

латентні представлення емоцій є частково мовно-інваріантними, що забезпечує ефективне перенесення знань.

Оцінка якості синтезованого мовлення визначається за об'єктивними метриками (мел-кепстральне спотворення (MCD), частка помилково синтезованих слів (WER), класифікація емоційного мовлення моделями глибокого навчання) та суб'єктивними (усерелнена суб'єктивна якісна оцінка (MOS) з 5 оцінювачами, суб'єктивний тест на розпізнавання емоцій).

Попередні експерименти показують, що адаптована модель здатна генерувати природне українське мовлення з MOS > 3.5. Очікується, що механізм контролю інтенсивності забезпечить плавний перехід між нейтральним та емоційними станами з точністю розпізнавання емоцій > 75%. LoRA адаптери демонструють ефективність у використанні пам'яті при збереженні якості емоційної виразності, порівнянної з повним донавчанням.

Запропонований підхід надає практичне рішення для розробки емоційно виразних систем синтезу мовлення для малоресурсних мов. Комбінація крос-мовного перенесення та LoRA шарів дозволяє ефективно адаптувати актуальні багатомовні моделі до української мови з контрольованою емоційною інтенсивністю. Подальші дослідження будуть зосереджені на розширенні набору емоцій, покращенні якості крос-мовного перенесення знань у моделях машинного навчання та удосконаленні підходу до моделювання емоційної інтенсивності.

Список використаних джерел

1. Ukrainian TTS using ESPnet framework // GitHub repository. 2021. URL: <https://github.com/robinhad/ukrainian-tts>.
2. Kim J., Kong J., Son J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech // Proceedings of the 38th International Conference on Machine Learning (ICML). 2021. P. 5530-5540.
3. Casanova E., Weber J., Shulby C.D. et al. YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone // Proceedings of the 39th International Conference on Machine Learning (ICML). 2022. P. 2709-2720.
4. Casanova E. et al. XTTS: a Massively Multilingual Zero-Shot Text-to-Speech Model // arXiv preprint arXiv:2406.04904. 2024.
5. Hu E.J., Shen Y., Wallis P. et al. LoRA: Low-Rank Adaptation of Large Language Models // International Conference on Learning Representations (ICLR). 2022.

ВДОСКОНАЛЕННЯ ВІЗУАЛІЗАЦІЇ МОДЕЛЕЙ АРХІТЕКТУР ПРОГРАМ ДЛЯ ЗМЕНШЕННЯ КОГНІТИВНОГО НАВАНТАЖЕННЯ КОРИСТУВАЧА / IMPROVING VISUALIZATION OF SOFTWARE ARCHITECTURE MODELS TO REDUCE USER COGNITIVE LOAD

Франків О.О. / Frankiv O.O.

Національний університет «Києво-Могилянська академія» / National University of Kyiv-Mohyla Academy

04070, м. Київ, вул. Г. Сковороди 2, каф. інформатики, тел. (044) 425 60 64

E-mail: o.frankiv@ukma.edu.ua, тел. (044) 425 60 64

The main priority in developing automated visualization tools is to provide a clear and easily interpretable experience for the user. When visualizing complex and abstract entities such as software architecture, reducing the cognitive effort required to perceive information is essential, allowing the user to focus on the actual domain. In this paper, we examine practical

approaches to improving the UX and perceptual quality of the ADAR software-architecture visualization tool. Although graph-based models are intuitively suitable for representing connections, the quality of graph layout can be crucial for meaningful comprehension. At the same time, recent research shows that commonly used layout metrics may be misleading. From this perspective, user studies offer a more reliable basis for quality assessment. Furthermore, targeted improvements to the tool itself are expected to reduce cognitive load and enhance the overall usability.

На сьогодні проблема постійного аудиту та своєчасного виявлення вад програм під час розроблення залишається актуальною. Хоча для виявлення відносно простих проблем існує великий набір інструментів, складним завданням залишається розроблення нових інструментів, що дозволяють легко аналізувати архітектуру програми та виявити вади на цьому рівні. Додатковим викликом є природня потреба у мінімізації додаткових затрат ресурсів на використання відповідних інструментів.

Раніше запропонований програмний комплекс ADAR – Architecture Displayer in Augmented Reality – дозволяє автоматично будувати візуальну модель програм написаних мовою Swift у формі зваженого орієнтованого графу та відображати її у доповненій реальності (AR) за допомогою спеціального застосунку [1]. У графовій моделі типи даних (класи), їхні поля та методи відображаються у вигляді вузлів, а зв'язки між ними у вигляді направлених зважених ребер, що позначають напрям та інтенсивність.

Очевидним є те, що для достатньо великих програмних модулів графова модель буде налічувати велику кількість вузлів та ребер. Проблема швидкодії в даному контексті є зрозумілою і її можна вирішувати завдяки інтелектуальним оптимізаціям [2]. Однак значно складнішими є питання якості подання та забезпечення легкого сприйняття, що є критичними для будь-якої візуалізації, адже саме зменшення когнітивного навантаження при сприйнятті інформації є визначальним фактором.

Візуальна естетичність розміщення графа в просторі природньо визначає складність його сприйняття. На сьогодні існує велика кількість критеріїв естетичності та метрик, що дозволяють чисельно визначити якість подання. Більшість таких критеріїв, як-от мінімізація накладання вузлів чи перетинів ребер є інтуїтивно зрозумілими і очевидно ведуть до зменшення когнітивного навантаження користувача. Задоволення цих базових критеріїв не гарантує ідеального подання, тому існує також велика кількість метрик, покращення яких додатково роблять розміщення естетичнішим. Однак нещодавні дослідження показують, що часто використання багатьох метрик одночасно може бути суперечливим, хоча кожна окремо має сенс [3]. Окрім цього саме формальне задоволення набору критеріїв та покращення метрик не є достатнім, адже було проілюстровано, що подання графів можуть формально цілком відповідати критеріям і виглядати привабливо з точки зору визначення метрик, однак бути складними для сприйняття людиною. З огляду на це можна припустити, що оскільки збільшення кількості метрик та критеріїв не обов'язково гарантують створення простого для сприйняття подання, балансом між швидкодією та якістю візуалізації залишається застосування класичних алгоритмів, наприклад силового.

З іншого боку хоча результат виконання силового алгоритму для великих графів дозволяє отримати привабливе відображення в просторі, це не обов'язково означає простоту для розбору та інтерпретації даних. Тому ортогональним підходом до зменшення когнітивного навантаження є вдосконалення самого інструменту, що відображає певне подання графу.

Створення якісного користувацького досвіду, що буде нести мінімальне когнітивне навантаження, це вагома частина розроблення будь-якого інструменту. На сьогодні існує багато досліджень, що пропонують певні метрики для числового визначення інтуїтивності інтерфейсу та подачі інформації загалом [4]. Позитивні показники частини з них може забезпечити загальний підхід до візуалізації, для інших — створення гнучких налаштувань чи інструментів взаємодії.

Читабельність подання (layout readability) визначають за допомогою метрик співвідношення фону (white space ratio) та співвідношення перекриття міток (label overlap ratio). Відображення моделі архітектури в просторі за допомогою доповненої реальності природнім чином збільшує частку фону на зображенні, що може свідчити про хорошу композицію. З іншого боку надання користувачу можливості повертати та масштабувати модель в просторі, а також показувати і ховати описові мітки дозволяє не лише мінімізувати к-сть міток і, як наслідок, їхніх перекриттів, але й чітко контролювати кількість наданої інформації.

Кількість перекриттів (occlusion rate) визначає рівень просторового сприйняття. Класичний силовий алгоритм забезпечує мінімізацію, але не повне усунення перекриттів віртуальних об'єктів, особливо з урахуванням того, що де-факто користувач сприймає тривимірну модель як двовимірну проекцію. Однак можливість легко взаємодіяти з моделлю дозволяє користувачу змінювати цю проекцію залежно від своїх потреб і конкретної моделі, таким чином гнучко мінімізуючи кількість перекриттів у зоні інтересу.

Проблема використання кольорів є важливим фактором в оцінці складності сприйняття [4][5]. До прикладу семантична метрика “складність легенди” вираховується за кількістю унікальних кольорів, що використовуються на схемі. Для зменшення навантаження на користувача при пошуку вад проектування в інструменті візуалізації передбачено окремий біколірний режим. Таким чином всі елементи моделі є нейтральними, а ті, в яких система виявила вади, позначаються контрастним кольором, на відміну від звичайного режиму з повною інформацією, для відображення якої необхідно щонайменше десять кольорів.

Ще однією метрикою, що визначає легкість сприйняття є співвідношення контрастів (contrast ratio). Особливість її застосування полягає в тому, що співвідношення має бути в певному “достатньому” діапазоні залежно від контексту та тривалості використання інтерфейсу. Забезпечення правильного контрасту є комплексною проблемою, особливо в доповненій реальності, де освітлення середовища сильно впливає на контрастність доповненого зображення. Для забезпечення максимально комфортного сприйняття в програмі передбачено автоматичне вирівнювання контрасту віртуальних об'єктів з реальним середовищем.

Однак насамкінець варто зазначити, що будь-які метрики є спробою формально визначити результати емпіричних досліджень на реальних користувачах і допускають певні неточності зумовлені відмінностями між об'єктами досліджень. Тому більш реальну та якісну оцінку рівня когнітивного навантаження можна отримати саме з використанням відповідних замірів отриманих в результаті емпіричних досліджень для конкретної системи.

Список джерел:

1. Франків О. О., Глибовець М. М. Автоматизована візуалізація компонентів архітектури програми для мови Swift. *Кібернетика та системний аналіз*. 2024. Т. 60, вип. 6. С. 190–198. URL: <https://doi.org/10.1007/s10559-024-00737-9>.
2. Frankiv O. Machine learning in enhancing visualization of the spatial software architecture model. *Bulletin of Taras Shevchenko National University of Kyiv. Physical and Mathematical Sciences*. 2025. Т. 80, вип. 1. С. 164–173. URL: <https://doi.org/10.17721/1812-5409.2025/1.22>.
3. Wageningen S., Mchedlidze T., Telea A. C. Same Quality Metrics, Different Graph Drawings. *International Graph Drawing Conference 2025*. URL: <https://doi.org/10.48550/arXiv.2508.15557>.
4. A Survey on Measuring Cognitive Workload in Human-Computer Interaction / T. Kosch та ін. *ACM Computing Surveys*. 2023. Т. 55. URL: <https://doi.org/10.1145/3582272>.

5. Zhang R., Son O.-S. A Measurement Model for Visual Complexity in HCI: Focusing on Visual Elements in Mobile GUI Design. *Electronics*. 2025. Т. 14, вип. 5. С. 942. URL: <https://doi.org/10.3390/electronics14050942>.

ПРИСКОРЕНЕ МОДЕЛЮВАННЯ ЕПІДЕМІЙ У ВЕЛИКИХ МЕРЕЖАХ: ОПТИМІЗАЦІЯ МЕТОДУ ГІЛЛЕСПІ ТА ДВОШАРОВИЙ ПІДХІД / ACCELERATED EPIDEMIC SIMULATION IN LARGE-SCALE NETWORKS: OPTIMIZATION OF THE GILLESPIE ALGORITHM AND A TWO-LAYER APPROACH

Куриляк Ю.А.¹, Еммеріх М.Т.М.² / Kuryliak Y.A.¹, Emmerich M.T.M.²

¹Національний університет «Львівська політехніка» / Lviv Polytechnic National University

вулиця Степана Бандери, 12, Львів, Львівська область, 79000, Україна

²Університет Юваскуля / University of Jyväskylä

Agora, Mattilanniemi 2, 40100 Jyväskylä, Jyväskylä, 40014, Finland

E-mail: ¹yulian.a.kuryliak@lpnu.ua, ²michael.t.m.emmerich@jyu.fi

Abstract: This study addresses the challenge of accelerating epidemic simulations in large-scale complex networks through algorithmic and structural optimization. Traditional Gillespie-based stochastic simulations accurately reproduce epidemic dynamics but become computationally prohibitive for networks exceeding tens of thousands of nodes. To overcome this limitation, we build upon such efficiency techniques as local rate updates and ordered event-selection structures, which reduce the computational complexity of each simulation step from $O(n)$ to $O(\log(n))$. Building on these principles, we propose a two-layer (micro–macro) modeling framework: the micro layer simulates intra-community dynamics, while the macro layer captures inter-community infections using hazard-integral rates derived from mobility data and the epidemic states of metanodes. This hierarchical approach enables scalable and parallelizable simulations that preserve stochastic accuracy while substantially reducing computational cost, allowing realistic modeling of epidemic spread across millions of agents and multiple cities.

Epidemics have accompanied humanity throughout history. Each time a new wave of disease emerges, societies seek to understand how it spreads and what can be done to contain it. Historically, epidemic dynamics were modeled using differential equations based on the mean-field assumption — treating the population as a homogeneous system without accounting for its contact structure. Today, with increasing computational power and advances in simulation technology, it has become possible to reproduce infection dynamics in realistic, irregular contact networks that reflect individual-level interactions.

Contact networks are sparse, heterogeneous in degree, and exhibit strong local clustering and a pronounced community structure — groups of nodes densely connected within communities but weakly linked to others. Therefore, accurate modeling requires capturing explicit node-to-node connections and their temporal state transitions. Agent-based modeling achieves this by representing each node as an individual whose contacts define potential infection paths.

Such processes can be formulated as Continuous-Time Markov Chains (CTMC). To simulate them efficiently, the Gillespie algorithm provides an exact stochastic representation of system dynamics. Its key principles are: (I) only feasible transitions from the current state are considered; (II) only one event occurs at a time; and (III) the time to the next event follows an exponential distribution, after which the event is selected proportionally to its rate. This ensures accuracy and avoids state-space explosion, but computational complexity still limits the model size. As shown in [1], the classical Gillespie algorithm can simulate up to several tens of thousands of nodes, which remains insufficient for realistic population scales. For comparison, modern agent-based frameworks such as Covasim [2] implement stochastic transitions in discrete time with a daily step, which simplifies computations but sacrifices temporal precision.