58. SciPy (2015)
14. Vahdati, A.: Agents.jl: agent-based modeling framework in Julia. Journal of Open Source Software 4(42), 1611 (10 2019), https://joss.theoj.org/papers/10.21105/joss.01611
15. Datseris, G., Vahdati, A.R., DuBois, T.C.: Agents.jl: a performant and feature-full agent-based modeling software of minimal code complexity. SIMULATION p. 003754972110688 (1 2022), http://journals.sagepub.com/doi/10.1177/00375497211068820
16. Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S.K., Huse, G., Huth, A., Jepsen, J.U., Jørgensen, C., Mooij, W.M., Müller, B., Pe'er, G., Piou, C., Railsback, S.F., Robbins, A.M., Robbins, M.M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R.A., Vabø, R., Visser, U., DeAngelis, D.L.: A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198(1-2), 115–126 (9 2006)
17. Reeves, J.: Code as Design: Three Essays by Jack W. Developer.* pp. 1–24 (2005)
18. Meisser, L.: The Code is the Model. International Journal of Microsimulation 10(3), 184–201 (12 2016), http://www.microsimulation.org/IJM/V10_3/IJM_2017_10_3_6.pdf
19. Kehoe, J.: Creating reproducible agent based models using formal methods. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial

Intelligence and Lecture Notes in Bioinformatics). vol. 10399 LNAI, pp. 42–70 (2017)
20. Luck, M., D'Inverno, M.: Formal Methods and Agent-Based Systems. In: Agent Technology from a Formal Perspective, pp. 65–96 (2006)
21. Fisher, M.M., Wooldridge, M.J.: Towards Formal Methods for Agent-Based Systems. In: Electronic Workshops in Computing. pp. 23–24 (1996)