

**АДАПТИВНИЙ ІНТЕРФЕЙС ДЛЯ ОБОРОННИХ ІНФОРМАЦІЙНИХ СИСТЕМ:
ПРИЗНАЧЕННЯ МОДЕЛІ ТА ЗАДАЧІ ПОБУДОВИ / ADAPTIVE INTERFACE FOR
DEFENSE INFORMATION SYSTEMS: MODEL AND OBJECTIVES**

Задворний С.В., Ільїна О.П. / Zadvornyi S. V., Ilyina O. P.

Інститут програмних систем НАН України / Institute of Software Systems, NAS of
Ukraine

03187, м. Київ, вул. Академіка Глушкова, 40, тел. (044) 526-51-52

E-mail: szadvornyy@gmail.com, ilyina.elena1@ukr.net

The article critiques the flawed practice of developing products without user-centered design, particularly in defense. It introduces the MOBS model, based on defense system classifications, to improve interface usability, efficiency, and adaptability through structured analysis and observation.

Розповсюджена практика створення програмних продуктів без урахування користувацько-орієнтованого проектування часто призводить до ігнорування ключових аспектів інтерфесної взаємодії [1]. Навіть залучення дизайнерів часто не позбавляє браку тестувань і зворотного зв'язку від реальних користувачів. Особливої критичності така тенденція набуває при створенні оборонних інформаційних систем, де якість інтерфейсу впливає на оперативність та безпекову якість рішень.

Для відповіді на ці виклики авторами наразі виконується розробка системи моделей адаптивного інтерфейсу (MOBS), що здійснює інтеграцію аналізу, тестування й адаптації інтерфейсів [2]. Це дозволяє виявляти проблемні зони, адаптувати інтерфейс до потреб користувачів та інтегрувати ці процеси у всі етапи життєвого циклу. Підхід може застосовуватися як до нових, так і до існуючих систем. Особливістю розроблюваної системи моделей є підтримка автоматизованого аналізу MOBS порушень, сценаріїв тестування та ітеративної адаптації. Це дозволяє швидко вдосконалювати інтерфейси, підвищуючи їх ефективність та інтуїтивність. Модель охоплює три етапи життєвого циклу інформаційної системи: створення, використання та актуалізацію, забезпечуючи стале покращення на основі реальних умов експлуатації. Подальші дослідження сфокусовані на апробації моделі в умовах, наближених до реальних.

Побудова моделі адаптації потребувала використання класифікації інформаційних технологій у сфері оборони, яка відображає структурні типи систем і їх функціональні особливості. Спираючись на поточні тенденції розробки такої класифікації в Україні [3], можна прийняти схему, подану наступною таблицею.

Категорія	Опис функціональних можливостей	Приклади систем
Тактичні системи	Управління бойовими операціями, виявлення загроз, координація дій військових підрозділів.	Системи управління боєм (Battle Management Systems, BMS).
Аналітичні системи	Обробка великих обсягів даних, прогнозування сценаріїв, прийняття рішень.	Системи аналізу розвідданих, центри ситуаційної обізнаності.
Логістичні системи	Забезпечення військових операцій матеріальними ресурсами, оптимізація постачання.	ERP-системи для логістики, склади та транспортування.
Комунікаційні системи	Захищений зв'язок між підрозділами, передача даних у реальному часі.	Тактичні радіостанції, супутникові комунікації.
Інтеграційні рівні		
Локальні системи	Працюють автономно, забезпечують окремі завдання.	Комп'ютери в транспортних засобах, локальні бази даних.
Регіональні системи	Об'єднують дані між підрозділами в певному регіоні	Командні центри, мобільні станції.

	для спільних операцій.	
Глобальні мережі	Інтеграція даних і управління в масштабах держави або альянсу.	NATO Federated Mission Networking (FMN).
Операційні середовища		
Наземні	Робота в польових умовах, враховує мобільність і захищеність.	Системи управління військовими транспортом.
Морські	Відстеження і управління морськими операціями.	Системи навігації, управління кораблями.
Повітряні	Координація повітряних операцій, контроль повітряного простору.	Системи управління дронами, ППО.
Кіберпростір	Захист інформаційних ресурсів, виявлення кібератак.	Системи кібербезпеки, моніторинг мереж.

Використання такої класифікації дозволяє створити модель MOBS адаптовану до специфічних умов роботи оборонних інформаційних систем.

В залежності від функціонального класу інформаційних систем (ІС) оборонного призначення, для його представників визначаються набори характеристик інтерфейсу, співвіднесених із функціональними спроможностями, які є суттєвими для підтримки діяльності у відповідній сфері.

Перш за все, визначаються: множина FC типових функціональних спроможностей (функцій) системи, множина FI функцій інтерфейсу, множина $CHR = \{(HR_{il}, MHR_{il})\}_{l=1, \dots, L_i}$ характеристик результативності роботи системи в аспекті i -тої спроможності з їх метриками MHR .

Їх подальша деталізація здійснюється у вигляді Моделі Спостережень за сеансом роботи MOBS, яка формалізує взаємозв'язки між функціональними компонентами інтерфейсу, забезпечуючи основу для оцінки ефективності й адаптації. Таким чином, модель має склад:

$$MOBS = (PSI, \{FC_i, CHR_i, \{FI_{ij}, \{CHQ_{ijk}, \{(OP_{ijk}, MOR_{ijk})\}_{r=1, \dots, R_{ijk}}\}_{k=1, \dots, K_{ij}}\}_{j=1, \dots, M_i}\}_{i=1, \dots, N})$$

де PSI - параметри інтерфейсу, значення яких встановлюються при виконанні сеансу, FC_i - i - та функціональна спроможність системи, FI_{ij} - актуальна для FC_i функція інтерфейсу з UF , CHQ_{ijk} - характеристика користувацької якості виконання FI_{ij} , OP_{ijk} - r -й безпосередньо спостережуваний параметр, що характеризує CHQ_{ijk} , MOR_{ijk} - метрика для OP_{ijk} .

Подальша деталізація моделі має на меті включення автоматизованих інструментів збору даних про поведінку користувачів, а також алгоритмів машинного навчання для аналізу метрик MHR . Це дозволить не лише виявляти проблемні аспекти, але й прогнозувати потенційні покращення, адаптуючи систему до змінюваних умов експлуатації. Важливим етапом подальшої діяльності є апробація моделі в реальних умовах. Очікується, що такі експерименти дозволять уточнити значення параметрів PSI та встановити оптимальні налаштування для різних типів функціональних класів інформаційних систем. Окрему увагу слід приділити підвищенню інформативності характеристик CHQ , зокрема через розширення множини FI .

Таким чином, запропоновано базові концептуальні рішення створення моделі MOBS для аналізу та вдосконалення інтерфейсів як нових, так і існуючих інформаційних систем. Її інтеграція в оборонні інформаційні системи забезпечить підвищення їхньої ефективності, інтуїтивності та адаптивності, що сприятиме покращенню загальної продуктивності та безпеки користувачів.

Список джерел:

1. Купер А. Психікарня в руках пацієнта: створення дизайну, орієнтованого на користувача. Київ: ArtHuss, 2020. 500 с.
2. Gullà F., Ceccacci S. Design Adaptable and Adaptive User Interfaces: a Method to Manage the Information // Ambient Assisted Living: Italian Forum 2014. 2014. № 3, pp. 2–4.
3. Ліпко І. О., Звір В. Б., Капілевич В. О., Сугак С. О. Запровадження класифікації інформаційних технологій у сфері оборони: досвід НАТО для України //

**АЛГОРИТМИ ТА МЕТОДИ СИНГУЛЯРНОГО РОЗКЛАДУ МАТРИЦЬ НА
ГРАФІЧНИХ ПРОЦЕСОРАХ З CUDA / ALGORITHMS AND METHODS FOR SINGULAR
VALUE DECOMPOSITION OF MATRICES ON CUDA-ENABLED GPUS**

Сухарський С. С. / Sukharskyi S. S.

Національний університет “Києво-Могилянська академія” / National University of “Kyiv-Mohyla Academy”

04655, Київ, вул. Григорія Сковороди, 2, каф. Мережних технологій, (044) 425-77-23

E-mail: s.sukharskyi@ukma.edu.ua

Singular Value Decomposition (SVD) is a fundamental tool in linear algebra with a wide range of applications, including data processing and machine learning. With the advancement of Graphics Processing Units (GPUs), optimizing computational methods for SVD has become a pressing task. This study introduces new approaches to bidiagonalization and diagonalization of matrices using Householder transformations and Givens rotations, effectively leveraging GPU capabilities. Developed algorithms demonstrate high speed and accuracy while minimizing memory usage. Experiments revealed that proposed bidiagonalization algorithm works quite well, while optimizing diagonalization by reducing block synchronization delays is the main challenge. Future work will focus on improving these aspects to enhance performance on large matrices.

Сингулярний розклад (SVD), який описується формулою $A = U\Sigma V^T$, де A вхідна матриця $m \times n$, U ортогональна матриця розміру $m \times m$, Σ діагональна матриця $m \times n$ з невід’ємними значеннями розташованими за спаданням, а V ортогональна матриця $n \times n$, є фундаментальним інструментом у лінійній алгебрі та чисельних методах, з широким спектром застосувань: від обробки сигналів і зображень до машинного навчання та аналізу великих даних. Зі зростанням обсягів даних і складності моделей виникає необхідність у більш ефективних обчислювальних методах. Графічні процесори (GPU) з технологією CUDA надають можливість значно прискорити обчислення завдяки їх масовому паралелізму. Тому розробка ефективних алгоритмів SVD для GPU є вкрай актуальною для підвищення продуктивності в різних наукових та інженерних областях. Багато дослідників розробляли паралельні версії традиційних алгоритмів SVD, таких як метод Джейкобі або QR-алгоритм, для виконання на GPU [3]. Наприклад Лагабар та Нараянан в своєму дослідженні [3] використовують метод Хаусхолдера та неявно зміщений QR алгоритм. При чому, серію трансформацій Хаусхолдера виконують лише на графічному процесорі, а діагоналізацію з залученням центрального процесора для контролю ходу виконання. В своєму дослідженні я використовую інший підхід, де саме бідіагоналізація виконується з контролем ходу виконання на центральному процесорі, а друга частина вже відбувається повністю на відеокарті, щоб максимально ефективно використати можливості відповідних процесорів. Частина дослідників сфокусувалась на рандомізованих алгоритмах, що обчислюють наближений SVD для великих матриць [5]. Цей підхід дозволяє пришвидшити обчислення, проте підходить не для всіх задач. Також є дослідження блокових алгоритмів сингулярного розкладу, що обробляють матриці частинами [4]. Цей підхід є перспективним, проте не розкриває проблему синхронізації даних між блоками, над чим я працюю в дослідженні.

Зміст цього дослідження полягає в розробці та реалізації оптимізованого алгоритму SVD для графічних процесорів, який перевершує існуючі рішення за критеріями часу виконання, точності обчислень та ефективності використання пам’яті. Рішення включатиме нові підходи до паралелізації обчислень, оптимізацію використання архітектури GPU, а також вдосконалення чисельних методів для зменшення похибок і