Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

DESIGN AND DEPLOYMENT OF A SIMULATION PLATFORM: CASE STUDY OF AN AGENT-BASED MODEL FOR YOUTH SUICIDE PREVENTION

Joshua Huddleston
Michael C. Galgoczy
Kareem A. Ghumrawi
Philippe J. Giabbanelli

Department of Computer Science
& Software Engineering
Miami University
205W Benton Hall, 510 E. High St.
Oxford, OH 45056, USA

Ketra L. Rice
Nisha Nataraj
Margaret M. Brown
Christopher R. Harper
Curtis S. Florence

National Center for Injury Prevention and Control
Centers for Disease Control and Prevention (CDC)
4770 Buford Highway NE
Atlanta, GA 30341, USA

ABSTRACT

Research has examined the process of engaging end-users in co-designing a simulation model and using it within workplace settings. In contrast, few studies document the software development aspect of creating, packaging, and deploying a simulation platform that satisfies end-users' needs. In this case study, we detail these aspects regarding a platform involving Agent-Based Modeling for youth suicide prevention. Goals and constraints are detailed in three categories of needs, each showing how competing demands are addressed via software solutions: 1) data, encompassing data security (e.g., no direct access to individual-level data) and data updates (e.g., changing a model's initialization without coding expertise); 2) deployment, to satisfy security protocols (e.g., secure intranet communications) and the end-users' experience (e.g., ease of installation); 3) accessibility, ensuring that individuals with impairments are not excluded from using simulations. By presenting practical solutions to each category, our case study supports modelers in addressing their own deployment needs.

1 INTRODUCTION

Simulation experts frequently work within interdisciplinary teams by creating models and accompanying platforms intended for end users who are domain experts. Best practices for producing these simulation models include identifying the needs and capacities of end users (Treasury 2015) as well as characterizing the environment within which they operate, since it shapes constraints in terms of platform dynamics (e.g., updating modules or data) and security (e.g., data access, authentication). These constraints impact our ability to successfully integrate a simulation platform within the existing software ecosystem of the end-users. In this paper, we describe the development and deployment of a simulation platform that must be operable within a government agency. Specifically, we examine a new suicide prevention simulation platform for which the end-users are public health researchers in suicide prevention.

Studies have emphasized the early and continuous involvement of end-users with models that support their decision-making activities (Gilbert et al. 2018), thus advocating for a 'participatory modeling' approach (Voinov et al. 2018; Giabbanelli et al. 2021). However, the nature of this involvement is often limited to the *conceptual design* (Barbrook-Johnson et al. 2019) of the model (e.g., which factors are important? which relationships are known? what level of abstraction is adequate? which intervention scenarios should be supported?) and its *validation* (e.g., how do results compare with expectations?). For example, participatory modeling studies document the involvement of stakeholders (who are also often the

end-users) in co-designing (Ahrweiler et al. 2019; Adams et al. 2021) a model or using it either within a workshop setting (Minyard et al. 2014; Lawler et al. 2018) or as part of a usability study (Giabbanelli et al. 2016). Complementary studies examined the early step of commissioning a model (Cox and Barbrook-Johnson 2021), developed frameworks on how interventions should be modeled (Skivington et al. 2021), or provided cautionary tales of overlooking a model's purpose and its limitations (Edmonds et al. 2018).

In contrast, little is known about the software part of the process, which is devoted to developing and integrating a simulation platform within an existing software ecosystem. Forgetting about the integration ultimately results in delivering platforms that suffer from low usability or cannot be adequately maintained for long term use. As noted by Gilbert et al. (2018), "securing long-term maintenance arrangements is thus a challenge that is so far rarely met properly". This challenge parallels a concern in the field of Modeling & Simulation (M&S) regarding the inability to either access models (Squazzoni et al. 2020) or reuse them (Vendome et al. 2020; Giabbanelli et al. 2019). The paucity of studies on the integration of modeling platforms in daily thus constitutes a key limitation in M&S, emphasized in calls to consider simulation platforms as *tools* subject to evaluation by their intended users (Giabbanelli et al. 2021; Giabbanelli and Crutzen 2017) rather than merely the final product or after-thought of a modeling process.

Our case study contributes to the research on the development and integration of a simulation platform built on an Agent-Based Model (ABM). In this model, each agent represents a youth (ages 10 to 19 years) whose trauma is updated yearly and can lead (depending on a balance of protective and risk factors) to suicidal thoughts, attempts, or death. Since the end-users are researchers who may operate in government agencies, our case study is subject to a comprehensive set of requirements. By presenting how each of these requirements was addressed, our study thus constitutes a 'case library' that augments the problem-solving capacity of modelers who may be faced with some of these requirements for the first time.

To facilitate this uptake of our experiences, our contributions are grouped into three subsets of requirements. The first subset is centered on *data*, which is both an essential input (to initialize a model) and the main output subject to interpretation by end-users. Several competing demands apply to data: *data security* prohibits the direct use of identifiable individual-level data for the platform, yet simulations are performed at an individual level with ABMs. In addition, *data updates* are necessary since population characteristics change over time and the platform is designed for long-term use, yet an update by one user could (unknowingly) affect the simulations performed by others without proper compartmentalization. For instance, a user could update the data once after-school programs have been implemented, while another user continues to research the potential effect of after-school programs by relying on the data as a baseline *without* such programs. The second subset is devoted to *deploying* a simulation platform, which is again subject to competing demands. Security protocols can prohibit a third-party simulation team from directly deploying software on a target system thus a platform must be *packaged* for easy deployment, yet the characteristics of the target system may be partly unknown and/or subject to change (e.g., as the end-user updates dependencies). In addition, performing large-scale simulations on a single machine can lead to a poor user experience when one attempts to use a system already under significant computational load, hence a *client/server architecture* is often used to separate a simulation web portal from the computations, which adds to the complexity of deploying the platform. Finally, *accessibility* in simulation platforms has primarily been investigated in the context of student learning (Winters et al. 2020; Smith et al. 2018) rather than with respect to inclusion for when end-users are domain experts. For example, in participatory modeling, a common perception is that visualizations are a universal medium (Marzouki et al. 2022) to engage with simulations, thus implicitly ignoring individuals with visual impairments. However, electronic and information technology (EIT) accessibility laws and policies (e.g., Section 508 of the Rehabilitation Act of 1973) require federal agencies to make their EIT accessible to people with disabilities. This, again, results in competing demands: we need to support the user experience through interactive elements (e.g., to view feedback about simulation results or set-up a new scenario), yet the user may have impairments (e.g., visual, motor control restrictions affecting hands). Each of these subsets of requirements is addressed in a dedicated section, following a brief overview of the objectives of our platform.

2 BACKGROUND: PURPOSE OF THE SIMULATION PLATFORM AND PRIOR WORK

Suicide is the second leading cause of death among adolescents ages 10-14, and third leading cause of death among adolescents ages 15-19 in the United States, accounting for 17.0% of all deaths among those aged 10-14 and 18.0% of all deaths among those aged 15-19 (Centers for Disease Control and Prevention 2020). Suicide prevention policies do not only seek to prevent death, but also to decrease their precursors: adverse childhood experiences (ACEs), and suicide ideation and attempts, which saw 461,980 emergency department visits and in-patient hospitalizations for adolescents in 2018 alone (Agency for Healthcare Research and Quality 2021). Suicide ideation, attempts, and fatality are caused by a complex web of factors including individual-level factors (e.g., mental health, substance abuse), relationships (e.g., bullying), the community (e.g., availability of mental health services), and the broader society (e.g., social norms). Previous efforts at mapping this complex system have captured from 18 concepts and 42 interactions (Page et al. 2017) up to 361 concepts and 946 interactions (Giabbanelli et al. 2021). These *conceptual models* have been operationalized as *simulation models* using various methods, such as System Dynamics (Occhipinti et al. 2021; Chung 2016) or Agent-Based Models (Keyes et al. 2019). Given the broad number of suicide-related concepts and the various socio-ecological levels at which they are situated (e.g., individual or societal), it is no surprise that there exist very diverse patterns in suicide epidemiology (Patel and Gonsalves 2019). Agent-Based Models (ABMs) can be a powerful tool to represent this heterogeneity *provided that* there is sufficient data to calibrate and validate them, as well as adequate computational power to run population-wide simulations at the individual level; these two constraints will be examined in sections 3 and 4.

A model must be *fit for purpose* or 'adequate' rather than replicating the entirety of the real-world, it should fulfill its role of decision-support tool for suicide prevention strategies of interest to end-users. These policies are "intersectoral, coordinated, and collaborative actions that target the range of risk factors" (Patel and Gonsalves 2019). Specifically, a comprehensive technical package of policies, programs, and practices to prevent suicide was developed to help communities and states prioritize prevention strategies based on the best available evidence (Stone et al. 2017). The model is thus fit for purpose if each of the actions articulated in this technical package can be tested in the ABM, with respect to its effects over *three outputs (suicide ideation, suicide attempts, and suicide fatality)*. Hence, we started by identifying corresponding concepts in our large systems map (Giabbanelli et al. 2021) for each of the suicide prevention strategies. These strategies or 'inputs' must eventually lead into observable effects over the outputs, thus we need to take a sufficient *subset* of the map that connects these inputs to outputs. This task then was guided by data availability, using graph algorithms that favored selecting concepts and interactions for which data was available in one of four U.S. based surveys or surveillance data sources that collect information on adolescent suicidal behaviors (Freund and Giabbanelli 2021). As a result, we have (i) a set of evidence-based policy actions as model inputs, and (ii) a sequence of rules leading to outputs and involving individual-level data. These were translated into an ABM executed via a platform that is the subject of the present manuscript.

3 PROVIDING BOTH DATA SECURITY AND DATA UPDATES

In our ABM, each agent is *initialized* (i.e., assigned a starting value for each attribute) and then repeatedly *updated* annually until the agent either (i) becomes an adult, in which case the agent leaves our model; or (ii) dies by suicide. The update is driven by a set of evidence-based rules, such as calculating the cumulative trauma of an agent (e.g., from adverse childhood experiences) and whether it exceeds a user-defined threshold. Simulation parameters such as this threshold can easily be exposed to end-users who can then make changes, for example via text fields or sliders in the Graphical User Interface of platforms such as NetLogo. For long-term maintainability of a simulation platform, it is necessary to also *allow end-users to change how agents are initialized*, for example to reflect changes in population demographics, the physical or social environment. If a user has *direct access* to individual-level data, then changing the initialization can be as simple as selecting an updated version of the data file, which is then read by a platform to initialize each agent based on a row of data describing a real-world individual (Giabbanelli and Crutzen

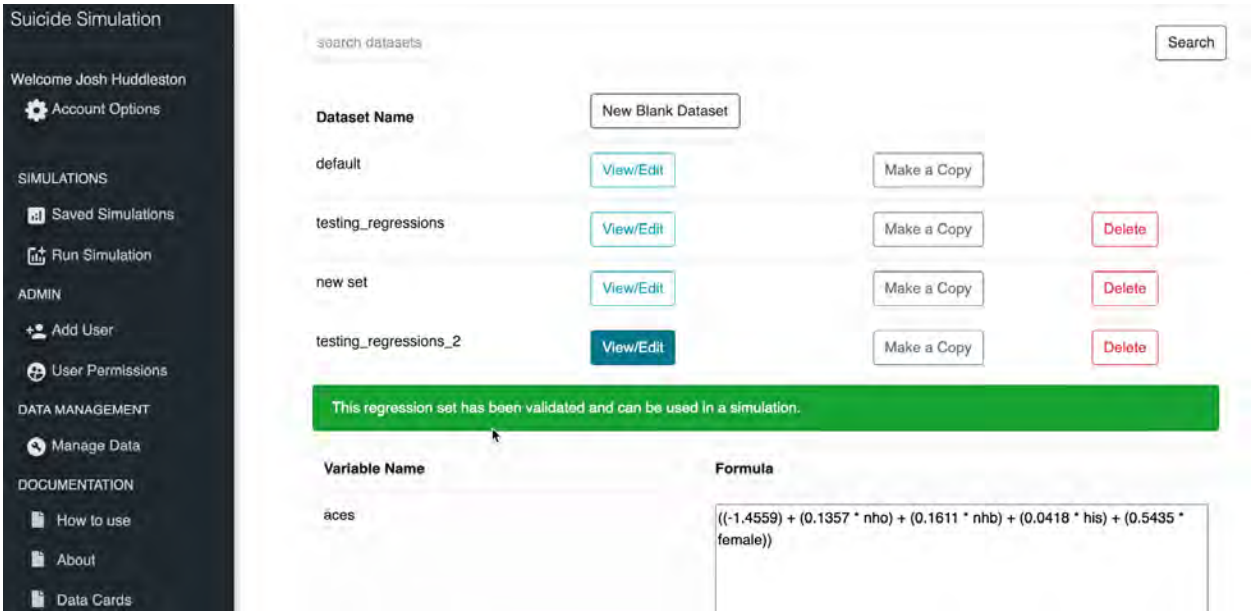

Figure 1: The regression editor allows end-users to edit the data that initializes each of the agents' attributes. Regressions must pass automated checks before being passed onto the simulation.

2013). A challenge arises when such individual-level data is not available to the end-user, for example due to *access control* policies as part of broader data security constraints. Indeed, end-users may not have access to national statistics files at the individual level, or access to these files is only granted in restricted environments which are not connected to the internet hence preventing linking with a remote simulation platform. In this case, the initialization is normally coded into the model. The matter of enabling end-users to modify the initialization of agents thus turns into the problem of exposing *some* of a model's inner logic (i.e., code that initializes agents) in a way that is both safe for the platform (e.g., user modifications cannot have impacts beyond the initialization) and feasible for users without programming backgrounds.

In our platform, these constraints are addressed through a 'dataset editor' (Figure 1). Each of the agents' attributes is governed by one regression, calibrated by a statistician with either direct access to data (in our case) or potentially from publicly available aggregated data. To change the initialization of an attribute, an end-user edits the corresponding text field. Changes are then locally saved and submitted for *checking*, which is an essential step both to give feedback to the user and to ensure that the regressions will run adequately once in the simulation. Two categories of checks are performed. At a low-level, we ensure that each variable cited in a regression does *exist*; our experience found that this frequently helps end-users to identify typos and it also prevents any form of code injection into the simulation. A 'variable name key' quickly reminds users about existing variables, while a more comprehensive 'data cards' section of the platform provides standardized definitions and recommendations for each variable in context. At a higher-level, we check that the dependencies in the regressions are *satisfiable*. This means that all factors *involved* in a regression must have been defined first; for example, in Figure 1, ACEs involve four other attributes which must have a value by the time the simulation attempts to initialize ACEs. We also check that there are no circular dependencies, such as when the initialization of a variable A depends on C, which itself ultimately involves A. These higher-level dependencies are checked by automatically building and analyzing a graph from the regressions (e.g., it must be acyclic to forbid circular dependencies).

When new data is released (e.g., yearly or biyearly), *some* regressions may change (e.g., population distribution by race and ethnicity) while others may still be valid (e.g., social norms). On one hand, re-writing all regressions from scratch just to change a few would be a frustrating user experience. On the other hand, directly making changes into a 'main set' of regressions would be risky (e.g., if users make

mistakes) and would prevent the comparison of simulations between data sources. To address these needs, users can have several sets of regressions (i.e., 'datasets'), obtained either by (i) making a blank new one and then writing all regressions; or (ii) copying the most closely related dataset and then only changing what is needed. When a simulation is configured, users also select the relevant dataset.

Because our platform supports *teams* of users, it is possible that one user modifies the dataset employed by another user. While this supports teamwork, it also shows that the 'ability' to modify data should not be taken lightly. Consequently, the user permission system for the platform allows administrators (typically team leads) to decide which team members to entrust with changing the datasets (typically statisticians). The platform is also configured to ensure that there is always at least one administrator in a team.

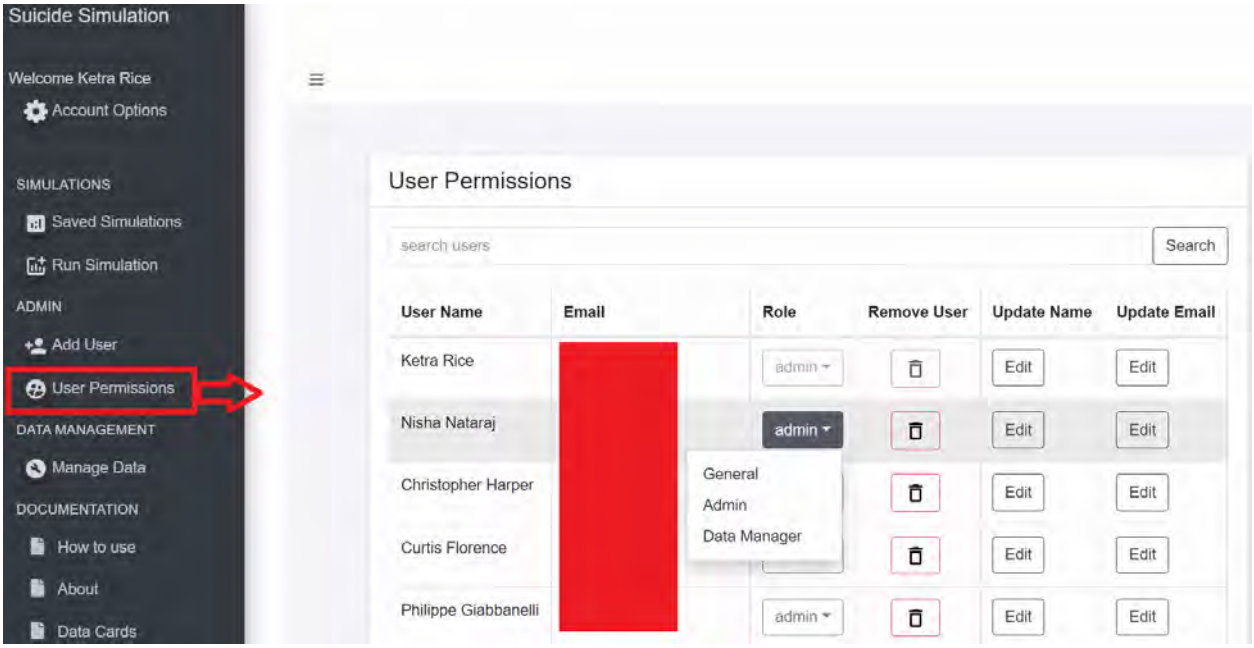

Figure 2: The user management system allows administrators ('admin') to promote a general user into a 'data manager', which lets them modify the datasets. *Red boxes are added for emphasis or data privacy.*

4 DEPLOYING A MODULAR ARCHITECTURE SUBJECT TO SECURITY PROTOCOLS

The model is stochastic, as the initialization of agents (c.f. regressions in section 3) decide the *probability* of certain baseline values. Consequently, every run of the model must be performed several times to obtain a distribution of the outcomes (suicide ideation, attempts, fatality). The duration for each of these runs increases as a function of the virtual population size (i.e., number of agents) chosen by the end-user. While a model does not need a virtual agent for every real-world person (i.e., 1:1 scale), a user may still require a sufficiently large population to study the specific experiences of agents with multiple characteristics (e.g., Non-Hispanic Black, 8-10 year old, Male, using healthcare services), which can become computationally intensive. Performing these runs on the *same* machine used to access the simulation platform (i.e., sharing CPU cycles) would thus become a problem, as either (i) heavy computations from simulations deteriorate the responsiveness of the platform and hence impact the user experience or (ii) user actions on the platform impact how long it takes to perform the simulations, resulting in fluctuating estimates for users to obtain their results. Consequently, we dissociate the *front-end* of the simulation, consisting of the Graphical User Interface through which users configure and analyze simulations, from the *back-end*, which performs simulations. This leads to a classic web client and server architecture, akin to how `AnyLogic` offers a web platform and then runs computations elsewhere (via a cloud). When a new simulation is requested

from the web portal, it is sent to the server, which then estimates how long it will take (depending on its current load and the requested population size) and informs the user. The overall architecture is shown in Figure 3.

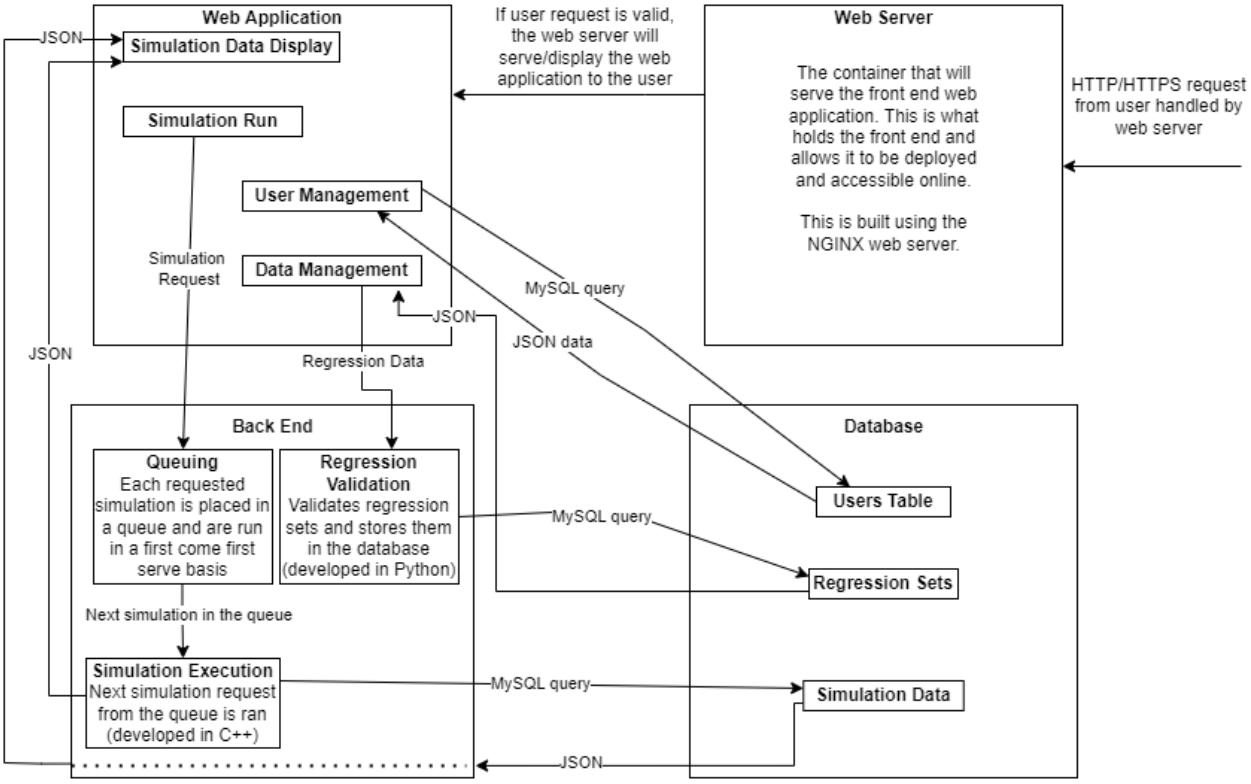

Figure 3: Blueprint of the client/server architecture and `Docker` containers.

Similar to other platforms (Bogach et al. 2017), we use C++14 to execute a simulation (due to performance) and Python 3.9 to analyze parameters (e.g., check regressions) and package the results. To package the complete architecture and facilitate its deployment with end-users, we use `Docker` 20.10, which is a *platform as a service* (PaaS) that uses operating system level visualization to deliver software in packages called *containers*. Unlike Marrocco et al. (2019) where *some* simulation components were in containers and others (e.g., `GAMA` and a client) were not, we ensured that *the entire simulation platform* could be deployed with `Docker`. This leads to four containers (Figure 3): the database, back-end, front-end, and web server. The `MySQL` 5.7 database stores user data (e.g., to differentiate admins from data managers), simulation data (for later retrieval and analysis), and the regression sets. The back-end built with `Laravel Lumen` checks the regressions and performs the simulations, whose results are both reported to the front-end and archived in the database. Reporting to the front-end allows the visualization of results upon their completion (Figure 4), while archiving in the back-end serves the dual purpose of (i) comparing results across simulations (e.g., identify which prevention strategy performs best); and (ii) supporting replicability efforts, in line with best practices in M&S. The front-end container holds the user interface. In the front-end, a user signs up or logs in to their account and can request new simulations or analyze previous ones. Administrative users also have the ability to add regression sets as well as add/remove users and change user roles (c.f., section 3). The last container is the web server using `NGINX`, which *serves* the front-end such that it is accessible to users on the deployed server. The web server handles the HTTP/HTTPS requests when a user seeks to access the site and, if successful, then displays the front-end.

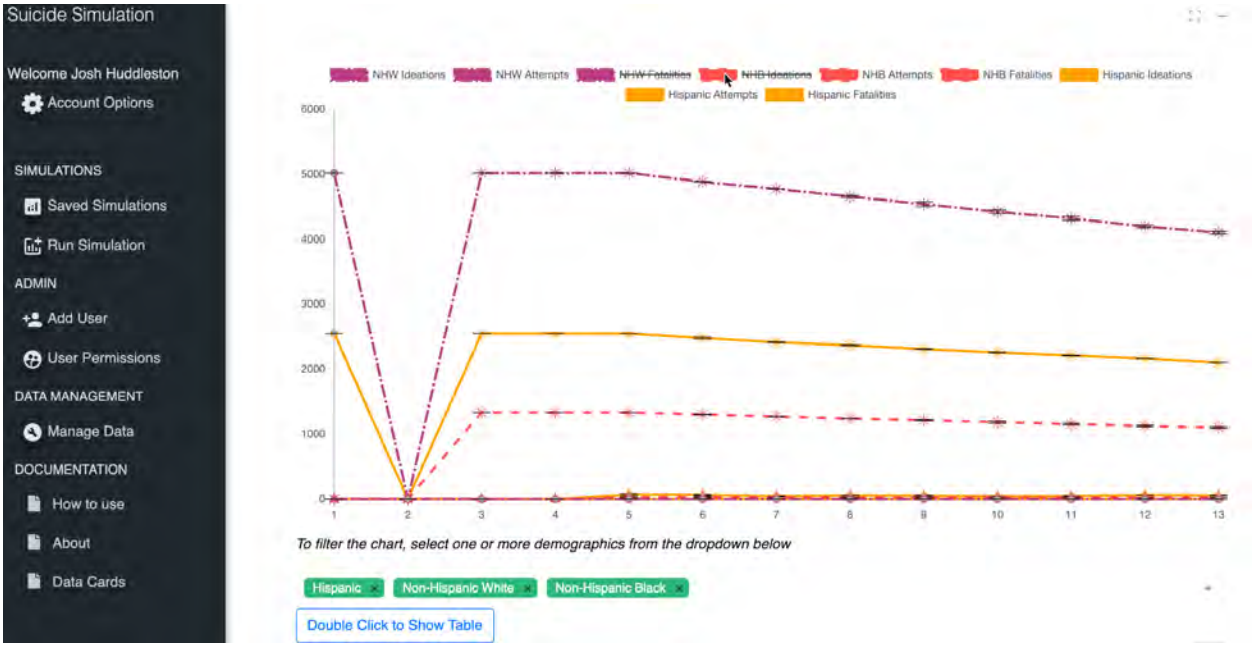

Figure 4: Results passed by the back-end are then visualized in the front-end. Ideations, attempts, and fatalities are broken down by race and ethnicity, which can be used as filters by end-users.

For security, the server can only be accessed through *HTTPS* (i.e., via port 443), which is achieved by *OpenSSL* certificates. A certificate and corresponding private key were generated and placed inside the web server container; that is, the configuration file of the web server directly points to the certificate and key. After these steps, the server is compliant with end-users security protocols by using secure sockets layer (SSL) and HTTPS, hence we can proceed with packaging.

Due to security protocols, we cannot assume that the end-user has access to the internet, hence the deployment can (i) neither rely on pulling from repositories such as the Docker Hub nor (ii) download dependencies. These constraints affect how the platform is packaged. We thus pre-install all dependencies, then package each docker container individually (as *tar* files) to maintain the dependency installations. These four files, the database migration file, and the main application configuration file, are finally gathered into a single archive (*zip* file) which is ready for deployment. Once the end-user has moved the archive to their machine, they follow a simple three steps process. First, they unzip the archive. Second, they load all containers (i.e., tar files), which will also load the installed dependencies. Then, using *Docker compose* 3.8, they can run the server and perform the database migration. At this point, the platform is ready and users can access it via their local machine (e.g., via localhost:8001).

Overall, our client/server architecture combined with several Docker containers and pre-installed dependencies was able to support a successful integration of the ABM platform on a secure server within one group of public health researchers. Security protocols were satisfied through the use of appropriate ports (using OpenSSL to generate a certificate/key then communicating via HTTPS) and a complete packaging of the platform such that no internet access is required for deployment.

5 DESIGNING A SIMULATION PLATFORM FOR ACCESSIBILITY

End-users of simulation platforms represent a broad range of skills and backgrounds, thus it is important to consider platforms from an inclusive perspective. This can also be a legal requirement for government organizations, for example relevant U.S. guidelines such as Section 508 (https://www.section508.gov/) requires to *provide access* to users with disabilities. To appreciate the applicability of this guideline, it is

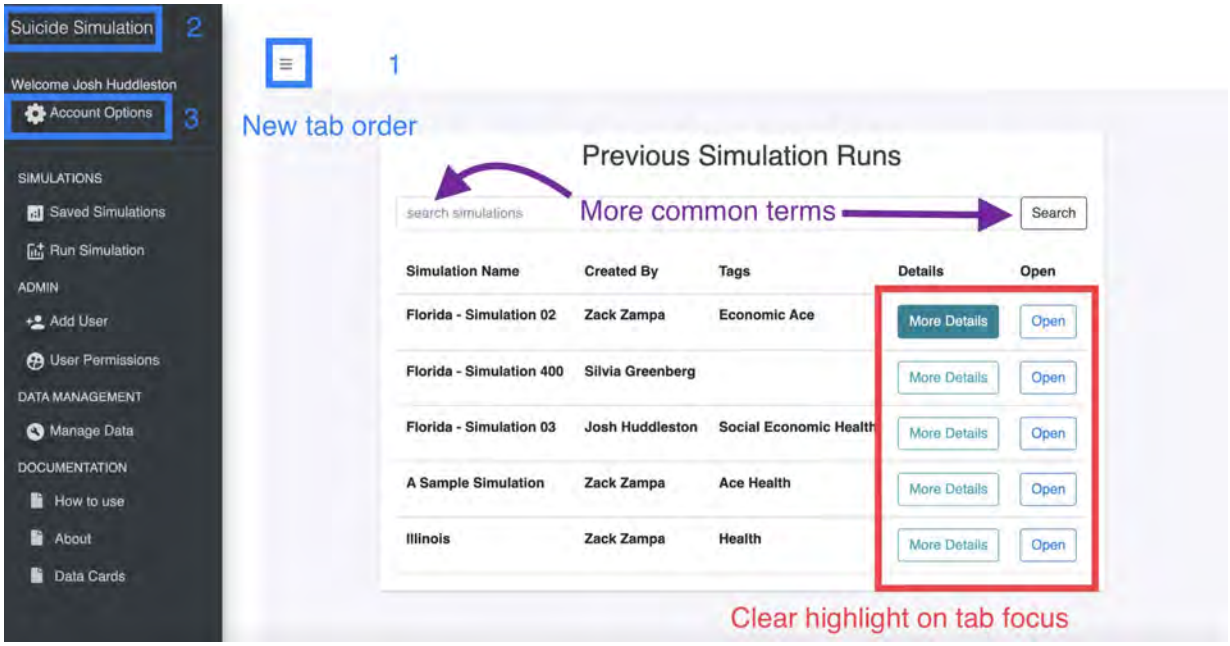

Figure 5: The homepage illustrates the ordering of tabs (for navigation via a keyboard), the clarity of terminology (e.g., 'search'), and the visible highlights when a button is in focus.

important to characterize the end-users of the simulation platform. Indeed, the 508 guidelines require the ability to provide manual assistance to anyone using the platform, should it be needed. If the platform was openly accessible, that requirement may need a dedicated help-desk with a ticket system. Given that our end-users are public health researchers, our objective is to *maximize compliance* with accessibility guidelines and hence minimize the possibility of needing manual assistance. At a high level, this includes making all content readable by a screen reader (Moore et al. 2018) (including providing captions for complex and dynamic images of simulation results), navigating through pages involved in defining/analyzing simulations with the keyboard alone, and using color schemes mindful of people with color blindness.

Ensuring our content is always **readable** starts with paying close attention to the *terminology* and keeping it uniform throughout the platform. For example, simulations that have been executed (i.e., previous runs) are available through a list. While 'filtering' a list is a common keyword for developers, 'search' is a more frequently used term for the operation at-hand and users can then assume that they will be presented with a search result instead of having to search through a filtered list. Similarly, the habit of editing a user's account by clicking on their name is not as meaningful as a button labeled 'account options' (Figure 5). To make the content readable, the code also had to be extended with ARIA-labels and descriptions for input fields, links, and tables. An *ARIA-label* is an additional markup to the HTML, which provides a more detailed description of the HTML element and its context; these descriptions are then read by screen readers. A challenge is to support screen readers when visualizing simulation results. When visuals are simple and static, standard practice to make them accessible is to add alternative text that describes the visuals. As charts and graphs get more complex (Figure 7), practice switches to providing a caption and a data table to supplement the visualization. Our platform thus sends these results in two ways: as a table summarizing the outcomes of the model (which can be used by screen readers given row/column names) and as detailed charts which can be used for visual, *interactive* data exploration. To avoid reading a massive amount of data at once, our platform offers the ability to break down these charts (and their tabular counterparts) by differing demographics. In addition, the complete data from the simulation (Figure 6) or rendered in a specific chart can be downloaded, to be handled by third-party accessible software (e.g., as Excel spreadsheets that conform to the Revised 508 Standards). Note that there are also innovative browser

extensions, such as deep neural networks that decode charts on a webpage for screen readers (Choi et al. 2019); our platform has no direct control over the extensions adopted by a user and further studies are needed to evaluate the quality of transcriptions for visualization of simulation results.

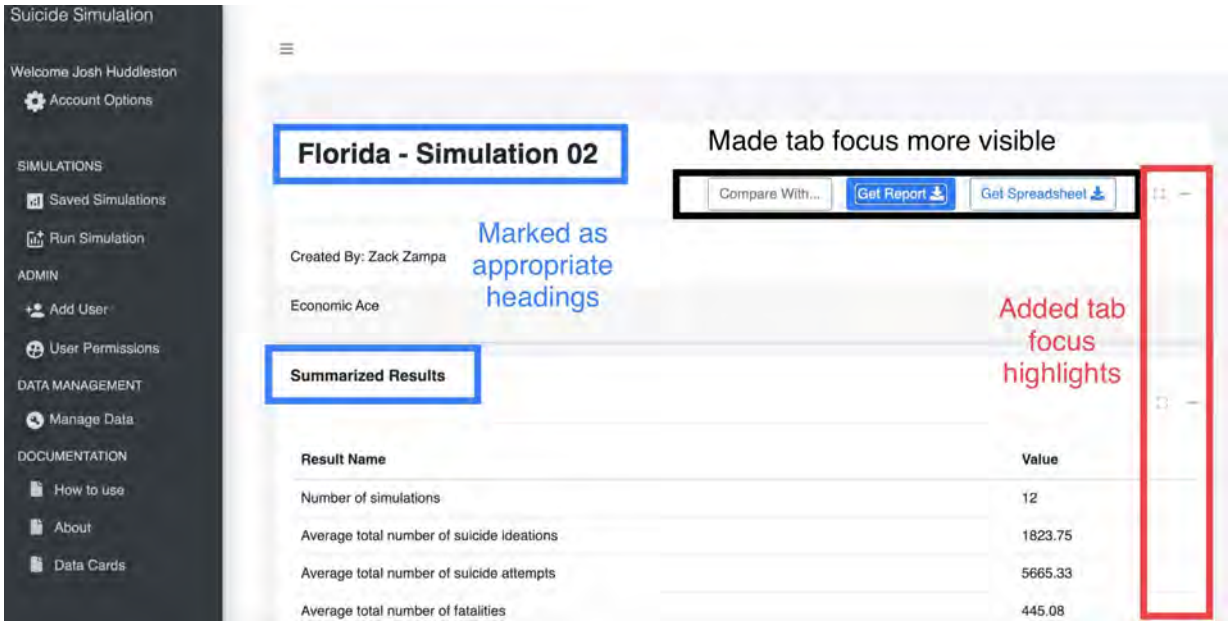

Figure 6: Improving accessibility for simulation results included the ability to download all results as a spreadsheet, or export an automatic executive summary (including simulation parameters and results).

Overall, the 508 guidelines contain a total of 79 specific items; for full disclosure, the complete list of items together with our explanations for compliance are provided on the Open Science Framework at https://osf.io/7nxp4/. Since 23 items are not applicable to a simulation platform (e.g., if a page uses multiple languages), our compliance with 42 items covers 75% of the guideline. The remaining points for non-compliance thus require third-party software or manual assistance. For example, guideline 6A states that all functionalities should be available using the keyboard, except for tasks such as drawing. We do not meet this guideline as it is not currently possible to drive interactive charts of simulation results with the keyboard. To allow this interaction, the HTML canvas element for each type of chart would need to include event handlers, ensure that keyboard controls do not override any existing accessibility control, and provide the exact same functionality as the user can get with a mouse. There is thus still a need for greater accessibility in libraries that commonly support interactive visualizations within simulation platforms.

6 CONCLUSION

Despite the growing use of Agent-Based Models in public health, few studies (Buckeridge et al. 2011) have examined the software infrastructure that supports them via custom simulation platforms. Our case study documents how this infrastructure is created as a result of competing demands in terms of security protocols (thus preventing direct data access or cloud-based resources), user experience, and accessibility. Our work thus contributes to a research gap in the development and integration of ABMs within the software ecosystems of end-users, which is critical for both the immediate success of a modeling project and its long-term maintainability. Although another team may not commonly be faced with the *complete* combination of goals and demands from our case study, our approach to grouping constraints into three categories can support teams into reusing *some* of our solutions to address their own practical needs.

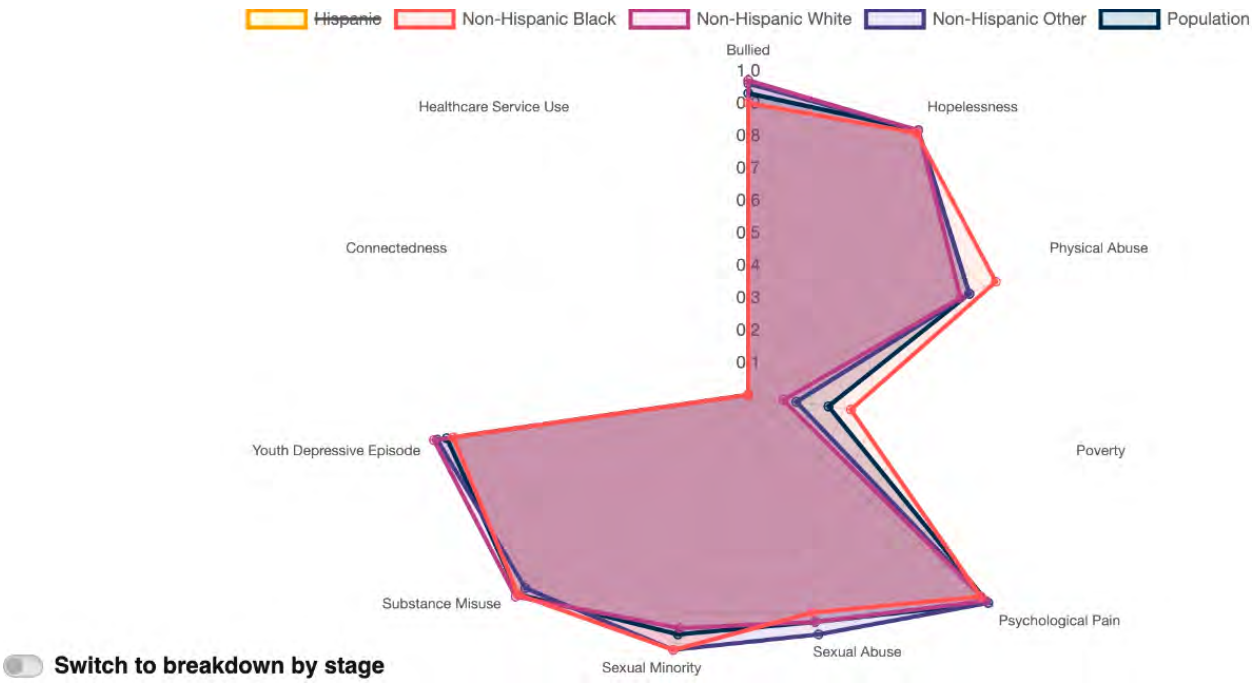

Figure 7: Results allow end-users to examine *why* agents had ideation, attempts, or fatalities with respect to key protective (connectedness, healthcare service use) and risk factors (e.g., physical abuse, poverty). Numbers show the fraction of agents who experienced each of these factors. Results can be further filtered by race and ethnicity, exemplified here by filtering out Hispanic. For example, Non-Hispanic Blacks (red curve) had the highest poverty and physical abuse, but the lowest (albeit elevated) level of sexual abuse.

ACKNOWLEDGMENTS

We thank Zack Zampa and Megan Foreman (Miami University) for their contributions to designing and implementing the Graphical User Interface (GUI) of our platform. We appreciate the input of Nimesh Patel (CDC) on deploying the platform. Finally, we thank CDC personnel for feedback on the manuscript. *Disclaimer: The findings and conclusions in this paper are those of the authors and do not necessarily represent the official position of the Centers for Disease Control and Prevention.*

REFERENCES

Adams, S., T. Rhodes, and K. Lancaster. 2021. "New Directions for Participatory Modelling in Health: Redistributing Expertise in Relation to Localised Matters of Concern". *Global Public Health*:1–15.

Agency for Healthcare Research and Quality 2021. "Healthcare Cost and Utilization Project (HCUP)". https://www.hcup-us.ahrq.gov/, accessed 21st September 2022.

Ahrweiler, P., D. Frank, and N. Gilbert. 2019. "Co-Designing Social Simulation Models for Policy Advise: Lessons Learned from the INFSO-SKIN Study". In *Proceedings of the 2019 Spring Simulation Conference (SpringSim)*, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Barbrook-Johnson, P., C. Schimpf, and B. Castellani. 2019. "Reflections on the Use of Complexity-Appropriate Computational Modeling for Public Policy Evaluation in the UK". *Journal on Policy and Complex Systems* 5(1).

Bogach, N., V. Dyachkov, A. Lamtev, and Y. Lezhenin. 2017. "Agent-Based Modeling Software for Natural and Rural Ecosystems". *Engineering for Rural Development* 16:742–747.

Buckeridge, D. L., C. Jauvin, A. Okhmatovskaia, and A. D. Verma. 2011. "Simulation Analysis Platform (SnAP): A Tool for Evaluation of Public Health Surveillance and Disease Control Strategies". In *AMIA Annual Symposium Proceedings*, Volume 2011, 161. American Medical Informatics Association.

Centers for Disease Control and Prevention 2020. "National Center for Injury Prevention and Control. Web-based Injury Statistics Query and Reporting System (WISQARS). Fatal Injury and Violence Data". https://www.cdc.gov/injury/wisqars/index.html, accessed 21st September 2022.

Choi, J., S. Jung, D. G. Park, J. Choo, and N. Elmqvist. 2019. "Visualizing for the Non-Visual: Enabling the Visually Impaired to Use Visualization". 38(3):249–260.

Chung, S. Y. 2016. *Suicide Attempts from Adolescence into Young Adulthood: A System Dynamics Perspective for Intervention and Prevention*. Ph.D. thesis, Washington University in St. Louis, St. Louis, MO, USA. https://openscholarship.wustl.edu/art_sci_etds/722/, accessed 21st September 2022.

Cox, J., and P. Barbrook-Johnson. 2021. "How Does the Commissioning Process Hinder the Uptake of Complexity-Appropriate Evaluation?". *Evaluation* 27(1):32–56.

Edmonds, B. et al. 2018. "Using Agent-Based Modelling to Inform Policy–What Could Possibly go Wrong?". In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, edited by P. Davidsson and H. Verhagen, 1–16. Springer.

Freund, A. J., and P. J. Giabbanelli. 2021. "The Necessity and Difficulty of Navigating Uncertainty to Develop an Individual-Level Computational Model". In *Computational Science – ICCS 2021*, edited by M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, 407–421. Springer.

Giabbanelli, P., and R. Crutzen. 2013. "An Agent-Based Social Network Model of Binge Drinking Among Dutch Adults". *Journal of Artificial Societies and Social Simulation* 16(2):10.

Giabbanelli, P., R. Flarsheim, C. Vesuvala, and L. Drasic. 2016. "Developing Technology to Support Policymakers in Taking a Systems Science Approach to Obesity and Well-Being". *Obesity Reviews* 17:194–195.

Giabbanelli, P. J., J. Badham, B. Castellani, H. Kavak, V. Mago, A. Negahban, and S. Swarup. 2021. "Opportunities and Challenges in Developing Covid-19 Simulation Models: Lessons From Six Funded Projects". In *Proceedings of the 2021 Annual Modeling and Simulation Conference (ANNSIM)*, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Giabbanelli, P. J., and R. Crutzen. 2017. "Using Agent-Based Models to Develop Public Policy About Food Behaviours: Future Directions and Recommendations". *Computational and Mathematical Methods in Medicine* 2017.

Giabbanelli, P. J., M. C. Galgoczy, D. M. Nguyen, R. Foy, K. L. Rice, N. Nataraj, M. M. Brown, and C. R. Harper. 2021. "Mapping the Complexity of Suicide by Combining Participatory Modeling and Network Science". In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'21)*, 339–342. New York, NY, USA: Association for Computing Machinery.

Giabbanelli, P. J., A. A. Voinov, B. Castellani, and P. Törnberg. 2019. "Ideal, Best, and Emerging Practices in Creating Artificial Societies". In *2019 Spring Simulation Conference (SpringSim)*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gilbert, N., P. Ahrweiler, P. Barbrook-Johnson, K. P. Narasimhan, and H. Wilkinson. 2018. "Computational Modelling of Public Policy: Reflections on Practice". *Journal of Artificial Societies and Social Simulation* 21(1).

Keyes, K. M., A. Hamilton, J. Swanson, M. Tracy, and M. Cerdá. 2019. "Simulating the Suicide Prevention Effects of Firearms Restrictions Based on Psychiatric Hospitalization and Treatment Records: Social Benefits and Unintended Adverse Consequences". *American Journal of Public Health* 109(S3):S236–S243.

Lawler, K., G. Landers, K. Minyard, and K. Fuller. 2018. "Using Systems Tools to Advance Collective Impact". In *Using Collective Impact to Bring Community Change*, edited by N. Walzer and L. Weaver, 78–95. Routledge.

Marrocco, L., E. C. Ferrer, A. Bucchiarone, A. Grignard, L. Alonso, K. Larson, and A. S. Pentland. 2019. "BASIC: Towards a Blockchained Agent-Based SImulator for Cities". In *Massively Multi-Agent Systems II*, edited by D. Lin, T. Ishida, F. Zambonelli, and I. Noda, 144–162. Cham: Springer International Publishing.

Marzouki, A., S. Mellouli, and S. Daniel. 2022. "Understanding Issues with Stakeholders Participation Processes: A Conceptual Model of SPPs' Dimensions of Issues". *Government Information Quarterly*:101668.

Minyard, K. J., R. Ferencik, M. Ann Phillips, and C. Soderquist. 2014. "Using Systems Thinking in State Health Policymaking: an Educational Initiative". *Health Systems* 3(2):117–123.

Moore, E. B., T. L. Smith, and J. Greenberg. 2018. "Keyboard and Screen Reader Accessibility in Complex Interactive Science Simulations: Design Challenges and Elegant Solutions". In *Universal Access in Human-Computer Interaction. Methods, Technologies, and Users*, edited by M. Antona and C. Stephanidis, 385–400. Cham: Springer International Publishing.

Occhipinti, J.-A., A. Skinner, F. Iorfino, K. Lawson, J. Sturgess, W. Burgess, T. Davenport, D. Hudson, and I. Hickie. 2021. "Reducing Youth Suicide: Systems Modelling and Simulation to Guide Targeted Investments Across the Determinants". *BMC Medicine* 19(1):1–13.

Page, A., J.-A. Atkinson, M. Heffernan, G. McDonnell, and I. Hickie. 2017. "A Decision-Support Tool to Inform Australian Strategies for Preventing Suicide and Suicidal Behaviour". *Public Health Research & Practice* 27(2).

Patel, V., and P. P. Gonsalves. 2019. "Suicide Prevention: Putting the Person at the Center". *PLoS Medicine* 16(9):e1002938.

Skivington, K., L. Matthews, S. A. Simpson, P. Craig, J. Baird, J. M. Blazeby, K. A. Boyd, N. Craig, D. P. French, E. McIntosh et al. 2021. "A New Framework for Developing and Evaluating Complex Interventions: Update of Medical Research Council Guidance". *BMJ* 374:n2061.

Smith, T. L., J. Greenberg, S. Reid, and E. B. Moore. 2018. "Parallel DOM Architecture for Accessible Interactive Simulations". In *Proceedings of the 15th International Web for All Conference*. New York, NY, USA: Association for Computing Machinery.

Squazzoni, F., J. G. Polhill, B. Edmonds, P. Ahrweiler, P. Antosz, G. Scholz, E. Chappin, M. Borit, H. Verhagen, F. Giardini et al. 2020. "Computational Models that Matter During a Global Pandemic Outbreak: A Call to Action". *Journal of Artificial Societies and Social Simulation* 23.

Stone, D. M., K. M. Holland, B. N. Bartholow, A. E. Crosby, S. P. Davis, and N. Wilkins. 2017. *Preventing Suicide: A Technical Package of Policies, Programs, and Practice*. Atlanta, GA, USA: National Center for Injury Prevention and Control, Centers for Disease Control and Prevention.

Treasury, H. 2015. *The Aqua Book: Guidance on Producing Quality Analysis for Government*. London, UK: HM Government.

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 *Proceedings of the 2020 Spring Simulation Conference (SpringSim)*, edited by F. J. Barros, X. Hu, H. Kavak, and A. A. D. Barrio, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Voinov, A., K. Jenni, S. Gray, N. Kolagani, P. D. Glynn, P. Bommel et al. 2018. "Tools and Methods in Participatory Modeling: Selecting the Right Tool for the Job". *Environmental Modelling & Software* 109:232–255.

Winters, R. M., E. L. Harden, and E. B. Moore. 2020. "Co-Designing Accessible Science Education Simulations with Blind and Visually-Impaired Teens". In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20)*. New York, NY, USA: Association for Computing Machinery.

AUTHOR BIOGRAPHIES

JOSHUA HUDDLESTON is a software developer at Epic Systems Corporation ('Epic'). He received a BS in Computer Science at Miami University and worked with Dr Giabbanelli on front-end design for the CDC. His email address is huddlejt@miamioh.edu.

MICHAEL C. GALGOCZY is a software engineer at John Deere. He received a BS in Computer Science at Miami University and worked with Dr Giabbanelli on back-end implementation for the CDC. His email address is galgocmc@miamioh.edu.

KAREEM A. GHUMRAWI is a cyber software engineer at Northrop Grumman. He received a MS in Computer Science at Miami University and worked with Dr Giabbanelli on Agent-Based Modeling. His email address is ghumraka@miamioh.edu.

PHILIPPE J. GIABBANELLI is an Associate Professor of Computer Science & Software Engineering at Miami University. He holds a Ph.D. from Simon Fraser University. He has over 100 publications, primarily on Modeling & Simulation and Machine Learning. He is an associate editor for five journals, including SIMULATION. His email address is giabbapj@miamioh.edu.

KETRA L. RICE is a Health Economist in the Division of Injury Prevention at the Centers for Disease Control and Prevention. She holds a Ph.D. in Public Policy and Management from The Ohio State University and her research interests include health economics, health policy, and health disparities. Her email is wss1@cdc.gov.

NISHA NATARAJ is a Statistician in the Division of Overdose Prevention at the Centers for Disease Control and Prevention. She holds a Ph.D. in Industrial and Systems Engineering from North Carolina State University. Her research interests include applied analytical and predictive modeling as applied to health systems research and delivery. Her email address is nzo6@cdc.gov.

MARGARET M. BROWN was a Behavioral Scientist in the Division of Injury Prevention at the Centers for Disease Control and Prevention. She is now at the Department of Defense. She holds a Doctorate in Public Health (DrPH) in Health Management and Policy from the University of Kentucky. Her email address is margaret.m.brown87.civ@mail.mil.

CHRISTOPHER R. HARPER is a Behavioral Scientist in the Division of Violence Prevention at the Centers for Disease Control and Prevention. He holds a Ph.D. in Psychology from Georgia State University. His email address is xgj4@cdc.gov.

CURTIS S. FLORENCE is the Senior Health Economist and Economics Team Lead for the Division of Injury Prevention at the Centers for Disease Control and Prevention. He holds a Ph.D. from the University of North Carolina at Chapel Hill. He previously served as a faculty member at Emory University and Tulane University. His email address is gul4@cdc.gov.