IDENTIFYING THE BUILDING BLOCKS OF SOCIAL SIMULATION MODELS: A QUALITATIVE ANALYSIS USING OPEN-SOURCE CODES IN NETLOGO

Shannon Cheng
Philippe J. Giabbanelli

Department of Computer Science & Software Engineering
Miami University
205W Benton Hall
Oxford, OH, USA
{chengs12, giabbapj}@miamioh.edu

Zaiyi Kuang

The Summit Country Day School

2161 Grandin Rd
Cincinnati, OH, USA
derekkuang0213@gmail.com

ABSTRACT

Repeatedly developing models from scratch raises the possibility of code bugs and may be an ineffective use of resources. While reusable building blocks or component libraries are available in multiple simulation software and commonly used in industrial engineering, they are often absent from social simulation models or only support data-centric tasks (e.g., import/export maps). We examined social simulation models for human migration (which emphasize the geographical aspect) and rumor spread (which focuses on information processing) to establish whether building blocks could be created across application domains. Based on 39 NetLogo models collected from two platforms (Github and CoMSES), our quantitative analysis with 11 software metrics confirmed the possibility of using building blocks given the current code complexity and time investment of modelers. Our qualitative thematic analysis found five themes in each application domain, of which three were shared and can be a priority to develop reusable building blocks.

Keywords: Building Blocks, Model Components, Mixed Methods, NetLogo, Reusability.

1 INTRODUCTION

Building blocks or 'component libraries' are now ubiquitous across modeling and simulation software. For example, `AnyLogic` provides a Material Handling Library so that users can quickly build a model involving routing strategies in production and storage facilities, instead of having to (re)define entities such as robots and operators or the notion of a workflow (Lebedev and Churkov 2018). These libraries have long been part of simulation software. For example, Sturrock and Pegden (2011) explained how the standard library in `Simio` provided objects with customizable properties and processes, such as a pre-defined vehicle with the notion of a route. Relevant components can also be automatically identified and arranged when generating a model from a blueprint, for instance by creating a manufacturing model from either a shop-floor layout (Pinto, Aguiar, and Gonçalves 2020) or high-level specifications of a production plant (Bär, Zeilmann, Nophut, Kleinert, Beyer, and Voigt 2021). These techniques have become increasingly sophisticated thanks to advances in Artificial Intelligence, hence it is now possible to extract process data from domain-relevant documents (e.g., via text mining and computer vision) and build a model, as exemplified for digital twins or electric vehicles (Azangoo et al. 2022, Davis et al. 2022). In contrast to the established practice of leveraging building blocks for simulations in industrial engineering (often through discrete event systems), there is a paucity of work on building blocks for social simulations (which frequently involve agent-based models).

Automation in the context of social simulations tends to focus on model calibration (D'Auria et al. 2020) and testing (Clark, Walkinshaw, and Hierons 2021) or synthetic population generation from data using machine learning rather than by orchestrating transparent blocks (Rao and Chernyakhovsky 2019, Bernard et al. 2023, Kotnana et al. 2022). In this paper, we examine the potential for developing building blocks in social simulations, which can guide short-term efforts in creating libraries and contribute to long-term needs for automation. This potential started to be investigated by Vendome, Rao, and Giabbanelli (2020), whose work at the Spring Simulation conference showed that artificial societies coded in `NetLogo` were often developed from scratch rather than through building blocks. When developers used libraries, it was primarily for data handling (e.g., import/export of CSV files or geographical data) rather than to build rules in a model. In a follow-up study focused on the *potential* for developing building blocks that support COVID-19 models in `NetLogo`, two observations were made from a thematic analysis (Schroeder et al. 2022). First, the level of sophistication of the models show that programmers have a sufficient skillset to engage in the identification and appropriate reuse of component libraries. Second, similarities between the design of models suggested that many of them could be significantly simplified by reusing components (hence saving development time and potentially avoiding bugs), however there is currently no library that provides the required components. Developing a library for each disease to provide reusable components may not be a sustainable approach, hence a broader evidence base is needed to guide efforts in software engineering and simulation. The main goal of this paper is to broaden the evidence base by identifying potential building blocks in social simulations originating from other application areas than COVID-19. Since the identification of blocks is currently a manual effort achieved by thematic analysis (i.e., reading codes several times and looking for similarities), we selected two domains that complement the information obtained in the previous work of Schroeder et al. (2022):

- *Human migrations*, which link human decision-making and places (involving e.g. resources, routes, spatial coordination), thus bringing in a useful geographical component in social simulations.
- *The spread of rumors*, to complement the prior investigation of disease spread models by emphasizing how individuals process information, which is an important element of artificial societies.

The remainder of this paper is structured as follows. In Section 2, we briefly introduce the two application domains and seminal modeling works in these domains, then we summarize how previous works have analyzed code using both quantitative (i.e., automatic software metrics from a parser) and qualitative (i.e., human readers) evaluations. These two subsections can be read independently and are intended to keep this paper self-contained. In Section 3, we explain our methods, including how we collect 39 `NetLogo` models and analyzed them in line with prior works. We present our results in Section 4 and discuss them in Section 5 with respect to our central research question of identifying building blocks across different types of social simulations to guide the development of future libraries.

2 BACKGROUND

2.1 Modeling Context: Human Migrations and Rumor Spread

Although the history of human migration is as long as the history of mankind, human migration is a particular relevant topic today. An estimated 271 million people live in a country other than their countries of birth as of 2020 (United Nations 2020), 118 million more than in 1990 and over three times the estimated number in 1970. Even after taking into account the population increase, migration has still increased by a considerable amount. Conflict, climate change, and weather-related hazards all have played a big part in this increase. For example, the recent Russian invasion in Ukraine has caused millions to flee their home country (Lloyd and Sirkeci 2022). Additionally, because of climate change, millions of people and entire nations are expected to be displaced in the future due to crop failure, water scarcity, and sea-level rise. This is one of the defining topics of the 21st century in Pacific Island Countries, North Africa, and South Asia (Malji, Obana, and Hopkins 2022, Campbell and Bedford 2022, Daoudy, Sowers, and Weinthal 2022). Understanding and predicting patterns of human migration is thus a current challenge so that governments and organizations can avert human crises.

The relation between individuals and places is at the heart of migration models. In the early model by Schelling (1971), the space was discretized and individuals would move (i.e., 'teleport' themselves) to a free location if they were not satisfied with their current surroundings. In this model of *voluntary* relocation, the journey was unimportant and agents could elect to remain in their current position. These two features stand in sharp contrast with models for *forced migrations* (Frydenlund 2021). For example, a decision to move may depend on weather conditions, both to prompt a relocation (e.g., water scarcity) and to determine a safe and accessible travel path. Consequently, models can take a coupled approach by combining weather data (e.g., precipitation at the micro-scale) and routes at the micro-scale with regional destinations at the macro-scale (Jahani et al. 2021). On the one hand, these various models differ vastly in their features, as some may focus on weather and paths (e.g., walking, river crossing) while others examine the management of refugee camps or armed conflicts. These differences exist because "the majority of ABMs of migration are built with a concrete real-world scenario in mind, often with a specific focus on one aspect of the situation" (Hinsch and Bijak 2022). Despite these differences, these models share an interest in geography since "migration is an inherently spatial process" and human decision-making activities shape whether people will leave and where they will go (Hinsch and Bijak 2022). The trade-off between the differences originating from purpose-built models and the similarities in terms of mechanisms is an essential theme in this paper, as we seek to find building blocks usable across models.

The spread of rumors (i.e., unconfirmed stories and anecdotes) is as ancient a phenomenon as migrations. Rumors have long been investigated by psychologists interested in the transmission process, sociologists studying the impact of social structure, and historians examining how rumors shaped major events (Pendleton 1998). The growth of the internet, network science and machine learning have brought a new dimension to the study of rumors (Moreno, Nekovee, and Pacheco 2004, Meel and Vishwakarma 2020) and enabled the creation of models that could be compared against large datasets (Ndii et al. 2018, Patel et al. 2017). For example, numerous Agent Based Models have been created to study rumor spread in Twitter and address problems such as rumor detection or countermeasures (Serrano, Iglesias, and Garijo 2015). These models have also gained visibility given the growing interest in 'fake news' (Gausen, Luk, and Guo 2021) and the recent anti-vaccination movements regarding COVID-19 (Sobkowicz and Sobkowicz 2021). Models differ depending on the research question, medium of interest (e.g., social media, word of mouth), or cognitive sophistication (e.g., forgetting and remembering mechanisms). However, they also share features such as the use of complex networks (e.g., small-world, scale-free) to model interactions or the categorization of agents (e.g., ignores the rumor, knows the rumor, no longer spreads it).

Several modeling techniques and an abundance of software allow modelers to implement their design. Unlike large-scale simulations on high performance computing clusters which need to optimize performances and often operate in a parallel and distributed setting, `NetLogo` models are deploying on a local machine and run on its (limited) resources (Ferdousi et al. 2022). In exchange for limited scalability, the framework is easy to learn and use, allowing modelers to implement their ideas relatively quickly. This software thus provides a fertile ground to study the potential for building blocks that would further simplify the process of model implementation. Agent-Based Models implemented in `NetLogo` are commonplace both for migrations and rumor spread, where they have been used for over a decade and continue to be developed. Liu and Chen 2011 proposed a `NetLogo` rumor spread in 2011, focusing on transmissions in a scale-free network. Kaligotla et al. 2015 used the same platform at the 2015 Winter Simulation conference when studying competing rumors and continued to use it when comparing three models in their 2022 study (Kaligotla et al. 2022). Other recent studies include Hosseini and Zandvakili (2022), who added fuzzy logic to rumor spread, or Pitman et al. (2022), whose mixed-methods study focused on countering rumor spreads. A plethora of studies using `NetLogo` can also be found for human migrations, from simulations of the dispersions of early settlements in the Iron Age (Olševičová et al. 2015) to investigations of contemporary migrations brought by climate change (Lin, Carley, and Cheng 2016, Magallanes, Burger, and Cioffi-Revilla 2014) or wars (Hébert et al. 2018).

2.2 Quantitative and Qualitative Analyses of Simulation Models

There is a vast literature on quantifying some aspects of a software, depending on the *purpose* (e.g., complexity, readability, usability). Metrics are further distinguished as *static* (i.e., obtained by analyzing the code) or *dynamic* (i.e., obtained from data generated during execution of the software). For example, 'size' from a static perspective can refer to the average number of lines per function, whereas from a dynamic perspective it could be the number of classes loaded at runtime or the number of instructions responsible for most of the execution. Analyzing software developed in a general-purpose language such as Java or C++ is a common task supported by many tools (Bhatia et al. 2022). In contrast, the analysis of *models* developed in a domain-specific programming language (e.g., `NetLogo` for agent-based modeling) has received little attention and there is no tool (whether public or commercial) that can readily be used to produce metrics. Previous authors thus wrote their own *parsers* (Schroeder et al. 2022, Vendome et al. 2020) to separate `NetLogo` codes into three parts for static analysis: the *comments* (e.g., number of comment lines and average number of words), the Graphical User Interface or *GUI* (e.g., number of visual elements such as buttons and sliders), and the code itself (e.g., number of conditionals, average length of a function). These metrics provide information for various purposes: GUI metrics are related to usability (Sagar and Saha 2017), comments are indicative of readability (Fakhoury et al. 2019), and the number of functions are associated with complexity (Antinyan et al. 2015).

While quantitative software metrics are automatically extracted, *qualitative* analyses are performed by human readers using methods such as thematic analyses. The process starts by reading documents to become familiar with the corpus and identify potential patterns, then formalizing these patterns into tentative themes (known as 'codes') which may be combined for efficiency, and applying them to the data. Thematic analyses are common for text documents such as interviews or research article; for example, modeling and simulation papers can be arranged by research fields (Mustafee et al. 2018) while individual researchers may be assigned themes based on their publications (Mustafee and Fishwick 2017). Although thematic analyses also happen for comments in relation to software code, it is less commonly applied to the code itself, perhaps due to the inherent difficulty of reading another developer's code and accurately capturing its intent. Previous authors performed thematic analyses of `NetLogo` codes to identify patterns that could be replaced by building blocks (Schroeder et al. 2022), although solely in the context of COVID-19 models.

3 METHODS

We gathered models from two hosting platforms: Github.com and CoMSES (Table 3). As of January 2023, Github has over 371million repositories, of which about 10% are publicly accessible (Github 2023). CoMSES is a leading platform in the modeling community for Agents Based Models (ABM), consisting of a library of 1,026 models as of January 2023. While Github's popularity allows users to find a multitude of models, it covers many different languages, of which `NetLogo` ABM models constitute a small portion. Hence, we filtered out the repositories that (i) contained at least one file with the `NetLogo` extension with the command 'language:NetLogo', and (ii) contained the keyword 'turtle' with the command 'turtles:infile',

Table 1: Number of models gathered at each step of data collection, for each source and topic.

	GitHub (>371M repos, ≈ 30M public)		CoMSES (>1,000 public models)		Total
Topic	Keyword	Filtered	Keyword	Filtered	
Human migration	1,539	15	61	7	22
Rumor spread	4,113	10	53	6	17

which in `NetLogo` signifies the creation of an agent in an ABM. From there, we used keywords to obtain the models of interest. For human migrations, the keywords were 'Migration', 'Refugees', 'Population Movement', and 'Relocation', which returned a total of 1539 repositories. Many of these results, however, did not pertain to the subject of *human* migration, hence we manually investigated a sample of the models to determine additional criteria: (i) must include the keyword 'human', and (ii) further filtered with other keywords that constantly appeared in extraneous models ('mammoth', 'mosquito', 'zombie', and 'final'). These additional criteria filtered the initial 107 repositories down to a final set of 15 for human migrations. The initial stage of filtering for ABMs was omitted in CoMSES, since the platform is exclusively intended for ABMs, many of which use `NetLogo`. The initial 61 CoMSES repositories on migrations were filtered down to 7 after after applying the additional criteria. We applied the same process for rumor spread model by using 'spread', 'rumor' and 'fake news' as search keywords and then filtering out irrelevant models, which reduced the samples from 4,113 (Github) and 53 (CoMSES) to 10 and 6, respectively.

We performed quantitative and qualitative analyses as described in section 2.2. We measured 11 quantitative metrics covered two categories of code characteristics: complexity (2 metrics; number of functions and conditional statements) and readability (9 metrics). We emphasized widely accepted readability metrics (Fakhoury et al. 2019) that are applicable to code in general, including the number of lines with and without comments (i.e., code lines), number of empty lines, number of loops, number of nested loops, the average line length and function size, and the number of quality identifiers. In addition, we counted the number of 'breeds', which signifies the types of agents within `NetLogo`. We created a *custom parser* to automatically extract eleven metrics and we manually counted the number of breeds, since a brief inspection at the top of the code was sufficient. The parser isolates code-related metrics by parsing a `NetLogo` file until reaching the sequence '@#$#@#$#@' which signals the end of the code segment. Within the code portion, we identify comments as all content following the ';' character. Comments are separated from the code, to avoid mistakenly counting a comment as a loop or conditional. Conditionals are measured by the number of 'if' and 'ifelse' statements. The statements 'ask', 'while', and 'repeat' signaled the presence of a loop. When counting a nested loop, we only counted loops that are directly within the scope of another loop. That is, if a loop appears inside a loop which is itself inside a loop, we only count it as one nested loop and disregard deeper loops. Identifiers were found through the 'let' keyword and had to be at least four letters long to be deemed sufficient (e.g., 'i' or 'j' is not considered a 'quality' identifier for software readability). The quantitative data was then statistically summarized via the mean, standard deviation, minimum and maximum. We also performed a correlation analysis to identify pairwise relations between metrics.

To perform the qualitative thematic analysis, we annotated the code in an online collaborative environment by transferring each `NetLogo` file to Google Documents. Since the analysis aims at (i) decomposing each code into its functional blocks and (ii) identify shared blocks across files, we familiarized ourselves with the model by running each whenever possible. In some cases, models had datasets that were unavailable, hence we solely relied on the code to understand how the model functioned. Then, when reading through the code, segments of interest were highlighted. These segments were later grouped into themes that were created and evolved as we read through the code. Figure 1 exemplifies how the content of two models was annotated with respect to the final themes. As summarized in section 2.1, the use of `NetLogo` as a unifying platform allowed us to apply the same analytical methods to the models of human migrations and rumor spread, but we kept the themes distinct across these two application domains.

Figure 1: Example of thematic annotation with themes and our notes on the left, and the annotated `NetLogo` code in the middle and right panels. In this model, all themes are present hence all five colors are used.

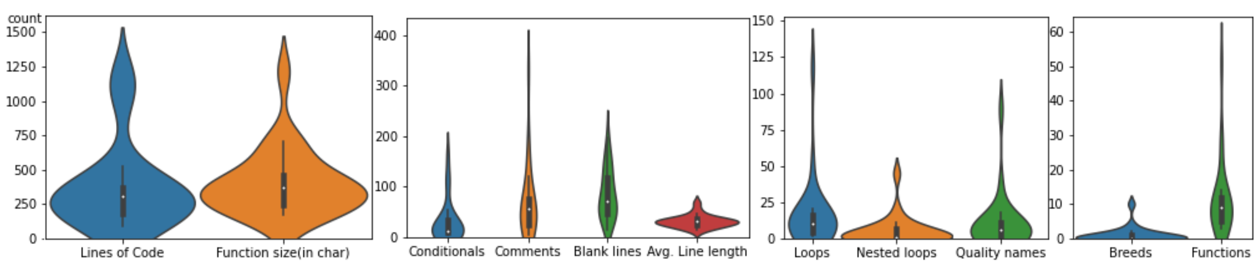

Figure 2: Distributions for the eleven quantitative software metrics, examplified for human migration models. Note that vertical (y-axis) scales differ and they are sorted from largest (left) to smallest (right).

4 RESULTS

For full disclosure, all annotated models (for the thematic analysis) and their software metrics are provided as supplementary online material on permanent hosting at https://doi.org/10.5281/zenodo.7567048.

Our eleven *quantitative* software metrics are summarized in Table 3 for the two application domains. Note that some of the metrics are correlated (Table 3), hence we do not discuss every one of them. For example, the number of comments scales with the size of the code or the complexity of the procedure (e.g., more nested loops come with more comments), and the number of blank lines is proportional to comment lines since they serve as separators. To contextualize these numbers and ensure the correctness of our parser, previous research has shown that metrics are *generally* higher for models created by professionals and in support of peer-review articles (e.g., on the CoMSES platform), lower for GitHub models, and at their lowest values for sample models provided with the `NetLogo` library (Vendome, Rao, and Giabbanelli 2020). Since our study examines two *specific* domains, our metrics also reflect the needs of modelers in these domains. For example, modelers rarely use 'breeds' (i.e., types of agents as shown in Figure 3-a); a closer inspection of

Table 2: Eleven quantitative software metrics applied across models each application field.

	Human Migrations			**Rumor spread**		
Metric	*Mean* ± *std*	*Min*	*Max*	*Mean* ± *std*	*Min*	*Max*
# Conditionals	33.1 ± 42.1	4.0	163.0	20.9 ± 10.7	7.0	38.0
# Functions	11.7 ± 11.0	3.0	51.0	15.5 ± 4.2	9.0	23.0
Avg. Function size	407.1 ± 239.6	173.2	1214.3	268.6 ± 198.0	113.9	866.1
Avg. Line length	32.5 ± 12.4	13.7	69.0	27.8 ± 11.1	15.3	63.0
# Lines of code	1298.0 ± 580.0	686.0	2686.0	369.9 ± 188.1	125.0	762.0
# Comment lines	72.8 ± 74.8	4.0	330.0	64.2 ± 45.3	13.0	185.0
# Blank lines	86.0 ± 50.0	16.0	198.0	91.1 ± 79.2	23.0	349.0
# Breeds (i.e., agent types)	1.0 ± 2.3	0.0	10.0	1.1 ± 1.5	0.0	5.0
# Loops	17.1 ± 24.8	2.0	118.0	23.0 ± 14.1	6.0	64.0
# Nested loops	5.6 ± 10.1	0.0	45.0	6.8 ± 6.5	0.0	19.0
# Quality identifiers	10.6 ± 19.2	0.0	89.0	9.8 ± 9.3	0.0	35.0

Table 3: Spearman correlation between every pair of metric (shown for human migration models). Values are symmetric along the diagonal, so they can be read from either direction for convenience.

Metrics	Comm–ents	Line length	Function size	Blank lines	Loops	Nested loops	Quality Identifiers	Lines of Code	Breeds
Comments		.281	.351	.643	.860	.866	.400	.906	-.112
Line length	.281		.175	.071	.178	.248	.093	.228	.315
Function size	.351	.175		.328	.238	.287	.030	.367	.090
Blank lines	.643	.071	.328		.504	.412	.560	.816	-.208
Loops	.860	.178	.238	.504		.958	.286	.693	-.106
Nested loops	.866	.248	.287	.412	.958		.304	.689	-.137
Identifiers	.400	.093	.030	.560	.286	.304		.567	-.207
Lines of Code	.906	.228	.367	.816	.693	.689	.567		-.122
Breeds	-.112	.315	.090	-.208	-.106	-.137	-.207	-.122	

model codes reveal that a common practice is to distinguish agents by colors, thus using graphical properties of the objects to categorize them instead of using breeds (Figure 3-b). Other numbers reflect that our models are a mixture of GitHub and CoMSES. For example, we have 86 to 91 blank lines, while general GitHub models have 77 and CoMSES models have 114 (Vendome, Rao, and Giabbanelli 2020). Our models also produce similar metrics to a past study on COVID-19 models (Schroeder et al. 2022). For instance, we have 17 to 23 loops on average, while COVID-19 models had 18 on average. In summary, the numbers produced by our parser align with general expectations.

The models had over 10 functions, several hundred lines of code, numerous loops and dozens of conditionals. These numbers are much higher than the sample models provided in the `NetLogo` library, hence they suggest that implementing models of such complexity must have required some time investment and programming skills. This is an important observation because it shows the *potential for using building blocks*. If the models were already very short, then there would be no need to replace large portions of code by calling reusable blocks. If the models were very simple, then programmers may lack the skills to successfully integrate reusable blocks into their code base.

Our *qualitative* analysis revealed four themes for human migration models: agent heterogeneity, area heterogeneity, agent interactions, and movements (which we further split into the sub-themes of specified movement and union movement). *Movement* is the most important theme and it is present in all but two models (Figure 3), as expected given that the application domain of migration is defined as 'the movement of people to a new area or country'. *Agent heterogeneity* is the second most important theme and serves to capture that places have different types of people, who will express various reactions under the same circumstances. *Area heterogeneity* is the third term, which reflects that people migrate to a different area because it has some characteristics (e.g., resources) that their place or origin may have lacked. As shown in Figure 3, models often chose between representative spatial or human heterogeneity. The final theme of *interactions between agents* is only used when such interactions play a role in the migration decision, which is not always the case. The differences between the extent to which each model expressed certain theme can be partly explained by their focus, as some examined migrations that were voluntary while others considered populations that were displaced (i.e., refugees). Many of the models based on voluntary relocation were similar to Schelling (1971), where areas are not inherently heterogeneous but instead differentiated by the characteristics of their residents. Consequently, these models have low spatial heterogeneity and instead primarily operate through human heterogeneity. In contrast, refugee models are all characterized by area heterogeneity (e.g., moving to an environment without war) and agents do not fully get to decide where they are moved hence decision-making processes and interactions are limited.

Although human migration and rumor spread are vastly different application domains, our qualitative analysis found that they shared three out of five themes. Indeed, rumor spread model also dealt with *agent heterogeneity*, which can take many lines of code as agents are assigned various characteristics. These models also dealt with *movements* by including conditions for agents' movements, direction and speed; this was perhaps unexpected as a rumor spread is not necessarily a spatial phenomenon and research often emphasizes the role of social networks. *Interactions* were essential, as a rumor can only spread when agents are in contact. The last two themes were unique to rumor spread: *reaction to the environment* and *opinion change*.

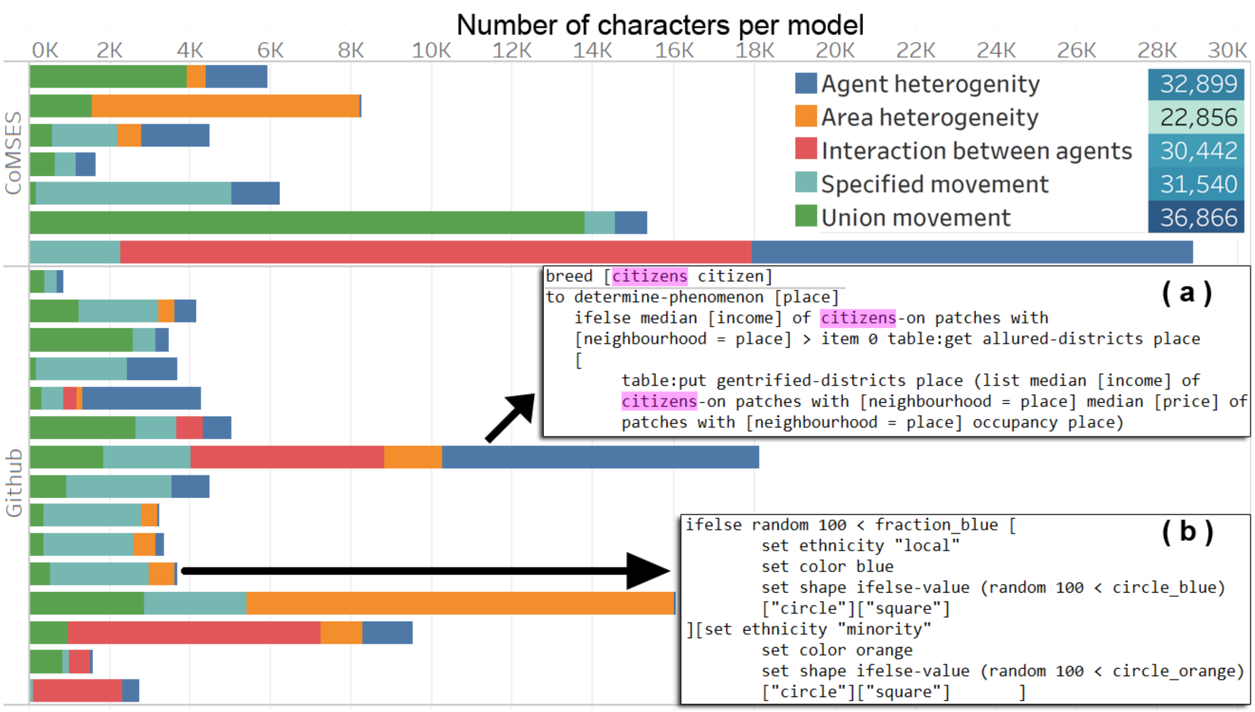

Figure 3: Number of *characters* devoted to each theme within models of human migration, broken down by platform (CoMSES, Github). Code excerpts show the use of breeds (a) or common work-arounds (b).

5 DISCUSSION

Building blocks are familiar constructs in industrial engineering (Lebedev and Churkov 2018, Sturrock and Pegden 2011), as they allow to "develop and use advanced features within a few minutes, which would otherwise have required a much larger modeling effort and that would have been error-prone" (Fezans and Koloschin 2022). In *social* simulations, building blocks are seldom used (Vendome, Rao, and Giabbanelli 2020). We recently showed the *potential* for building blocks in COVID-19 given similarities between models, but the needs of a single application area were insufficient to identify blocks that should be developed to support social simulations across domains. In this paper, we investigated two additional domains, thus complementing the emphasis of Schroeder et al. (2022) on epidemiological simulations with geographical (i.e., human migration) and information-centric phenomena (i.e., rumor spread). Based on eleven quantitative software metrics, we confirmed that the code for social simulations in `NetLogo` was sufficiently complex to indicate potential room for building blocks and the necessary skills for their integration. Most importantly, we found that models across the two application domains shared three themes (agent heterogeneity, agent movement, agent interactions), of which two were also found in COVID-19 models (heterogeneity and movement). These results suggest that the development of libraries to create heterogeneous agents and orchestrate their movements should be a priority to support social simulations going forward.

Generators are available to create synthetic mobility data (Berke et al. 2022) as well as synthetic populations (Zhou et al. 2022), hence making them available as libraries could be a useful starting point. However, it would not suffice to support all forms of agent heterogeneity. Indeed, numerous methods to create heterogeneous agents focus on variation in individual characteristics, such as age, sex, or income. However, agents are also heterogeneous because they *think* differently hence there is a need to generate different behavioral rules. Although there are algorithms to create agents who think in unique ways (Bernard et al. 2023), these algorithms are sufficiently different from other approaches to synthetic population generation that they ought to be packaged in different libraries. We note that these libraries do not have to be developed solely for `NetLogo`, as this environment is now able to also run code in Python and R (Sulis and Taveter 2022), which also support agent-based modeling (e.g., via `Mesa`).

Creating libraries would not *suffice* to change current practices. First, these libraries should be discovered by their intended users. For instance, building blocks may capture the core mechanisms of social behaviors at different *levels*, hence potential users should be matched with a library that has the required functionality at the desired level. Second, the building blocks must be correctly integrated. `NetLogo` users could theoretically access a library developed in Python by going through a NetLogo-Python interface, but that may create a barrier if the developers have limited programming skills. In addition, the building blocks offered by libraries may need to be fine-tuned in order to ensure that the overall behavior of the model is adequate (Voinov and Shugart 2013). Our research in identifying key building blocks is thus only the beginning of a longer process for our scientific community including development, identification, and integration.

REFERENCES

Antinyan, V., M. Staron, J. Derehag, M. Runsten, E. Wikström, W. Meding, A. Henriksson, and J. Hansson. 2015. "Identifying complex functions: By investigating various aspects of code complexity". In *2015 Science and Information Conf.*, pp. 879–888. Institute of Electrical and Electronics Engineers, Inc.

Azangoo, M., L. Sorsamäki, S. A. Sierla, T. Mätäsniemi, M. Rantala, K. Rainio, and V. Vyatkin. 2022. "A methodology for generating a digital twin for process industry: a case study of a fiber processing pilot plant". *IEEE Access* vol. 10, pp. 58787–58810.

Bär, R. M., M. Zeilmann, C. Nophut, J. Kleinert, K. Beyer, and T. Voigt. 2021. "Simulation of Energy and Media Demand of Beverage Bottling Plants by Automatic Model Generation". *Sustainability* vol. 13 (18), pp. 10089.

Berke, A., R. Doorley, K. Larson, and E. Moro. 2022. “Generating synthetic mobility data for a realistic population with RNNs to improve utility and privacy”. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 964–967.

Bernard, D., S. Cussat-Blanc, and P. Giabbanelli. 2023. “Fast Generation of Heterogeneous Mental Models from Longitudinal Data by Combining Genetic Algorithms and Fuzzy Cognitive Maps”. In *Proceedings of the 56th Hawaii International Conference on System Sciences*, pp. 1570–1579.

Bhatia, P. K. et al. 2022. “A Survey of Static and Dynamic Metrics Tools for Object Oriented Environment”. In *Emerging Research in Computing, Information, Communication and Applications*, pp. 521–530. Springer.

Campbell, J. R., and R. D. Bedford. 2022. “Climate change and migration: Lessons from Oceania”. In *Routledge Handbook of Immigration and Refugee Studies*, pp. 379–387. Routledge.

Clark, A. G., N. Walkinshaw, and R. M. Hierons. 2021. “Test case generation for agent-based models: A systematic literature review”. *Information and Software Technology* vol. 135, pp. 106567.

Daoudy, M., J. Sowers, and E. Weinthal. 2022. “What is climate security? Framing risks around water, food, and migration in the Middle East and North Africa”. *Wiley Interdisciplinary Reviews: Water* vol. 9 (3), pp. e1582.

D’Auria, M., E. O. Scott, R. S. Lather, J. Hilty, and S. Luke. 2020. “Distributed, automated calibration of agent-based model parameters and agent behaviors”. *Autonomous Agents and Multiagent Systems (AAMAS)*.

Davis, C. W., A. J. Jetter, and P. J. Giabbanelli. 2022. “Automatically Generating Scenarios from a Text Corpus: A Case Study on Electric Vehicles”. *Sustainability* vol. 14 (13), pp. 7938.

Fakhoury, S., D. Roy, A. Hassan, and V. Arnaoudova. 2019. “Improving source code readability: Theory and practice”. In *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*, pp. 2–12. Institute of Electrical and Electronics Engineers, Inc.

Ferdousi, T., A. Kishore, L. Machi, D. Machi, C. J. Kuhlman, and S. Ravi. 2022. “A Web-Based System for Contagion Simulations on Networked Populations”. In *2022 IEEE 18th International Conference on e-Science (e-Science)*, pp. 306–315. Institute of Electrical and Electronics Engineers, Inc.

Fezans, N., and A. Koloschin. 2022. “Extending Simulink with Blocks Managing Complex Objects Collaboratively–Application to Air-to-Air Refueling Simulation”. In *AIAA SCITECH 2022 Forum*, pp. 0808.

Frydenlund, E. 2021. “Modeling and Simulation as a Bridge to Advance Practical and Theoretical Insights About Forced Migration Studies”. *Journal on Migration and Human Security* vol. 9 (3), pp. 165–181.

Gausen, A., W. Luk, and C. Guo. 2021. “Can We Stop Fake News? Using Agent-Based Modelling to Evaluate Countermeasures for Misinformation on Social Media”. In *Workshop Proceedings of the 15th International AAAI Conference on Web and Social Media*.

Github 2023. “Github Repository Results”.

Hébert, G. A., L. Perez, and S. Harati. 2018. “An agent-based model to identify migration pathways of refugees: the case of Syria”. In *Agent-based models and complexity science in the age of geospatial big data*, pp. 45–58. Springer.

Hinsch, M., and J. Bijak. 2022. “Principles and State of the Art of Agent-Based Migration Modelling”. *Towards Bayesian Model-Based Demography*, pp. 33–49.

Hosseini, S., and A. Zandvakili. 2022. “Information dissemination modeling based on rumor propagation in online social networks with fuzzy logic”. *Social Network Analysis and Mining* vol. 12 (1), pp. 1–18.

Jahani, A., H. Arabnejad, D. Suleimanova, M. Vuckovic, I. Mahmood, and D. Groen. 2021. "Towards a Coupled Migration and Weather Simulation: South Sudan Conflict". In *International Conference on Computational Science*, pp. 502–515. Springer.

Kaligotla, C., E. Yücesan, and S. E. Chick. 2015. "An agent based model of spread of competing rumors through online interactions on social media". In *2015 winter simulation conference (WSC)*, pp. 3985–3996. Institute of Electrical and Electronics Engineers, Inc.

Kaligotla, C., E. Yücesan, and S. E. Chick. 2022. "Diffusion of competing rumours on social media". *Journal of Simulation* vol. 16 (3), pp. 230–250.

Kotnana, S., D. Han, T. Anderson, A. Züfle, and H. Kavak. 2022. "Using Generative Adversarial Networks to Assist Synthetic Population Creation for Simulations". In *2022 Annual Modeling and Simulation Conference (ANNSIM)*, pp. 1–12. Institute of Electrical and Electronics Engineers, Inc.

Lebedev, P., and N. Churkov. 2018. "New anylogic material handling library". In *Proceedings of the 2018 Winter Simulation Conference*, pp. 4244–4244.

Lin, L., K. M. Carley, and S.-F. Cheng. 2016. "An agent-based approach to human migration movement". In *2016 Winter Simulation Conference (WSC)*, pp. 3510–3520. Institute of Electrical and Electronics Engineers, Inc.

Liu, D., and X. Chen. 2011. "Rumor propagation in online social networks like twitter–a simulation study". In *2011 Third International Conference on Multimedia Information Networking and Security*, pp. 278–282. Institute of Electrical and Electronics Engineers, Inc.

Lloyd, A. T., and I. Sirkeci. 2022. "A Long-Term View of Refugee Flows from Ukraine: War, Insecurities, and Migration". *Migration Letters* vol. 19 (4), pp. 523–535.

Magallanes, J. M., A. Burger, and C. Cioffi-Revilla. 2014. "Understanding migration induced by climate change in the Central Andes of Peru via agent-based computational modeling". In *Proceedings of the 5th. World Congress on Social Simulation*, pp. 271–283.

Malji, A., L. Obana, and C. Hopkins. 2022. "When Home Disappears: South Asia and the Growing Risk of Climate Conflict". *Terrorism and Political Violence*, pp. 1–19.

Meel, P., and D. K. Vishwakarma. 2020. "Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities". *Expert Systems with Applications* vol. 153, pp. 112986.

Moreno, Y., M. Nekovee, and A. F. Pacheco. 2004. "Dynamics of rumor spreading in complex networks". *Physical review E* vol. 69 (6), pp. 066130.

Mustafee, N., and P. Fishwick. 2017. "Analysis of M&S Literature Published in the Proceeding of the Winter Simulation Conference from 1981 to 2016". In *Advances in Modeling and Simulation*, pp. 333–351. Springer.

Mustafee, N., K. Katsaliaki, T. Le Forsonney, and S. Diallio. 2018. "Profile of modeling and simulation research published in the Journal of Defense Modeling and Simulation". *J. Def. Model. Simul.* vol. 15 (3), pp. 337–349.

Ndii, M. Z., E. Carnia, and A. K. Supriatna. 2018. "Mathematical models for the spread of rumors: a review". *Issues and trends in interdisciplinary behavior and social science*, pp. 65–73.

Olševičová, K., J. Procházka, and A. Danielisová. 2015. "Reconstruction of prehistoric settlement network using agent-based model in NetLogo". In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 165–175. Springer.

Patel, I., H. Nguyen, E. Belyi, Y. Getahun, S. Abdulkareem, P. J. Giabbanelli, and V. Mago. 2017. "Modeling information spread in polarized communities: Transitioning from legacy media to a Facebook world". In *SoutheastCon 2017*, pp. 1–8. Institute of Electrical and Electronics Engineers, Inc.

Pendleton, S. C. 1998. "Rumor research revisited and expanded". *Lang. Commun.* vol. 18 (1), pp. 69–86.

Pinto, R., S. Aguiar, and G. Gonçalves. 2020. "Generating Simulation Models From CAD-Based Facility Layouts". In *Proc. 9th Int. Conf. on Intelligent Systems and Applications*, pp. 24–31. IARIA.

Pitman, L., G. Nandakumar, and J. Richman. 2022. "A Gamified Synthetic Environment for Evaluation of Counter-Disinformation Solutions". *Journal of Simulation Engineering* vol. 3.

Rao, D. M., and A. Chernyakhovsky. 2019. "Automatic generation of agent-based models of migratory waterfowl for epidemiological analyses". *Int. J. Comput. Aided Eng. Technol.* vol. 11 (6), pp. 747–762.

Sagar, K., and A. Saha. 2017. "A systematic review of software usability studies". *International Journal of Information Technology*, pp. 1–24.

Schelling, T. C. 1971. "Dynamic models of segregation". *J Math Sociol* vol. 1 (2), pp. 143–186.

Schroeder, S. A., C. Vendome, P. J. Giabbanelli, and A. M. Montfort. 2022. "Towards reusable building blocks to develop COVID-19 simulation models". In *Proceedings of the 2022 Winter Simulation Conference (WSC)*. Institute of Electrical and Electronics Engineers, Inc.

Serrano, E., C. A. Iglesias, and M. Garijo. 2015. "A survey of twitter rumor spreading simulations". In *Computational Collective Intelligence*, pp. 113–122. Springer.

Sobkowicz, P., and A. Sobkowicz. 2021. "Agent based model of anti-vaccination movements: Simulations and comparison with empirical data". *Vaccines* vol. 9 (8), pp. 809.

Sturrock, D. T., and C. D. Pegden. 2011. "Recent innovations in Simio". In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pp. 52–62. Institute of Electrical and Electronics Engineers, Inc.

Sulis, E., and K. Taveter. 2022. "Agent-Based Simulation with NetLogo". In *Agent-Based Business Process Simulation: A Primer with Applications and Examples*, pp. 53–75. Springer.

United Nations 2020. "International Migration 2020: Highlights".

Vendome, C., D. M. Rao, and P. J. Giabbanelli. 2020. "How do modelers code artificial societies? investigating practices and quality of netlogo codes from large repositories". In *2020 Spring Simulation Conference (SpringSim)*, pp. 1–12. Institute of Electrical and Electronics Engineers, Inc.

Voinov, A., and H. H. Shugart. 2013. "'Integronsters', integral and integrated modeling". *Environmental Modelling & Software* vol. 39, pp. 149–158.

Zhou, M., J. Li, R. Basu, and J. Ferreira. 2022. "Creating spatially-detailed heterogeneous synthetic populations for agent-based microsimulation". *Comput Environ Urban Syst* vol. 91, pp. 101717.

AUTHOR BIOGRAPHIES

SHANNON CHENG was a senior at Sycamore High School in Cincinnati and a student at Miami University's regional campus in Hamilton, both in Ohio U.S.A. She was also a volunteer research intern in summer 2022 in the group of Dr. Giabbanelli at Miami University. Her email address is chengs12@miamioh.edu.

PHILIPPE J. GIABBANELLI received his Ph.D. from Simon Fraser University and is Associate Professor in the Computer Science & Software Engineering Department at Miami University. He has published over 110 articles, primarily on machine learning and simulation. His email address is giabbapj@miamioh.edu.

ZAIYI 'DEREK' KUANG was a senior at The Summit Country Day School in Cincinnati and a volunteer research intern in summer 2022 with Dr. Giabbanelli. His email address is derekkuang0213@gmail.com.