



Є. В. Левус, Ю. О. Гучок, О. О. Нитребич

Національний університет "Львівська політехніка", м. Львів, Україна

АРХІТЕКТУРНЕ РІШЕННЯ ДЛЯ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ ІНТЕРФЕЙСІВ КОРИСТУВАЧА

Розглянуто проблему побудови інтелектуальних інтерфейсів користувача (IUI). Хоча дослідження в галузі IUI проводять із часів перших розробок штучного інтелекту, на сучасному етапі розвитку інженерії програмного забезпечення існує небагато успішних рішень. Проаналізовано найвідоміші приклади реалізації IUI – проекти компанії Microsoft: The Office Assistant та Lumiere Project. Узагальнено недоліки цих реалізацій. Актуальним є розроблення концепції програмного забезпечення на підставі підходу "мінімальної взаємодії", який охоплює використання інтелектуальних інтерфейсів. Важливою проблемою IUI, яка насамперед потребує вимагає вирішення, є порушення встановлених Usability-принципів для систем із класичним інтерфейсом. Допоміжним методом "пом'якшення" наслідків від порушення принципів до побудови інтерфейсів є чітке розділення адаптивного інтерфейсу і класичного. Як рішення запропоновано використовувати штучний інтелект у вигляді віддалених баєсових мереж; хмарних технологій для використання баєсових мереж у локальному ПЗ наживо та поповнення мережі знаннями, отриманих від окремих локальних користувачів; спілкування між мережею та локальним ПЗ за допомогою API. Метою розробленого програмного продукту є створення гнучкої та спрощеної системи високого рівня для імплементації аспектів IUI в ПЗ інших розробників. Цю мету реалізовано у вигляді REST API сервісу, який знаходиться на віддаленому сервері та призначений для побудови баєсових мереж. Перспективою розвитку представленого методу є врахування інших особливостей користувачів уже на рівні більш глибокого штучного інтелекту і передбачення можливих дій з погляду не тільки статистики, але й логіки.

Ключові слова: програмне забезпечення; класичний і адаптивний інтерфейси користувача; баєсова мережа; REST API сервіс.

Вступ. З огляду на постійне розширення функціональних можливостей програмного забезпечення (ПЗ), традиційні візуальні інтерфейси користувача є переважними та складними у використанні. Альтернативою спрощення використання комп'ютерних систем є інтелектуальні інтерфейси користувача (англ. IUI), які забезпечують персоналізацію інтерфейсів. Незважаючи на те, що дослідження IUI ведуть ще з часів перших розробок у галузі штучного інтелекту, сьогодні існує небагато програмних рішень з IUI. Багато користувачів стикаються з проблемами використання ПЗ, більшість з них – завдяки недосконалим користувацьким інтерфейсам. Такі тенденції зумовлюють потребу в удосконаленні людино-машинних інтерфейсів, які, в ідеалі, розуміють потреби людей, можуть їм допомагати і пояснювати використання всього арсеналу функцій програми.

Відомий дослідник у галузі Ерік Хорвітц (англ. *Eric Horvitz*), керівник одного з найбільш вдалих проектів із використання IUI – Lumiere Project (компанія Microsoft), звертає увагу на те, що велика частина проблем у розробленні IUI – це постійне стикання комп'ютерної

системи з невизначеностями про мотиви і потреби користувачів (Horvitz, et al., 2018). Він зазначає, що використання методів машинного мислення і логічних висновків може значно збільшити можливості подолання цієї проблеми.

Вагомий внесок у розвиток IUI, запропонувавши три наріжних камені до побудови IUI, зробив Генрі Лайбермен (англ. Henry Lieberman) з медіалабораторії Массачусетського Технологічного Інституту (Birnbbaum et al., 1997):

- Ніколи не переривайте взаємодію користувача і системи. Проте користувач завжди повинен мати можливість ігнорувати агента.
- Проводьте обчислення в реальному часі, а найкраще – в моменти, коли користувач "зайнятий".
- Завжди спостерігайте за користувачем і користуйтеся "вільною інформацією", яку можна отримати з користувацьких дій.

Фахівець вважає, що штучний інтелект у UI доцільно використовувати для створення "асистентів" до ПЗ, а не для всесильної сутності, яка вирішує проблеми. Наприклад: у ситуації, коли перед користувачем постає складний вибір (Laviers, 2011).

Інформація про авторів:

Левус Євгенія Василівна, канд. техн. наук, доцент, кафедра програмного забезпечення. **Email:** yelevus@gmail.com

Гучок Юрій Олегович, магістрант, кафедра програмного забезпечення. **Email:** yurajatrix@gmail.com

Нитребич Оксана Олександрівна, канд. техн. наук, ст. викладач, кафедра програмного забезпечення.

Email: ksenija.volynj@gmail.com

Цитування за ДСТУ: Левус Є. В., Гучок Ю. О., Нитребич О. О. Архітектурне рішення для побудови інтелектуальних інтерфейсів користувача. Науковий вісник НЛТУ України. 2018, т. 28, № 8. С. 155–160.

Citation APA: Levus, Ye. V., Huchok, Yu. O., & Nytrebych, O. O. (2018). Architectural solution for intellectual user interface creation. *Scientific Bulletin of UNFU*, 28(8), 155–160. <https://doi.org/10.15421/40280831>

До найпопулярнішого ПЗ, що використовує IUI, можна віднести персонального асистента The Office Assistant, який працював у тандемі з ПЗ Microsoft Office у кінці 90-х років. The Office Assistant (відомий як Clippy) широко базується на попередній розробці The Lumiere Project (Microsoft).

Окрім небагатьох комерційних розробок, існує багато внутрішніх досліджень від університетів та компаній, які супроводжуються прототипами ПЗ із використанням IUI. Одним із них є, вже зазначений, The Lumiere Project від Microsoft Research. Дуже популярними є такі сучасні розробки, як: Siri, Google Now, Cortana. Всі вони діють за схожими до IUI принципами, хоча насправді інтелектуальний інтерфейс повною мірою вони не використовують.

Наявні рішення: особливості та недоліки. Враховуючи тенденції та недоліки існуючих програм, актуальною проблемою є розроблення концепції ПЗ на підставі підходу "мінімальної взаємодії" з використанням IUI. Ідея полягає в реалізації максимальної ефективності за якнайменшої кількості вхідних даних. Користувачів необхідно звільнити від потреби приділяти багато часу на менеджмент самого ПЗ та дати їм можливість втручатись тільки в речі, важливі саме їм.

Важливо зафіксувати причини неуспішності розробок IUI (рис. 1), які описали фахівці, зокрема Девід Курландер (David Kurlander) з Microsoft Research (Birnbbaum et al., 1997).

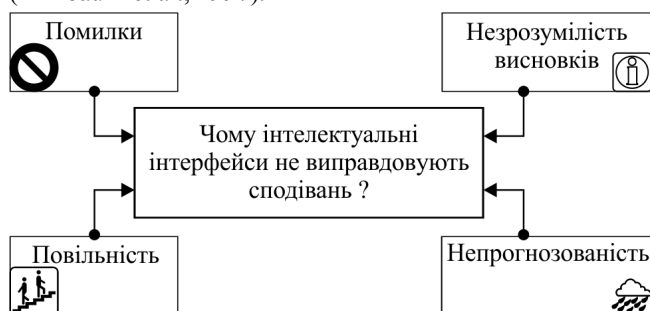


Рис. 1. Основні причини неуспішності IUI

У своїй публікації Джуді Кей (англ. Judy Kay) пропонує надавати користувачам способи інспектувати і змінювати свою ж модель користувача – цим самим надавати контроль над адаптацією (Cook, 1994). Залежно від цільової аудиторії ПЗ така мета може бути складною в реалізації. Система адаптивних підказок, яку описує Томас Кумм (нім. Thomas Kühme), може бути прикладом цього рішення. Користуючись нею, користувач має змогу "підганяти" механізми адаптації під себе, але для цього йому потрібно вивчити багато термінів: ціль, дія, взаємодія тощо. Завдяки такій концепції користувач може самостійно конструювати свої правила адаптивності (Benyon, 1993). Дозвіл користувачам змінювати свою модель створює додаткові проблеми. Одна з них – відволікання від функціоналу і примусова втрата часу на менеджмент ПЗ. Цей підхід може бути ефективним, якщо мова йде про індустріальні вимоги до адаптації, коли налаштування ПЗ протягом багатьох тижнів є допустимими, а для широкої аудиторії користувачів рішення є несприятливим.

У літературі можна побачити такі настанови до проектування IUI (Rohrbach & Schmidt, 2017):

- Користувацький інтерфейс повинен розроблятися паралельно з "бек-ендом" програмного забезпечення.

- Завжди ставити користувача вище за цілі системи – тільки надавати пропозиції, ніколи не діяти власноруч.
- Виконувати роботу в режимі реального часу. Найактуальніший час роботи IUI – під час активної роботи користувача.
- Завжди використовувати вільний час для опрацювання даних, наприклад у перерві між роботою користувача.
- Активно спостерігати за діями користувача. Іншими словами, повністю використовувати наявні вхідні дані.
- Давати змогу користувачеві самому обирати тип спілкування інтелектуального інтерфейсу.

Важливою проблемою IUI, яка насамперед потребує, вимагає вирішення, є порушення встановлених Usability-принципів для систем із прямим керуванням – класичних інтерфейсів (Höök, 2017).

Програма, яка базується на IUI, в будь-якому випадку повинна містити частину інтерфейсу, яку можна вважати "класичною". IUI краще працюють у парі з традиційними, а традиційна компонента має бути модифікована для підтримки інтелектуальної частини. В іншому випадку, інтелектуальний інтерфейс, як окрема сутність застосування, тільки робить використання важчим. Системи, які мають на меті адаптацію до користувачів і постійну зміну поведінки, потребують за своїм визначенням порушувати ці принципи. У галузі IUI було запропоновано кілька підходів для вирішення цієї проблеми.

Хорошим рішенням, яке дає змогу користувачам надавати контроль над адаптацією інтерфейсу, є розроблення адаптивної частини програми паралельно з функціоналом. Такий підхід допомагає будувати інтерфейси з чітко визначеними способами для корекції активних адаптацій.

Допоміжним методом "пом'якшення" наслідків від порушення принципів до побудови інтерфейсів є чітке розділення адаптивного інтерфейсу і класичного. У таких випадках частини інтерфейсу не мають права впливати одна на одну. Тобто, користувач має можливість постійного вибору між персоналізованими даними і конкретним функціоналом класичного інтерфейсу (рис. 2).

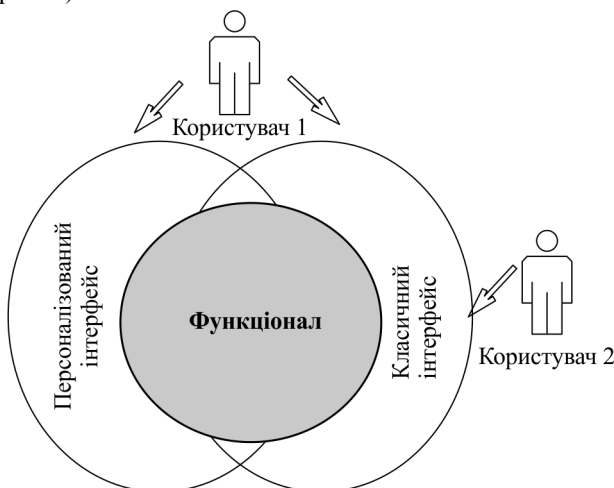


Рис. 2. Структура ПЗ з IUI: користувач 1 використовує IUI; користувач 2 використовує тільки класичний UI

Яскравим прикладом такого використання є ПЗ під назвою Letezia від Microsoft. Система постійно моніторить веб-сторінки і надає персоналізовані пропозиції для користувачів у вигляді посилань. Користувачі в такий спосіб мають змогу повністю ігнорувати пропозиції.

Найбільш популярним і успішним рішенням у розробці ІUI є використання баєсових мереж у моделюванні користувачів. Прикладами є два проекти компанії Microsoft: The Office Assistant та Lumiere Project.

Внутрішня архітектура The Office Assistant базується на використанні баєсових мереж для надання відповідної допомоги користувачам способом прийняття рішень на підставі користувацьких дій, їх опису та стану застосунку.

Посилаючись на подібний до цього граф (рис. 3) та значення апріорних ймовірностей, які задані в кожному вузлі, програма здатна приймати рішення про наміри користувача за допомогою теореми Баєса й активно сприяти виконанню функцій.

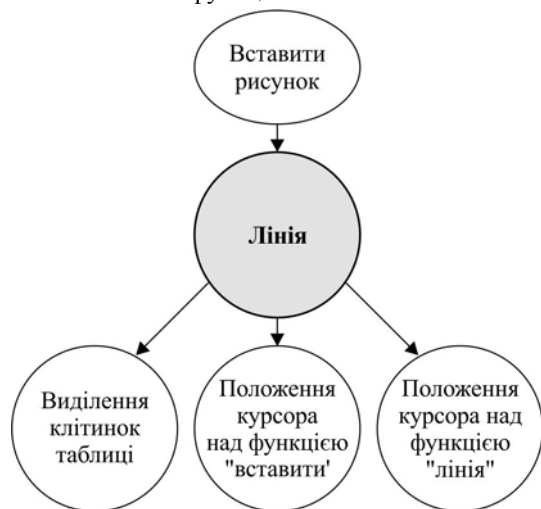


Рис. 3. Частина баєсової мережі програми Clippy

Наприклад, у ситуації, коли користувач виділяє клітинку в Excel та наводить мишку на секцію вибору малюнка, система може визначити, що з 93-відсотковою ймовірністю це має бути лінія, і вибрати лінію заздалегідь, зекономивши час користувача (Horvitz, 1997).

Через декілька років після випуску Clippy отримав багато негативних відгуків і його описували, як "дратівливий". Найбільша критика Clippy висвітлює проблему постійних дій інтерфейсу – програма не зважає на ймовірність того, що користувач взагалі може не потребувати допомоги в певний момент.

Ця програма має такі основні проблеми:

- Система ніколи не бере до уваги модель користувача – тобто система не зважає на дії, виконані користувачем для оцінювання його компетентності, для більш відповідної допомоги. Система працювала зі статичною моделлю користувача, а отже, мала більш "тверду" будову, ніж цього б хотілося.
- Система не використовує комбінацію подій протягом використання програми; натомість, система бере до уваги тільки малий набір, відносно атомарних, дій користувача.
- У системі не було ніяких автоматичних процесів, які б могли активно моніторити поведінку користувача і визначати ступінь "потрібності" допомоги.

The Lumiere Project використовує складну баєсову мережу зі статичною побудовою для репрезентації функціоналу програми. Мережу використовують для активного обчислення ймовірностей намірів використання певних функцій користувачем. Програма постійно робить висновки про потреби і мотиви користувача, беручи до уваги всі активності користувача протягом певного часу за допомогою баєсових мереж. The Lumiere Project також застосовує аналіз ймовірностей слів у запитах користувачів для ідентифікації інформаційних пот-

реб користувачів. Це зводить сумісність з іншим ПЗ до нуля, а інтеграція, в разі необхідності, є складною (Horvitz et al., 2018).

Незважаючи на негативні відгуки, компанія Microsoft стала інноватором використання баєсових мереж в UI. До цих розробок баєсові мережі широко застосовували для гуманітарних цілей: обчислення можливості певних хвороб, несправностей в апаратному забезпеченні та ін. Хоча компанія Microsoft була першою, яка використала цей метод у комерційних цілях, NASA використовувала його для визначення найбільш актуальної інформації для виведення під час польотів.

Як неодноразово зазначали дослідники, використання складних алгоритмів і технологій зовсім не гарантує задовільний результат у плані ІUI.

Пропоноване рішення до реалізації інтелектуальних інтерфейсів

Як таке рішення пропонують:

1. Використання штучного інтелекту у вигляді віддалених баєсових мереж з метою розробки інтелектуального інтерфейсу зі змішаною ініціативою.
2. Використання хмарних технологій для використання баєсових мереж у локальному ПЗ наживо та поповнення мережі знаннями, отриманих від окремих локальних користувачів.
3. Спількування між мережею та локальним ПЗ за допомогою API.

Оскільки баєсові мережі покладаються на статистичну інформацію, використання колективних знань про користувачів зводить помилкові рішення штучного інтелекту до мінімуму (Bernardo & Smith, 1994).

Безпека даних користувачів забезпечується ядром віддаленої системи, яка прямо працює з мережею. Відкриті API запити до локальних ПЗ ніколи не містять персональної інформації, а тільки висновки і рішення інтелектуальної системи.

Для реалізації ІUI у цьому проекті пропонують використовувати баєсові мережі, оскільки їх застосування було найбільш популярним рішенням серед розробників під час активних досліджень. Обрано баєсові мережі з динамічною побудовою, тобто мережі, які б будувалися самостійно з часом використання програми користувачем. Звичайно, це зменшує точність прийняття інтелектуальних рішень за самим визначенням, але водночас дає змогу реалізовувати елементи ІUI за менший період і забезпечує високу сумісність, порівняно з традиційним використанням баєсових мереж зі статичною побудовою.

Системи прийняття рішень на підставі баєсових мереж можуть бути достатньо точними для ПЗ, однак вони забирають багато часу на розробку та вдосконалення.

Метою розроблюваного програмного продукту є створення гнучкої та спрощеної системи високого рівня для імплементації аспектів ІUI в ПЗ інших розробників. Цю мету реалізовували у вигляді REST API сервісу, який знаходиться на віддаленому сервері. Призначенням сервісу є побудова баєсових мереж і здійснення рішень, що робить його повністю самостійним вузлом ІUI. Це також надає перевагу в плані доступності до даних користувачів – оскільки теорема Баєса базується на статистиці, дані всіх користувачів сприятимуть правильним рішенням.

Удар по продуктивності сервера може здійснювати велика кількість користувачів, тому функціонал стане обмеженішим, порівняно з аналогами, але перевагами є висока сумісність та гнучкість. Зменшення функціоналу зумовлюється ігноруванням функцій, які охоплюють аналіз дій користувача в реальному часі. За ігноруванням функцій відбувається та втрата точності даних і, як наслідок, збільшення ймовірностей неправильних рішень ІІІ.

Незважаючи на такі недоліки, програмний продукт має потенціал за рахунок своєї легкодоступності та низького порогу входження для інших розробників.

Функції, передбачені в системі:

1. Створення персонального розкладу.
2. Створення нотаток.
3. Створення нагадувань та використання автоматичних нагадувань, які формуються на згенерованому розкладі користувача.
4. Налаштування будильника.
5. Автоматичні підказки, які базуються на місці перебування користувача.

Архітектура комплексного рішення. У проекті використовують архітектуру з API рішенням, програма працює в режимі запитання-відповідь (рис. 4). "Запитання" надходять від іншого ПЗ з метою занести або отримати результати обчислень з баєсової мережі, яка знаходиться на віддаленому сервері.

Спосіб роботи архітектури:

- Користувач виконує функцію програми X. Розроблена програма передбачає занесення факту виконання цієї функції та прикладних даних у базу знань (баєсову мережу).
- Програма надсилає запит на віддалений сервер, повідомляючи занесення нових даних у мережу. Для цього використовується REST API інтерфейс.
- Ядро віддаленого сервера обробляє отриманий запит і заносить нові дані в баєсову мережу.

- Ядро віддаленого сервера перевіряє та обчислює потрібні ймовірності в баєсовій мережі, реагуючи на зміни кореневих елементів мережі або окремих зовнішніх запитів з особливим "прапорцем".
- У разі успішного обчислення певної ймовірності сервер надсилає потрібні дані локальній програмі, яка обробляє дані та діє на підставі свого алгоритму комунікації з користувачем у конкретному випадку.

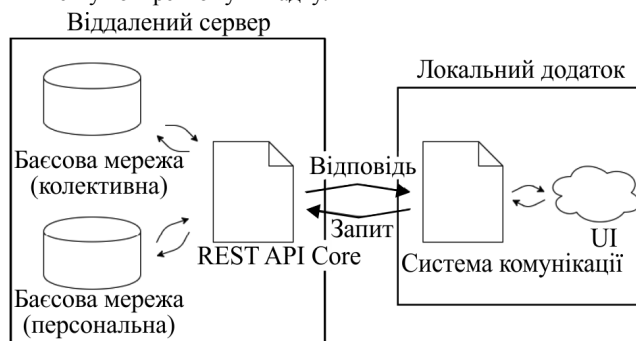


Рис. 4. Архітектура комплексного рішення

За допомогою такої модульної системи сервісом баєсової мережі можуть користуватися будь-які програми. Загальний принцип – програми створюють свої баєсові підмережі та оперують ними за допомогою простого високорівневого оперування даними і надсилення даних на віддалений сервер.

Враховуючи те, що баєсова мережа – це передусім статистичне поняття, розташування мережі на віддаленому сервері дає змогу обчислювати ймовірності, використовуючи дані всіх користувачів певної програми.

Архітектурно інтерфейс складається з трьох основних частин: управління взаємодією, бази знань та центру автентифікації (рис. 5). Жодна частина ІІІ не знаходиться фізично на користувацькому пристрої, а натомість – діє на віддаленому сервері.

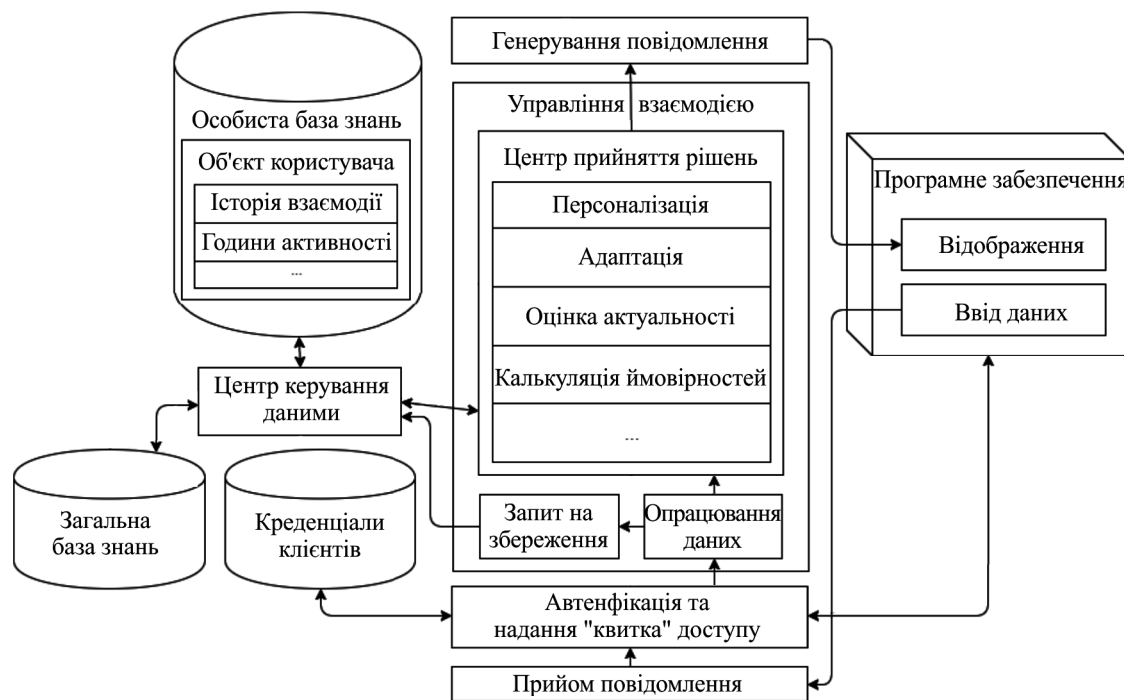


Рис. 5. Складові ІІІ з обробкою на віддаленому сервері

Блок бази знань поділяють на два підблоки: персональна база знань та загальна база знань. Персональна база знань містить всю інформацію про локального користувача у вигляді об'єктного представлення користувача.

Складовими є інформація з апіорною цінністю (ім'я, дата народження, місце роботи, професія та інші особисті дані), історія взаємодії (явні вхідні дані, які вводив користувач за всю історію використання програми), годи-

ни активності (телеметрію активності користувача з погляду астрономічного часу), функціональна інформація (явно введені дані, які належать до виконаних функцій), інформація налаштувань програми й історія налаштувань (дані, які стосуються певних параметрів роботи програми, які визначає користувач), побудована функціональна баєсова мережа (вузли та зв'язки між ними).

Архітектура баєсової мережі з динамічною побудовою складається з трьох рівнів:

1. Рівень вузлів кореневих "свідчень". Цей рівень охоплює головні вузли мережі – час, географічна локація, день тижня тощо.
2. Рівень функціональних вузлів. Цей рівень відображає вузли використання функцій користувачами.
3. Рівень вузлів специфікацій функцій. У цей рівень входять всі додаткові параметри, які використовуються у виконанні функцій програми. Може складатися з багатьох підрівнів.

Приклад використання віддаленої мережі у випадку, коли загальні дані будуть доречними:

- Користувач приходить на зупинку. Локальна програма надсилає повідомлення віддаленому серверу про зміну локації.
- За допомогою обчислень у баєсовій мережі випливає, що ймовірність того, що користувачі шукають розклад трамваїв у цьому географічному місці є дуже високою.
- Віддалений сервер надсилає цю інформацію локальній програмі.
- Локальна програма обробляє інформацію і пропонує користувачеві конкретний веб-сайт із розкладом трамваїв.
- Цей процес має тривати від сотень мілісекунд до декілька секунд. З погляду користувача програма запропонувала веб-сайт одразу після активації пристрою (телефону).

Швидкодія та навантаження системи. Одним з основних недоліків розробленої системи є продуктивність обчислень і навантаження системи, коли йдеться про часте використання сервісу великою кількістю користувачів.

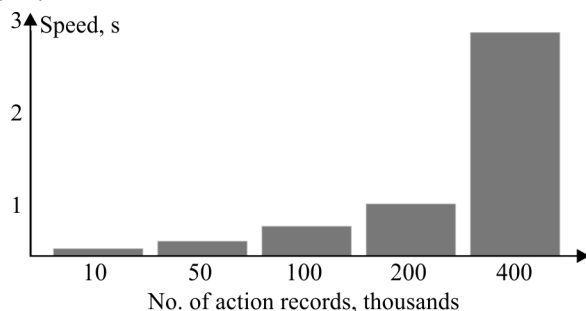


Рис. 6. Швидкість оброблення інформації для одного запиту

У базі даних системи зберігаються усі дії користувачів у вигляді записів у таблицях MySQL. Головною метою логуювання кожної дії є можливість чіткого обчис-

лення апріорних ймовірностей вузлів баєсової мережі. Було виконано експерименти з визначення продуктивності оброблення статистичної інформації з різною кількістю даних (рис. 6). Витрати часу стосовно кількості записів мають нелінійний характер.

Важливо мати на увазі, що кожен "action_report" перетворюється в кілька записів, а деколи – і в кілька десятків записів у базі даних.

Висновки. Запропонована динамічна побудова баєсових мереж на віддаленому сервері дає змогу реалізувати елементи ІUI для іншого ПЗ. Це певною мірою зменшує функціональні можливості ІUI, але істотно покращує сумісність і легкість інтеграції з іншим ПЗ. Точність переважно сильно обмежується можливостями побудови логічних зв'язків, які базуються тільки на окремих запитах від ПЗ.

Наступним кроком до покращення ефективності методу, який представлено у цьому проекті, є врахування інших особливостей користувачів уже на рівні більш глибокого штучного інтелекту і передбаченні можливих дій з погляду статистики, але й логіки.

Перелік використаних джерел

- Benyon, D. R. (1993). Accommodating Individual Differences through an Adaptive User Interface. In M. Schneider-Hufschmidt, T. Kühme, & U. Malinowski (Eds). *Adaptive User Interfaces – Results and Prospects*. Elsevier Science Publications, North-Holland, Amsterdam.
- Bernardo, J., & Smith, A. (1994). *Bayesian Theory*, John Wiley and Sons.
- Birnbaum, L., Horvitz, E., Kurlander, D., Lieberman, H., Marks, J., & Roth, S. (1997). *Compelling Intelligent User Interfaces – How Much AI?* ACJf, Orlando Florida USA.
- Cook, R. & Kay, J. (1994). The justified user model: a viewable, explained user model. In *Proceedings of the fourth international conference on user modeling*, 145–150.
- Höök, K. (2017). *Steps to take before Intelligent User Interfaces become real*. SICS, Kista, Sweden.
- Horvitz, E. (1997). *Agents With Beliefs: Reflections on Bayesian Methods for User Modeling*. Invited talk at Sixth International Conference on User Modeling, Chia Laguna, Sardinia.
- Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (2018). *The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users*. Microsoft Research, Redmond, WA 98052-6399.
- Laviers, K. R. (2011). *Despite Decades of Research, why have Intelligent user Interfaces not Gained Wide Usage?* Department of Electrical and Computer Engineering Air Force Institute of Technology.
- Rohrbach, M. & Schmidt, U. (2017). *Intelligent User Interfaces: Modelling the User*. University of British Columbia.

Е. В. Левус, Ю. О. Гучок, О. О. Нытрэбыч

Національний університет "Львівська політехніка", г. Львів, Україна

АРХИТЕКТУРНОЕ РЕШЕНИЕ ДЛЯ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИНТЕРФЕЙСОВ

Рассмотрена проблема построения интеллектуальных интерфейсов (IUI). Хотя исследования в области IUI проводятся со времен первых разработок искусственного интеллекта, на современном этапе развития инженерии программного обеспечения существует немного успешных решений. Проанализированы известные примеры реализации IUI – проекты компании Microsoft: The Office Assistant и Lumiere Project. Обобщены недостатки этих реализаций. Актуальной является разработка концепции программного обеспечения на основе подхода "минимального взаимодействия", который включает использование интеллектуальных интерфейсов. Важной проблемой IUI, которая в первую очередь требует решения, является нарушение установленных Usability-принципов для систем с классическим интерфейсом. Дополнительным методом "смягчения" последствий нарушения принципов к построению интерфейсов есть четкое разделение адаптивного интерфейса и классического. В качестве решения предложено использование искусственного интеллекта в виде отдаленной байесовской сети; облачных технологий для использования байесовской сети в локальном ПО живую и пополнение сети знаниями, получен-

ных от отдельных локальных пользователей; общения между сетью и локальным ПО с помощью API. Целью разработанного программного продукта является создание гибкой и упрощенной системы высокого уровня для имплементации аспектов IUI в ПО других разработчиков. Данная цель реализована в виде REST API сервиса, который находится на удаленном сервере и предназначен для построения байесовской сети. Следующим шагом к улучшению эффективности представленного метода является учет других особенностей пользователей уже на уровне более глубокого искусственного интеллекта и предвидения возможных действий с точки зрения логики, а не только с точки зрения статистики.

Ключевые слова: программное обеспечение; классический и адаптивный интерфейсы; байесовская сеть, REST API сервис.

Ye. V. Levus, Yu. O. Huchok, O. O. Nytrebych

Lviv Polytechnic National University, Lviv, Ukraine

ARCHITECTURAL SOLUTION FOR INTELLECTUAL USER INTERFACE CREATION

Traditional visual user interfaces are overloaded and not user friendly due to constant expanding of software functionality. The intelligent user interface (IUI) is an alternative to simplifying computer systems usage and solves a task of an interface personalization. Considering trends and shortcomings of existing solutions the relevant issue is a software concept development based on the "minimum interaction" approach which uses IUI. Either an Artificial Intelligence in the form of the remote Bayesian network or cloud technologies for Bayesian network usage in local software and for updating network knowledge obtained from individual local users or communication between a network and a local software with REST API are proposed as a solution for the issue mentioned. The purpose of the developed software is a creation of a flexible and simplified high-level system for the IUI aspects implementation in other software systems. This goal is implemented in the form of REST API service located on a remote server and designed for Bayesian networks construction. In terms of architecture the interface consists of three main parts: interaction management, knowledge base and authentication center. IUI parts are not located physically on a user device as they operate on a remote server instead. A knowledge base block is divided into two sub-blocks such as personal and general knowledge bases. The personal knowledge base contains all the information about the local user in a form of an object representation. The proposed method is implemented as "Personal Assistant" software type. The Bayesian network architecture with dynamic construction consists of the following three levels: the node level of the root "evidence" which includes the main network nodes that are time, geographical location, day of the week, etc.; the functional node level which displays the nodes of functions used by users; the function specification node level which includes all the additional parameters used in software function executions. The present method can be even more promising when not just bare statistics is taken into account, but other users' specifications at a deeper Artificial Intelligence level are considered as well as possible actions in terms of logic (not just in terms of statistics) are predicted.

Keywords: software; classic and adaptive user interfaces; Bayesian network; REST API service.