UDC 004.4'2
DOI https://doi.org/10.32782/2663-5941/2023.2.1/37

Sarnatskyi V.V.
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

Baklan I.V.
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

CTRACEENV: A PLATFORM FOR DEVELOPMENT AND ANALYSIS OF AGENT-BASED EPIDEMIOLOGICAL MODELS USING CTRACE LANGUAGE

Over the past few years, epidemiological modeling has become an increasingly crucial aspect of controlling and managing infectious diseases. One of the most promising approaches for modeling infectious disease dynamics is agent-based models (ABMs), which offer a detailed and flexible framework for simulating disease transmission in populations. In this study, we describe our efforts to create an environment that simplifies the development and analysis of ABMs for epidemiological modeling. Our approach is based on the use of a CTrace modeling language, which provides tools for defining ABMs with contact tracing transmission.

Our approach is centered around the CTrace modeling language, which offers an intuitive syntax for defining ABMs with contact tracing transmission. With the language's user-friendly tools, modelers can specify the individual agents in a population, their interactions, and how the disease spreads among them. Furthermore, the language's compiler is capable of generating software libraries, which we successfully utilized in the environment we developed.

To assist the development of ABMs using CTrace, we have designed an environment that facilitates the integration of the language with external tools. The environment includes an intuitive user interface that guides users through the process of defining the model and running simulations. Additionally, one of the environment's key features is its capability to perform basic analysis of the model, such as displaying epidemic statistics over time. This information can then be exported for further analysis using external tools like R or Python.

Our study highlights the importance of creating user-friendly environments that simplify the development and analysis of ABMs for epidemiological modeling. The use of CTrace modeling language and the developed environment offers a powerful and efficient tool for researchers to simulate disease transmission and investigate the effectiveness of various control strategies. As such, our approach may prove invaluable in the development of more accurate and effective models for managing infectious diseases.

Key words: *epidemiology, modeling, agent-based, software, modeling environment.*

Introduction. Since the early days of human civilization, infectious disease outbreaks caused significant damage to the population [1, p. 1; 2, p. 21; 3, p. 150]. For example, the ongoing COVID-19 pandemic of SARS-CoV-2 virus [4, p. 536] caused death of over 6.8 million people as on March 2023. Apart from human casualties, the economic impact of such global pandemics can also be devastating [5, p. 1].

In order to lower the impact of these events, active disease spread prevention measures must be taken. These include obligatory face mask regime, vaccination, partial quarantine, etc. But, too strict quarantine can have significant impact on the economy as, for example, workers of critical infrastructure must remain at their workplace. Finding this point of equilibrium between epidemiological and economical damage requires powerful and representative models, the development of which is complicated due to the size and complexity of the target environment.

The modeling software tools can be a solution to this problem, as they can significantly simplify the development of epidemiological model by exposing easy-to-use human interface and untying model computational performance from user’s experience in programming. The development of powerful, performant and flexible tools for epidemiological modeling became a hot topic with the rapid advance in available computational power and accumulated amounts of statistical data.

The main contributions of this paper are summarized as follows:

- This paper reviews the current state of ready-to-use epidemiological modeling environments;
- The epidemiological modeling and analysis tool, based on a domain-specific programming language was developed;

- The computational performance of the developed tool was compared to the existing ones.

Task statement. The aim of this research is to develop an epidemiological modeling and analysis environment, based on CTrace modeling language; compare the developed solution to the existing ones.

Analysis of recent research and publications. In this section, we describe our findings on existing tools and methods of epidemiological modeling. These tools can be divided into three distinct groups:

- Programming languages, modules/libraries;
- Specialized software for epidemiological modeling;
- General-purpose modeling environments.

We review each group separately.

Programming language modules/libraries

Kendrick [6, p. 1] is both a domain-specific programming language and environment for compartmental epidemiological models development and analysis. The language allows one to develop a model utilizing domain-specific abstractions such as disease, transmission rules, compartments, etc. For example, the following listing, provided by the authors, defines a model for measles disease:

```
KendrickModel SEIR
attribute: #( status -> S E I R );
parameters: #( beta lambda gamma sigma mu );
transitions: #(
S -- lambda --> E.
E -- sigma --> I.
I -- gamma --> R.
status -- mu --> Empty.
Empty -- mu --> S.
).
Composition Measles
model: 'SEIR'.

Scenario MeaslesParameters
on: 'Measles';
beta: 0.0000214;
gamma: 0.143;
mu: 0.0000351;
sigma: 0.125;
lambda: #( beta*I ).

Scenario MeaslesPopulation
on: 'Measles';
populationSize: 100000;
S: 99999;
I: 1;
others: 0.
```

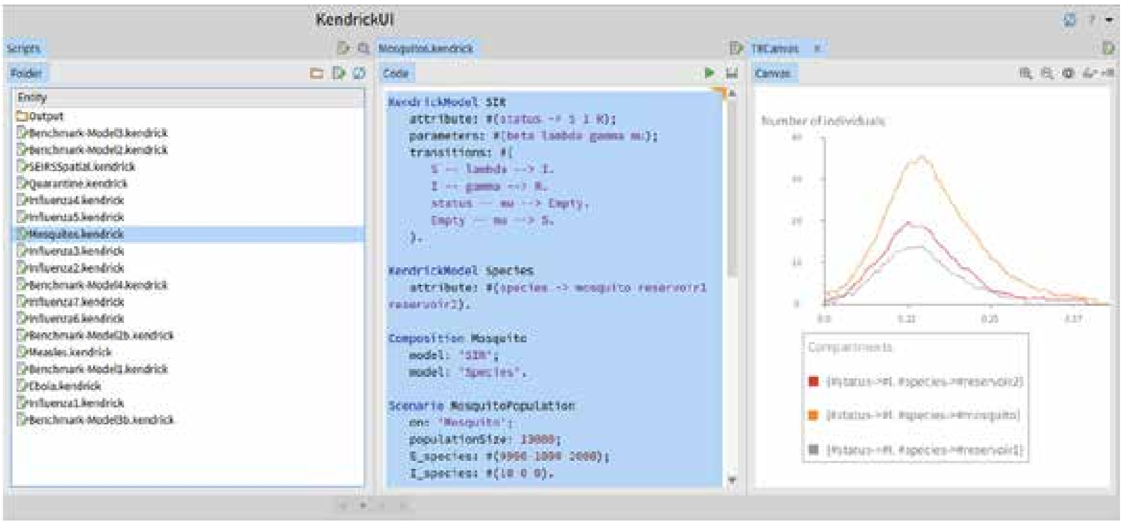
These models can be developed and analyzed using Kendrick modeling platform (Fig. 1), which provides syntax highlighting, model simulation results.

Even though Kendrick can be related to the second group, we think that the language is the main contribution of the research.

Swarm [7, p. 1] is a set of libraries for Java programming languages which defines a model as a set of interacting agents, dynamics of their evolution and the schedule of events. This package is no longer maintained and has its successor — Ascape [8, p. 212], which was simplified from the perspective of programming interface. Ascape also delivers a simple graphical interface for model adjustments and inspection and spreadsheet data export.

Specialized software for epidemiological modeling

FluTE [9, p. 1] is a software component implemented in the C/C++ programming language,



```
KendrickModel SIR
    attribute: #(status -> S I R);
    parameters: #(beta lambda gamma mu);
    transitions: #(
        S -- lambda --> I.
        I -- gamma --> R.
        status -- mu --> Empty.
        Empty -- mu --> S.
    ).

KendrickModel Species
    attribute: #(species -> mosquito reservoir1
reservoir2).

Composition Mosquito
    model: 'SIR';
    model: 'Species'.

Scenario MosquitoPopulation
    on: 'Mosquito';
    populationSize: 13000;
    S_species: #(9000 1000 2000);
    I_species: #(10 0 0).
```

Fig. 1. Kendrick modeling platform interface

built on the top of the agent-based epidemiological model. FluTE models a typical American society divided into communities of 500-3000 people. Each community is represented by families with 1-7 people in each, living in private houses. At night, relatives come in contact with each other, as well as with other members of their community, contributing to the spread of the disease. During the day, children visit educational institutions assigned to each community and interact with other children in their study groups. Preschool children come into contact with other children by playing with them on playgrounds. During this time, busy adults visit workplaces assigned to their own or neighboring communities, making contact with their colleagues there. Both adults and children can join short-term trips that simulate vacations and other trips. The process of introducing a disease into society occurs through the infection of a random group of the population at the beginning of the simulation, or every day. All data, from the age distribution of individuals to the distribution of the number of nights a person spends on vacation, is built based on the statistical data of the society of the United States of America.

Infection occurs through contact tracing, with the probability of person-to-person transmission calibrated to disease incidence data for H1N1 and H2N2 virus subtypes. The software component corresponds to a configuration file that allows you to parameterize the model.

The authors used FluTE to model the spread of influenza within the city of Seattle and throughout the United States. Figure 2 presents the simulation results according to FluTE.

Global Epidemic and Mobility (GLEaM) and the corresponding analysis environment — GLEaMvis [10, p. 1] is a software complex with client-server architecture for creating, configuring and analyzing compartment models. It consists of three main components: a custom software component, an intermediate service, and a simulation engine. The first of them provides the user with the ability to configure the compartments in detail, the connections between them, the details of the simulation, and view analytics on the results of the model. Figure 3 shows the user interface of the compartment model builder, figure 4 shows the simulation setup steps. The intermediate service and the simulation engine are responsible for interaction with the client software component and the operation of the epidemiological model. The compartmental model underlying GLEaMvis consists of three layers: population, mobility, and epidemiology. The first layer is statistical data on the population in the form of a grid with a grid step of 15 arc minutes. The mobility layer is suitable for modeling population movements over both short and long distances. The epidemiological layer simulates the dynamics of the spread of an infectious disease within individual communities according to the compartment model specified by the user.

GLEaMviz provides the user with the opportunity to view detailed statistics on the model's performance in visual visualization, an example of which is shown in figure 5.

Spatiotemporal Epidemiological Modeler (STEM) [11, p. 134] is an environment for modeling the spread of infectious diseases implemented in the Java programming language. The STEM epidemiological model use graphs to represent geographic locations and traffic flows between them. Using the built-in graph editor (figure 6, the user can create a mobility layer of the developed model. This layer can include several

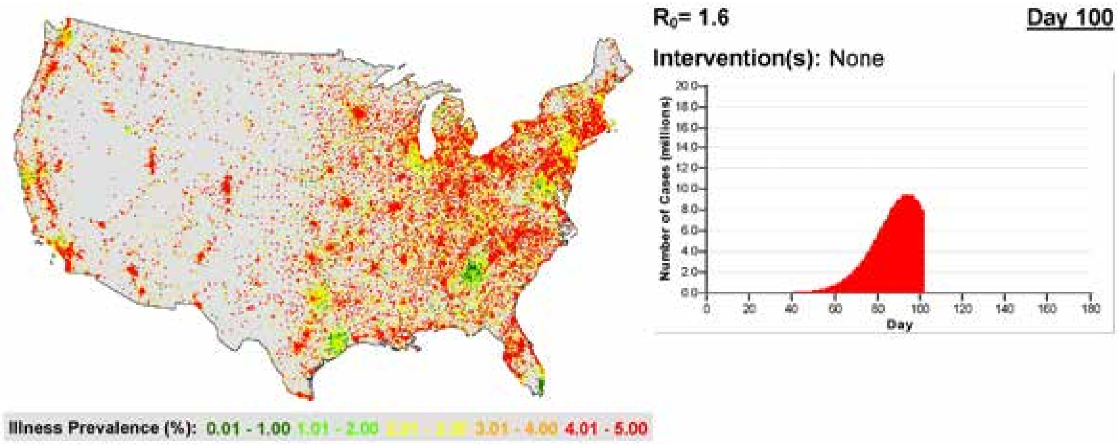


Fig. 2. Distribution of the virus within the territory of the United States according to FluTE. The color indicates the spread of the disease

graphs, each of which can correspond, for example, to different types of transport. The built-in editor of a set of compartments and connections between them allow one to develop custom compartmental models. For analytical purposes, STEM allows one to visualize the process of the spread of an infectious disease on a geographic map of the modeling area.

The Framework for Reconstructing Epidemic Dynamics (FRED) [12, p. 1] uses an agent-based model to simulate the spread of infectious diseases. Similar to FluTE, this method is characterized by deep modeling of the sociodemographic structure of the population, taking into account such features of each agent as gender, age, race, employment, etc., to use them in the modeling of daily movements and the development of the disease. As part of the movements, the agent visits certain institutions on a daily basis, where he comes into contact with other agents, contributing to the spread of the disease. And although the user does not have the ability to edit existing and add new types of these institutions, FRED allows you to configure the rate of infection of agents for each of them.

General-purpose modeling environments

NetLogo [13, p. 1] is modeling environment, consisting of the following elements:

- Subject-oriented programming language, used for describing agents behavior, their inner states evolution and interactions between each other;
- Tool set for the analysis of experiments data as the result of modeling.

Overall, NetLogo allows one to flexibly define models, but this flexibility comes with certain limitations. First of all, as showed in [14, p. 1], the overall performance of NetLogo simulation engine is significantly lower, compared to similar models implemented using other tools. NetLogo features fixed modeling space geometry in the form of rectangular grid, where each agent is "attached" to a certain cell of it and can interact with agent on adjacent cells. Despite the interaction of agents positioned far from each other can be implemented, it is not supported internally, can be cumbersome and limits computational performance of a simulation.

RePast [15, p. 1] is a software tool set, designed for modeling of complex systems. There are several implementations of it, based on different programming languages such as Java, C++. Repast also features an implementation, suitable for high-parallel environments [16, p. 1215].

Main research materials. CTrace [17, p. 426] is a domain specific programming language for an agent-based epidemiological modeling. It offers simple syntax together with powerful semantics, allowing one to research, develop and analyze epidemiological models quickly without requiring strong knowledge in programming and computer science. Its compiler is designed in a way to output Python module as translation artifact, which exposes a programming interface to corresponding epidemiological model.

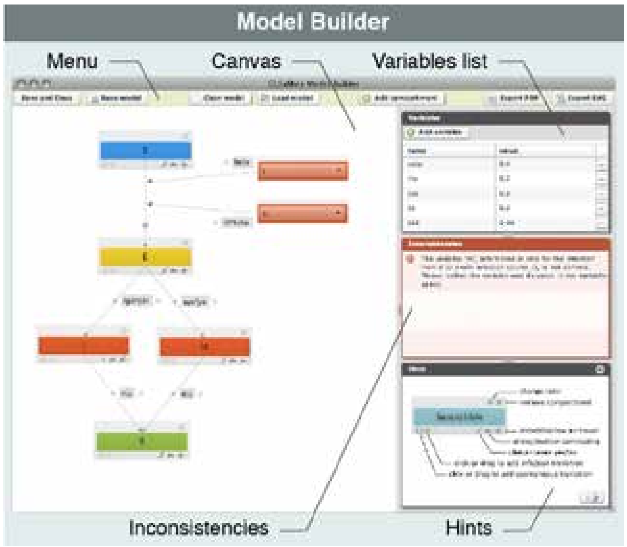


Fig. 3. User interface of the tool for building a compartmental model by specifying compartments, relationships between them and parameters in the GLEaMvis modeling environment

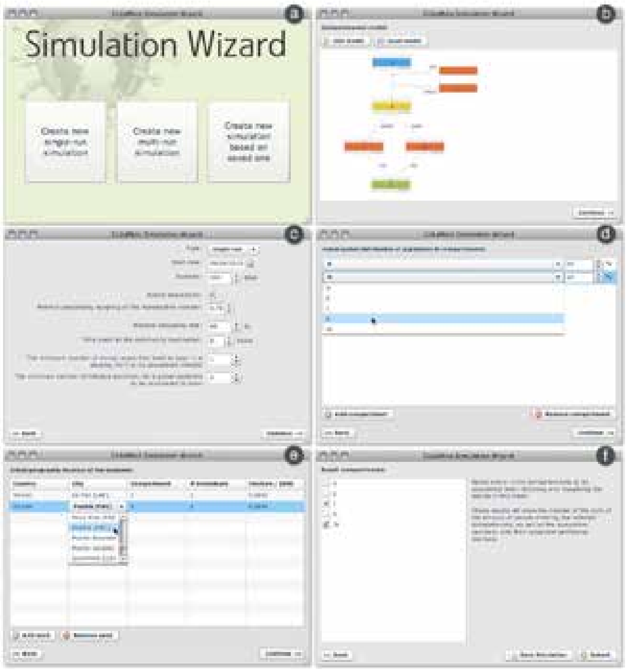

a
b
c
d
e
f
Simulation Wizard
Create new single-run simulation
Create new multi-run simulation
Create new simulation based on saved one

Fig. 4. The user interface of the simulation setup tool in the GLEaMvis environment. a - simulation type selection window; b - compartment model selection window; c - window for setting simulation parameters; d - window for setting the initial size of the compartments; e - window for configuring the geography of the epidemic; f - a window for selecting compartments for presentation in analytics

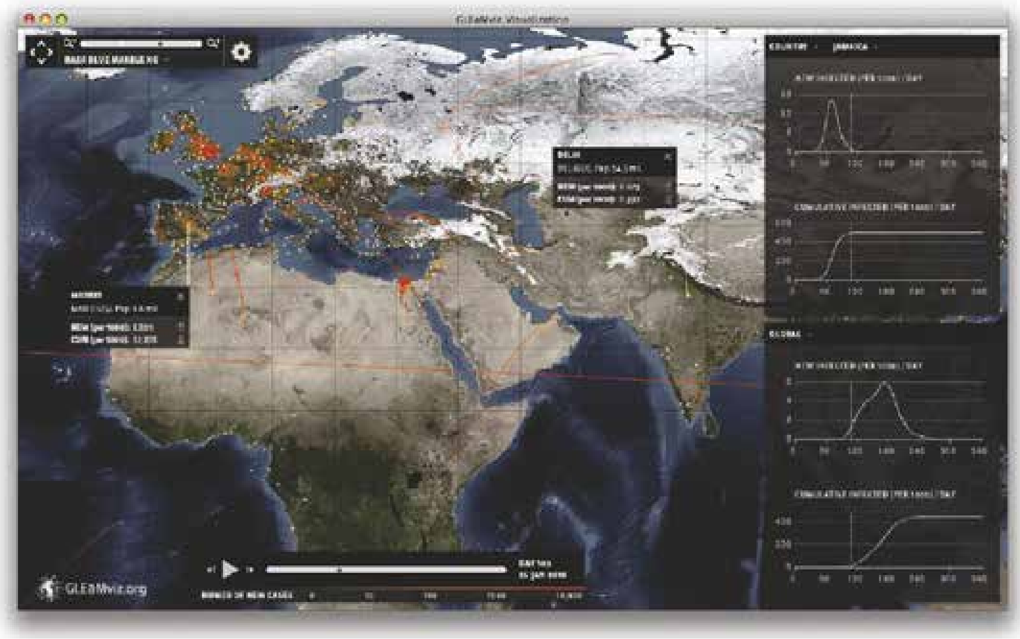
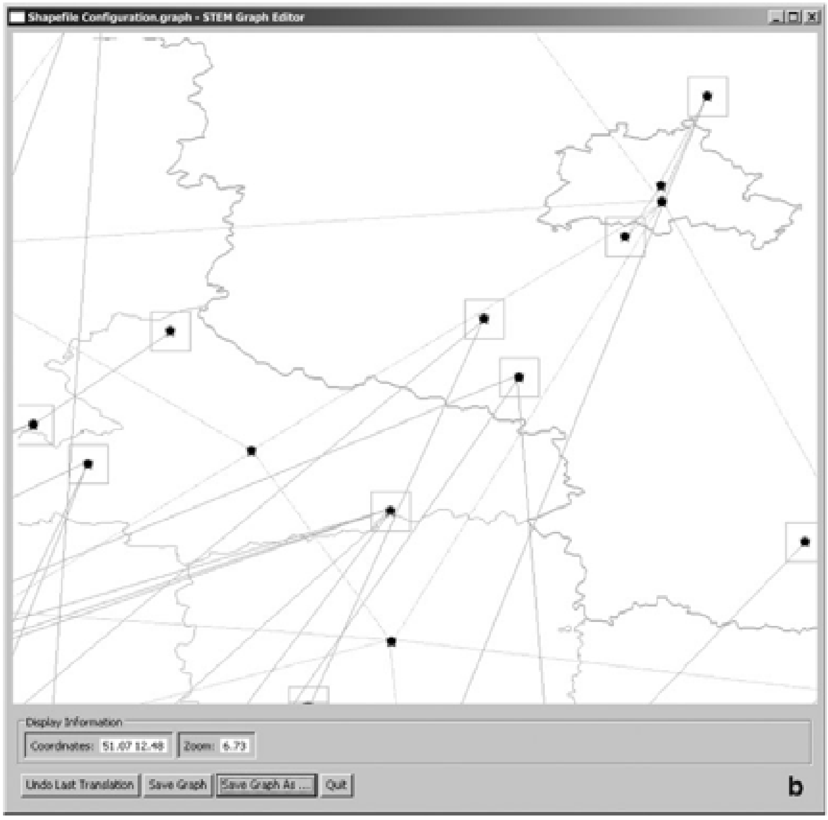


Fig. 5. The user interface of the analytics tool in the GLEaMvis simulation environment



Coordinates: 51.07 12.48

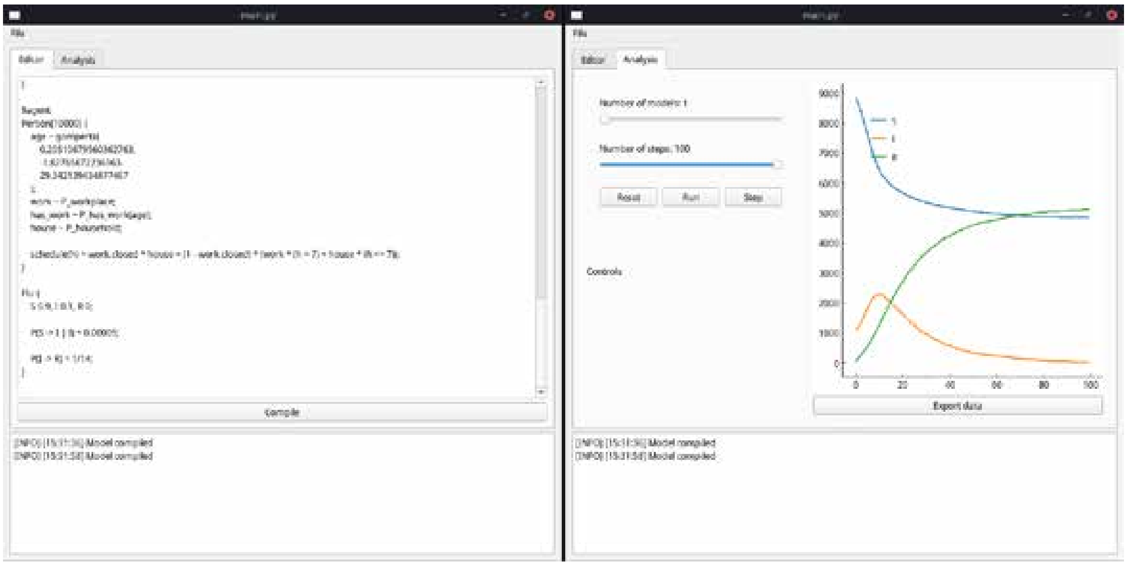
Zoom: 6.73

Fig. 6. The user interface of the graph editing tool in the STEM simulation environment

These design choices allowed us to develop an epidemiological modeling environment CTraceEnv which provides necessary tools for the development and analysis of such models.

The user interface of CTraceEnv consists of two main sections: model editor and model analysis (Fig. 7).

Model editor allows one to describe agent-based epidemiological model using CTrace language. It



```
}

%agent
Person[10000] {
  age ~ gompertz(
    0.20510479560342763,
     1.6270147273756563,
    29.342539603487467
  );
  work ~ P_workplace;
  has_work ~ P_has_work(age);
  house ~ P_household;

  schedule(t) = work.closed * house + (1 - work.closed) * (work * (t < 7) + house * (t >= 7));
}

Flu {
  S(.9), I(.1), R(0);

  P(S -> I | I) = 0.00005;

  P(I -> R) = 1/14;
}
```

```
[INFO] [15:11:36] Model compiled
[INFO] [15:21:58] Model compiled
```

Fig. 7. CTraceEnv application interface. Left column - model editor section, right column - model analysis section

offers compilation functionality together with error logging interface.

Model analysis window itself consists of two columns: simulation controls and epidemiological dynamics graph. Simulation controls allow one to control number of model iterations as well as number of parallel models. If model contains global variables, controls for changing their values will be displayed in the corresponding section. Dynamics graph offers simple visualization of evolution of number of agents with each infection state. In the case, when multiple models are run in parallel, the graph displays average, 25th and 75th percentile of corresponding values. To support more complex analysis of developed models, CTraceEnv offers interface for data export in CSV format.

CTraceEnv environment is build using PyQT graphical user interface library for Python programming language to ensure cross-platform capabilities.

Conclusions. The purpose of this paper was to introduce CTraceEnv, an epidemiological modeling environment that is built on top of the CTrace modeling language. Our aim was to create a user-friendly tool that enables researchers and epidemiologists to efficiently develop and analyze epidemiological models without the need for advanced programming or computer science skills.

By offering a simple and accessible tool, CTraceEnv provides a powerful platform for developing and analyzing complex epidemiological models. It is our hope that this tool will enable researchers to gain a better understanding of the dynamics of infectious diseases, which will ultimately lead to improved public health outcomes.

Bibliography:

1. Cyrus H Gordon. “The new Amarna tablets”. In: Orientalia 16.1 (1947), pp. 1–21.
2. Suzanne Austin Alchon. A pest in the land: new world epidemics in a global perspective. UNM Press, 2003, p. 21.
3. James P Warbasse. “Anomalies and Curiosities of Medicine”. In: Annals of Surgery 26.1 (1897), p. 150.
4. Coronaviridae Study Group of the International Committee on Taxonomy of Viruses et al. “The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2”. In: Nature microbiology 5.4 (2020), p. 536. doi: 10.1038/s41564-020-0695-z.
5. Raj Dandekar and George Barbastathis. “Quantifying the effect of quarantine control in Covid-19 infectious spread using machine learning”. In: medRxiv (2020).
6. Mai Anh BUI T et al. “The Kendrick modelling platform: language abstractions and tools for epidemiology”. In: BMC bioinformatics 20.1 (2019), pp. 1–13.
7. Swarm. 2021. url: http://www.swarm.org/wiki/Main Page.
8. Padmavathi Patlolla et al. “Agent-based simulation tools in computational epidemiology”. In: International Workshop on Innovative Internet Community Systems. Springer. 2004, pp. 212–223.
9. Dennis L Chao et al. “FluTE, a publicly available stochastic influenza epidemic simulation model”. In: PLoS computational biology 6.1 (2010).

10. Wouter Van den Broeck et al. “The GLEaMviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale”. In: BMC infectious diseases 11.1 (2011), pp. 1–14.

11. Alexander Falenski et al. “A generic open-source software framework supporting scenario simulations in bioterrorist crises”. In: Biosecurity and bioterrorism: biodefense strategy, practice, and science 11.S1 (2013), S134-S145.

12. John J Grefenstette et al. “FRED (A Framework for Reconstructing Epidemic Dynamics): an open-source software system for modeling infectious diseases and control strategies using census-based populations”. In: BMC public health 13.1 (2013), pp. 1–14.

13. Uri Wilensky. NetLogo 6.2.2 User Manual. 2022. url: https://ccl.northwestern.edu/netlogo/docs/.

14. KK Avilov and O Yu Solovey. “Institute of numerical mathematics of Russian Academy of Sciences Moscow Institute of Physics and Technology (State University)”.

15. Repast. 2021. url: https://repast.github.io/.

16. Nicholson Collier and Michael North. “Parallel agent-based simulation with repast for high performance computing”. In: Simulation 89.10 (2013), pp. 1215–1235. doi: 10.1177/0037549712462620.

17. Vladyslav Sarnatskyi and Igor Baklan. “CTrace: Language for Definition of Epidemiological Models with Contact-Tracing Transmission”. In: International Scientific Conference “Intellectual Systems of Decision Making and Problem of Computational Intelligence”. Springer. 2022, pp. 426–448.

Сарнацький В.В., Баклан І.В. CTRACEENV: ПЛАТФОРМА РОЗРОБКИ ТА АНАЛІЗУ АГЕНТНИХ ЕПІДЕМІОЛОГІЧНИХ МОДЕЛЕЙ З ВИКОРИСТАННЯМ МОВИ CTRACE

За останні кілька років епідеміологічне моделювання стає все більш значущим аспектом контролю та боротьби з інфекційними захворюваннями. Одним з найбільш перспективних підходів до моделювання динаміки інфекційних захворювань є агентні моделі (АМ), які пропонують детальний і гнучкий інструментарій для імітації передачі хвороби у середовищі населення. У цьому дослідженні ми описуємо наші зусилля по створенню середовища, яке спрощує розробку та аналіз АМ для цілей епідеміологічного моделювання.

Наш підхід базується на використанні мови моделювання CTrace, яка пропонує інтуїтивно зрозумілий синтаксис для визначення АМ. За допомогою зручних інструментів цієї мови користувачі можуть визначити окремих агентів у популяції, їх взаємодію і те, як хвороба поширюється серед них. Крім того, компілятор мови здатний генерувати програмні бібліотеки, які ми успішно використали в розробленому нами середовищі.

Щоб полегшити розробку АМ за допомогою CTrace, ми розробили середовище, яке полегшує інтеграцію мови із зовнішніми інструментами. Середовище включає інтуїтивно зрозумілий інтерфейс, який проводить користувачів через процес визначення моделі та запуску симуляцій. Крім того, однією з ключових особливостей середовища є його здатність виконувати базовий аналіз моделі, наприклад, відображати статистику епідемії в часі. Потім цю інформацію можна експортувати для подальшого аналізу за допомогою зовнішніх інструментів, таких як R або Python.

Наше дослідження підкреслює важливість створення зручних для користувача середовищ, які спрощують розробку та аналіз АМ для епідеміологічного моделювання. Використання мови моделювання CTrace та розробленого середовища пропонує потужний та ефективний інструмент для дослідників для моделювання передачі захворювань та дослідження ефективності різних стратегій контролю. Таким чином, наш підхід може виявитися корисним у розробці більш точних та ефективних моделей для управління інфекційними захворюваннями.

Ключові слова: *епідеміологія, моделювання, агентне моделювання, програмне забезпечення, середовище моделювання.*