

УДК 004

Є. Кузьменко, магістр гр. КН-21М-1,4,

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПЕРЕГЛЯДУ ХМАРНИХ СЕРВІСІВ

У статті розроблено програмне забезпечення, яке призначено для системи перегляду хмарних сервісів. Метою розробки є дослідження та програмна реалізація системи перегляду хмарних сервісів. Об'єктом дослідження є процес перегляду хмарних сервісів. Предметом дослідження є методи перегляду хмарних сервісів. Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи перегляду хмарних сервісів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерні науки, хмарні сервіси

Постановка проблеми. Сучасний світ неможливо представити без глобальної мережі Інтернет. Через Інтернет зараз відбувається доступ до різного виду інформації (починаючи від фільмів та музики й закінчуючи новинами та науковими статтями), спілкування між людьми, які розташовані на величезних відстанях один від іншого, покупки в Інтернет-магазинах та ще багато-багато інших функцій. Інтернет надає доступ до різного роду хмарних сервісів.

Але для того, щоб працювати з інтернетом необхідна програма доступу до веб-сторінок, з яких складається Інтернет. Такою програмою є веб-браузер, або, якщо казати по іншому – Інтернет-браузер, як складова системи перегляду хмарних сервісів. Веб-оглядач, або браузер, як складова системи перегляду хмарних сервісів – це програмне забезпечення для пошуку, перегляду веб-сайтів, тобто для запиту веб-сторінок (переважно з Інтернету), для їхньої обробки, висновку й переходу від однієї сторінки до іншої. Більшість браузерів також наділені здатностями до перегляду змісту FTP-серверів. Браузери постійно розвивалися із часів зародження Інтернету, і з його ростом ставали усе більше важливою програмою типового персонального комп'ютера. Нині браузер, як складова системи перегляду хмарних сервісів – комплексний додаток для обробки й виведення різних складових веб-сторінки, і для надання інтерфейсу між веб-сайтом і його відвідувачем. Практично всі популярні браузери поширюються безкоштовно або «у комплекті» з іншим додатком: Microsoft Edge (як невід'ємна частина Microsoft Windows), Mozilla Firefox, Opera, Safari (разом з Mac OS або безкоштовно для Windows). Крім цих браузерів існує ще велика кількість інших браузерів, які мають ті або інші функції, крім основної, перегляду веб-сторінок. Але не дивлячись на таке розмаїття браузерів, немає вітчизняного продукту, який міг би створити конкуренцію цим браузерам. У цьому магістерському проекті зроблена спроба створити такий браузер, як складова системи перегляду хмарних сервісів.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи перегляду хмарних сервісів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи перегляду хмарних сервісів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем перегляду хмарних сервісів.
- Дослідження системи перегляду хмарних сервісів.

– Програмна реалізація системи перегляду хмарних сервісів.

Об'єктом дослідження є процес перегляду хмарних сервісів.

Предметом дослідження є методи перегляду хмарних сервісів.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Вимоги висунуті до браузерів

Базова ідея Web-технологій полягає в тім, що будь-який користувач будь-якого браузера може підключитися до будь-якому Web-додатку. Однак на перевірку ця потенційно приваблива ідея є фікцією. Універсальний доступ до інформації став можливим лише завдяки титанічним зусиллям розроблювачів додатків для Web, яким довелося пройти дуже складний шлях, перш ніж їхні програми «навчилися» розпізнавати різні версії браузерів і динамічно адаптувати інтерфейс до особливостей цих браузерів. Це кропітка й стомлююча робота, яку доводиться проробляти щораз після появи нового браузера або чергової його версії. Але ж всі ці зусилля можуть виявитися даремними, коли повсюдне поширення загальноприйнятих стандартів зведе їх нанівець.

Відповідальність за твердження стандартів, що стосуються браузерів, покладена на Консорціум World Wide Web і Європейську асоціацію виробників комп'ютерної техніки (European Computer Manufacturer's Association – ECMA). Незважаючи на те що представники компаній, що займаються розробкою браузерів, беруть участь у діяльності комісій, вони не можуть скільки-небудь істотно вплинути на прийняття остаточних рішень. Адже тут перетинаються інтереси самих різних груп. Розроблювачам браузерів хотілося б робити враження затятих апологетів стандартів – і в той же час мати цілком лояльних клієнтів. Microsoft і Netscape протягом тривалого часу вбудовували у свої продукти всі нові й нові специфічні можливості, намагаючись у такий спосіб залучити розроблювачів. Користувачі ж віддають перевагу найбільш зручному для себе браузеру й вимагають, щоб додатка були сумісні з обраним ними варіантом. Природно, все це жадає від розроблювачів додаткових зусиль, а код Web-додатків непомірно роздувається, оскільки в ньому повинні враховуватися всі розходження між представленими на ринку браузерами.

З випуском кожної нової версії виробники пропонують своє оригінальне трактування стандартів. Якби всі браузери дійсно відповідали стандартам, розроблювачеві досить було б написати всього один інтерфейс свого Web-додатка, і цей інтерфейс виглядав би абсолютно однаково в середовищі будь-якого браузера. Захвати, що супроводжували появу кожної нової версії, через деякий час перемінялися розчаруванням, оскільки користувачі починали зауважувати, що прірва, яка розділяє продукти різних виробників, стає усе ширше.

На щастя, часи міняються. Зокрема, останні версії браузерів компаній Microsoft, Mozilla і Opera Software повністю відповідають вимогам існуючих стандартів. Те ж саме можна сказати й про Konqueror – складову частину відомого середовища KDE (K Desktop Environment). Звичайно, повністю гарантувати те, що всі браузери будуть точно відповідати специфікаціям стандартів, ще не можна, але розроблювачі вже переходять до створення Web-додатків, орієнтованих на стандарти, а не на специфічні характеристики конкретних браузерів.

Призначення стандартів

Стандарти браузерів охоплюють два основних аспекти Web-технологій у частині інтерфейсу: його зовнішній вигляд і функціональна реалізація тих або інших можливостей. До стандартів зовнішнього подання екранного інтерфейсу відносяться затверджені W3C специфікації HTML 5 і CSS (Cascading Style Sheets) рівня 3. Мова HTML визначає базову структуру й форматування Web-документа, а каскадні таблиці стилів дозволяють додати до неї точний контроль за атрибутами екранного інтерфейсу: шрифтами, квітами й схемами розташування. Стандарти функціональної реалізації ECMA-262 (або ECMAScript) і запропонована W3C об'єктна модель документа CSS DOM (Document Object Model) описують клієнтські сценарії і їхню можливість динамічно змінювати інформаційне наповнення Web-сторінки.

Специфікації HTML 5, затверджені W3C в 1999 році, описують лише децимію базових конструкцій, що лежать в основі Web-додатків. Однак незважаючи на розходження, що існують між браузерами різних виробників, всі вони підтримують HTML – якщо й не версії 4.01, то 4.0. У користувача, що має у своєму розпорядженні останню редакцію браузера Opera, Firefox, Microsoft Edge або Konqueror, не повинне виникати ніяких труднощів з переглядом HTML-сторінок. Їхній код не містить специфічних розширень типу керуючих елементів Active або модулів Netscape. Будь-який розроблювач, що бажає встановити, чи відповідає його Web-сторінка вимогам стандартів, має можливість перевірити це за допомогою служби HTML Validator на сайті www.w3c.org. Цей програмний інструментарій ідентифікує всі нестандартні конструкції HTML, які можуть бути оброблені браузерами неправильно.

Формування стилів

Стандарт W3C CSS1 (CSS Level 1) з'явився ще в 1996 році, але виробники браузерів не поспішали підтримувати його у своїх продуктах. У браузерах Netscape до появи шостої версії було дуже багато помилок у підтримці цього стандарту, розроблювачі не забезпечили повної сумісності з CSS1. Корпорація Microsoft реалізувала функції CSS1 тільки у версії Microsoft Edge 6.0. Сьогодні обоє розроблювача повністю підтримують стандарт CSS1. Недавно до браузерів шостого покоління Microsoft і Netscape приєдналися продукти Opera 5 і Konqueror 2.2, розроблювачі яких також заявили про повну сумісність їхнього програмного забезпечення з CSS1. Через п'ять років після твердження специфікацій стандарту нарешті з'явилася повсюдна підтримка CSS1.

Але що це дає розроблювачам Web-додатків? Кожному елементу HTML-коду, у тому числі й відображуваним на екрані (наприклад, заголовкам і гіперпосиланням), ставиться у відповідність певний стиль CSS1. Стандарт CSS1 і модель DOM дозволяють розроблювачам динамічно змінювати властивості поточної сторінки. Підтримка стандарту CSS1 всіма виробниками браузерів стала великою перемогою розроблювачів додатків для Web.

В 1998 році W3C затвердив специфікації стандарту CSS2 (CSS Level 2), у якому підтримка таблиць стилів була поліпшена за рахунок функцій розміщення текстових і графічних елементів у довільному місці сторінки. На щастя, всі виробники браузерів уже оголосили про сумісність майбутніх версій своїх продуктів з CSS2.

Підтримка сценаріїв

Мова Netscape JavaScript, доповнена технологією Microsoft JScript, перетворилася згодом у стандарт ECMA-262. У грудні 1999 року ECMA затвердила третю редакцію стандарту ECMA-262, що розроблялася під найменуванням ECMAScript. Сьогодні всі популярні браузери підтримують ECMAScript, а Microsoft і Netscape оголосили про повну сумісність своїх продуктів із третьою редакцією ECMAScript. Проте розроблювачі як і раніше продовжують використовувати клієнтський код, у якому присутні нестандартні функції JavaScript і JScript. Виробникам варто ретельно вивчити свої клієнтські сценарії й внести зміни в ті з них, де невиправдано використовуються специфічні можливості браузера. Конструкції JavaScript і JScript можна трансформувати в код ECMAScript при мінімальних зусиллях.

Додати до браузера базовий рівень підтримки сценаріїв досить просто. Організувати керування зв'язками між кодом сценаріїв і документом Web складніше, тут браузери найбільшою мірою відрізняються один від іншого. Всі популярні браузери підтримують сьогодні сукупність стандартів DOM1 (DOM Level 1) і DOM2 (DOM Level 2), надаючи за допомогою сценаріїв простий доступ до елементів HTML і властивостям таблиць CSS. В останніх версіях Microsoft Edge, Navigator і Konqueror забезпечена повна підтримка DOM1. А от Opera має тут дуже серйозні обмеження, особливо це стосується можливості додавання й видалення елементів HTML на відображуваній сторінці.

Гарантії безпеки

Відповідність Web-браузерів вимогам стандартів залишає поки бажати кращого, однак уже сьогодні розроблювачі додатків для Web можуть використовувати набір базових

функцій, що буде підтримуватися всіма популярними браузерами. Здається, що першим кроком у будь-якому новому проекті створення додатків для Web (а також у будь-якому проекті, ціль якого полягає в адаптації вже існуючого коду до особливостей стандартів) повинне стати надання клієнтам гарантій технічної підтримки й відновлення ПЗ в майбутньому. Практика показує, що не можна змусити користувачів Navigator перейти на Microsoft Edge, але при цьому більшість охоче встановлюють у себе останню версію улюбленого браузера.

Попередньо необхідно все вивчити й протестувати, щоб виявити області, у яких функції часткової підтримки браузерами специфікацій DOM2 і CSS2 перетинаються.

В ідеалі всі ці стандарти повинні позбавити нас від додаткових зусиль по розпізнаванню браузерів і адаптації до їхніх особливостей. На практиці ж завжди залишаються користувачі, у яких будуть установлені застарілі версії або не настільки широко розповсюджені програми браузерів. Якщо виявиться, що клієнтський браузер, як складова системи перегляду хмарних сервісів не підтримує якісь функції вашого додатка, переадресуйте його на більше простий сайт і повідомите про переваги, які принесе установка оновленого варіанта браузера.

GUI інтерфейс браузера

У сучасному браузері повинні бути реалізовані наступні елементи інтерфейсу GUI:

- Адресний рядок.
- Автопошук.
- Вибір протоколу за замовчуванням в адресному рядку.
- Механізм автопідстановки адреси сайту.
- Префікс і суфікс за замовчуванням. За замовчуванням стандартним префіксом є префікс www. а стандартним суфіксом – .com.
- Механізм автозаповнення – браузер, як складова системи перегляду хмарних сервісів перевіряє рядок, що ви вводите, з рядками з галузі реєстру, і якщо знаходить збіги, тоді пропонує вам вибрати їх у списку, що розкривається.

- Панель пошуку.
- Зміна пошукової машини за замовчуванням.
- Меню Виправлення.
- Меню Вид.
- Меню Вибране.
- Меню Сервіс.
- Меню Довідка.
- Спливаючі панелі.
- Вкладка Додатково.
- Кнопка режиму передперегляду.
- Панель команд.
- Контекстне меню браузера.
- Зміна масштабу.

Стандартні функції браузеру

До стандартних функцій сучасного браузеру відносяться наступні:

- перегляд із вкладками;
- здатність зберігати множина вкладок під однією закладкою;
- download менеджер;
- блокувальник спливаючих вікон;
- фішинг-фільтр;
- вбудований RSS-агрегатор;
- підтримку інтернаціональних доменних імен;
- підтримку засобів групової політики;

- підтримку HTML 5;
- підтримку CSS Level 1;
- підтримку XML 1.0;
- підтримку DOM Level 1 і частково CSS Level 2 і DOM Level 2;
- підтримку підключення розширень, що реалізується через об'єктну модель компонентів (COM);
- hotclick;
- переклад тексту на іншу мову;
- перевірка орфографії;
- розпізнавання мишачих жестів;
- функцію блокування рекламних баннерів і флеш-роликів;
- меню швидкого перемикання прокси-серверів зі списку;
- опцію коректування змінної User Agent String, відповідальної за ідентифікацію браузера веб-серверами;
- конвертор сторінок у графічні зображення форматів JPG, BMP, GIF, PNG або TIFF;
- механізм підключення користувальницьких сценаріїв для маніпулювання HTML-контентом на стороні браузера;
- функцію відновлення всіх відкритих вкладок, що збереглися з попередньої робочої сесії браузера;
- опцію Super Drag&Drop, що дозволяє банальним перетаскуванням об'єктів на веб-сторінці виконувати звичні речі: робити швидкий пошук виділеного тексту в заданому за замовчуванням пошуковому сервісі, ефектно зберігати зображення й документи, а також проробляти інші речі спритним рухом мишки;
- кілька готових надбудов і скриптів для скачивання відеороликів з Google Video, Youtube і інших онлайн-сервісів.

Панелі

Функція, пов'язана з переглядом електронної пошти й іншого. Крім того, користувач може завантажити додаткові панелі або створити свою власну.

- Контакти – служить як адресна книга;
- Історія – надає журнал з усіма сторінками, які були відвідані, в хронологічному порядку, починаючи із самої останньої;
- Посилання – перераховані всі посилання, які користувач використовував у період перегляду сайтів;
- Примітки – дозволяє користувачеві копіювати й вставляти вміст зі сторінок в убудований текстовий редактор для наступного перегляду й редагування;
- Інформація – показує інформацію про сторінку, у тому числі її тип MIME, розмір і кодування;
- Вікна – приводиться коротка інформація із всіх вкладок і вікон, відкритим у цей момент.

Перспективи розвитку функцій браузера

До перспективних функцій браузера за результатами дослідження OpenAjax Alliance відносяться наступні:

- Більш повна й послідовна підтримка 2D і векторної графіки, спрямована на забезпечення платформної й вендору-незалежності.
- Скасування обмеження у два одночасних підключення до сервера. Воно проявляється, якщо браузер, як складова системи перегляду хмарних сервісів запитує більше 2 сторінок одночасно й серйозно ускладнює роботу Ajax web 2.0 додатків. Зараз браузери, що встановили більше 2 сесій спеціально блокують наступні з'єднання до завершення попередніх.

–Швидкодія HTML DOM операцій. У частині споживання ресурсів, це самі не економічні із всіх JavaScript операцій, що вириваються «уперед» на тисячі відсотків. І тому що вони є одними з найважливіших, необхідно починати роботу з їхньої оптимізації.

Опис розроблюємого Інтернет–браузера

При написанні різних браузерів спостерігається таке явище: основна кількість браузерів роблять на якихось певних движках.

Движок – виділена частина програмного коду для реалізації конкретної прикладної задачі – програма, частина програми, комплекс програм або бібліотека, залежно від задачі й реалізації. Як правило, прикладна частина виділяється із програми для використання в декількох проектах і/або роздільної розробки/тестування.

Використання готового движка при розробці програми, сайту або іншого продукту скорочує час розробки, дозволяє приділити більше часу розробці інших підсистем, наприклад користувальницькому інтерфейсу (або інформаційної наповненості сайту).

Разом з тим продукти, зроблені з використанням движків, успадковують їхні помилки й проблеми безпеки.

Найбільше браузерів зроблено на движку Microsoft Edge, це й не дивно, тому що браузер, як складова системи перегляду хмарних сервісів древній як динозавр, і ще не швидко він вимре, тому що його розвитком і підтримкою займається Майкрософт. На розробку браузерного движку потрібні чималі ресурси, тому програмістам легше робити різні оригінальні примочки й доповнення до движка, іменованому Microsoft Edge, і називати цей браузер, як складова системи перегляду хмарних сервісів уже своїм, оригінальним ім'ям, хоча це все той же Microsoft Edge тільки модифікований як у крашу, так і в гіршу (що буває досить часто) сторону.

У магістерському проекті за основу браузера взято движок Trident. Trident (також відомий як MSHTML) – назва браузерного движка для Windows–версії Microsoft Microsoft Edge. Уперше Trident був реалізований у четвертій версії Microsoft Edge, і з тих пор піддавався постійним поліпшенням і переробкам.

Trident був розроблений як програмний компонент, що дозволяв розроблювачам програмного забезпечення легко додавати можливість перегляду веб-сторінок у їхні власні додатки. Він використовує технологію COM (*компонентну модель об'єктів*) для перегляду й виправлення веб-сторінок у будь-якому оточенні, що підтримує інтерфейс цієї моделі – наприклад, в Delphi, C++ або .NET. Відповідний елемент керування може бути доданий, скажемо, у програму, написану на Delphi – і Trident буде використовуватися для доступу до задалегідь заданого веб-сторінці, для читання або зміни інформації, що перебуває на ній. Події елемента керування будуть перехоплюватися й передаватися в основну програму. Функції ядра Trident стануть доступні при додаванні бібліотеки mshtml.dll до програмного проекту.

Розробка структурної схеми. Структурна схема розробленої системи зображена на рисунку 1.

Веб-сервер

Веб-сервер – це програмне забезпечення, встановлене на комп'ютері, за допомогою якого по протоколу HTTP надається доступ до веб-сторінок. Найпоширенішими веб-серверами є Apache, що працює на платформі Unix/Linux і IIS (Internet Information Service), що працює під керуванням Windows. Також веб-сервером називається й сам комп'ютер, на якому встановлене це програмне забезпечення й зберігаються файли веб-сайтів. Для захисту сервера від атак і взлому звичайно встановлюється мережевий екран і відповідне програмне забезпечення.

Функції сервера:

- на сервері зберігаються різні ресурси (наприклад веб-сторінки);
- для кожного ресурсу встановлюються права доступу;
- сервер обробляє запити клієнтів, яким необхідний той або інший ресурс;
- сервер здійснює обмін інформацією із клієнтами й іншими серверами.

Internet-браузер, як складова системи перегляду хмарних сервісів (клієнт)

Розроблений Internet-браузер, як складова системи перегляду хмарних сервісів являється програмою клієнтом.

Функції клієнта:

- браузер, як складова системи перегляду хмарних сервісів здійснює запит необхідного ресурсу;
- браузер, як складова системи перегляду хмарних сервісів обробляє отриманий ресурс;
- у рядку адреси браузера набирається адреса сайту, на який хоче потрапити користувач (наприклад `http://www.eusr.com`);
- браузер, як складова системи перегляду хмарних сервісів відправляє запит спеціальному комп'ютеру, який зветься DNS-сервер (Domain Name System);
- DNS-сервер перетворить набрану адресу в числову (IP-адреса) адресу сервера, на якому розташований сайт (наприклад 212.147.139.162) і поверне його браузеру;
- браузер, як складова системи перегляду хмарних сервісів відправляє запит на отриману адресу й у відповідь одержує запитуваний ресурс.
- Після того, як ресурс переданий, з'єднання між клієнтом і сервером розривається.

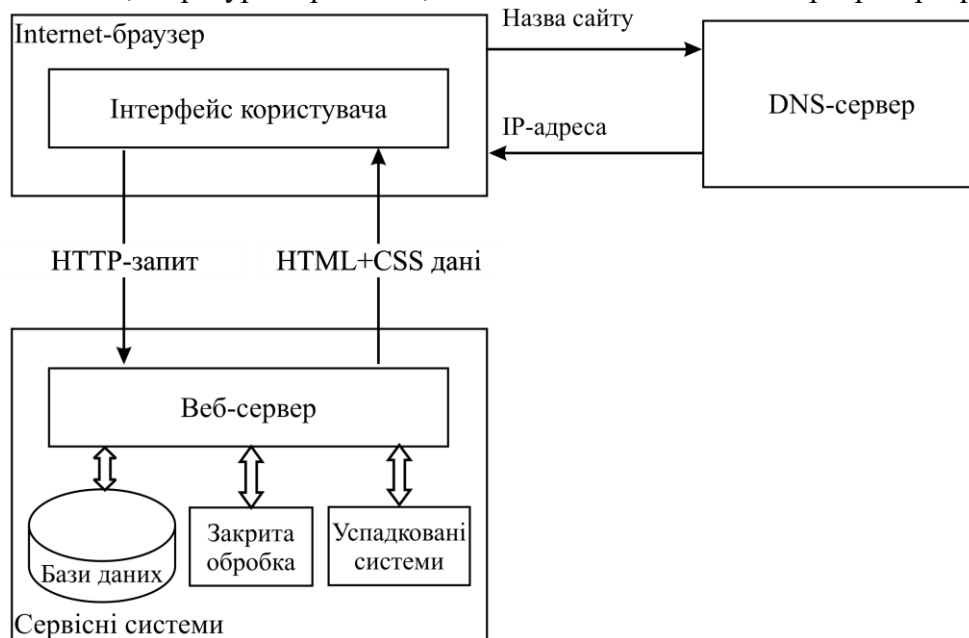


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів перегляду хмарних сервісів. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем перегляду хмарних сервісів; Досліджена система перегляду хмарних сервісів; На основі отриманих результатів досліджень створена програмна реалізація системи перегляду хмарних сервісів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання перегляду хмарних сервісів. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Список літератури

1. Смірнова Т.В., Поліщук Л.І., «дослідження хмарних технологій як сервісів для системи інженерних розрахунків» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛІОС», 2020.С. С. 106-121.
2. Smirnova, T., Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems, 2021, vol. 152, pp. 85–98. (Scopus).
3. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184. (Scopus).
4. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399. (Scopus).
5. Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Smirnov O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus)
6. Смірнова Т.В., Дреєв О.М., Смірнов О.А., «Експертна система оптимізації процесу відновлення та зміцнення поверхонь деталей типу «вал» електродуговим напиленням», Системи управління, навігації та зв'язку, № 2 (54). с. 149-154, 2019.
7. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М., «Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей», Центральньоукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.
8. Смірнова Т.В., Дреєв О.М., Смірнов О.А., Солових Є.К., «Методи оптимізації технологічних процесів відновлення сталевих покриттів», Shipbuilding & marine infrastructure / Суднобудування і морська інфраструктура № 1 (11). с. 48-57, 2019.
9. Smirnova T., Ageev M., Lopata L., Dudan A., Lopata A., «Of combined electric arc coatings», Problems of Tribology, Vol. 24 № 3/93. P. 51-61, 2019.
10. Смірнова Т.В., Поліщук Л.І., Смірнов О.А., Буравченко К.О., Макевнін А.О., «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
11. Смірнова Т.В., «Формалізація та реалізація структури технологічного процесу електродугового напилення для оптимізаційної експертної системи», Технічні науки та технології. № 1(19). С. 104-113. 2020.
12. Smirnova T., Smirnov I., Lopata A., Lopata L., «Improvement of functional properties of gas-thermal coatings by electro-contact treatment», Problems of Tribology, Vol. 25, № 1/95, P. 41-48. 2020.
13. Смірнова Т.В. «формування евристичних правил, бази знань та формалізація структури й правил технологічного процесу для оптимізаційної хмарної інформаційної системи», Системи управління, навігації та зв'язку, № 2 (60). с. 101-104, 2020.
14. Смірнова, Т.В., Смірнов, С.А., Минайленко, Р.М., Доренський, О.П., Сисоєнко С.В., «Хмарна автоматизована система інтелектуальної підтримки прийняття рішень для технологічних процесів». Вісник Черкаського державного технологічного університету. Технічні науки. №4, 2020, С. 84-92.
15. Смірнова Т.В., Пархоменко Д.О., Голубець Р.О., Щербань А.В., Багдасарян Е.К., «Формалізація проблеми підтримки технологічних процесів у хмарних сервісах». Системи озброєння і військова техніка. 2021. № 3(67). С. 105-112.
16. Смірнова Т.В., Столяренко М.П., Янков М.О., Грудік В.В., Моторін Ю.Ю., «Модель реалізації структури технологічного процесу у хмарному сервісі». Збірник наукових праць Харківського національного університету Повітряних Сил. 2021. № 4(70). С. 132-142.
17. Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., Смірнов О.А., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
18. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Бурмак Ю.А., Оспанова Д.М., «Удосконалений модуль криптографічного захисту інформації в сучасних інформаційно-комунікаційних системах та мережах». Кібербезпека: освіта, наука, техніка. 2021. № 2(14). С. 176-185.
19. Смірнова Т.В., Бурмак Ю.А., Улічев О.С., Усік П.С., Доренський О.П., «Стійка функція шифрування удосконаленого модуля криптографічного захисту інформації в інформаційно-комунікаційних системах» Кібербезпека: освіта, наука, техніка. 2021. № 1(13). С. 183-201.
20. Смірнова Т.В., Моторін Ю.Ю., Буравченко К.О., Бочуля Т.В., Коваленко О.В. «Вибір оптимальної технології побудови хмарної інформаційно-комунікаційної системи автоматизації виробничих процесів». Вимірювальна та обчислювальна техніка в технологічних процесах, № 1 (2022). С. 15-26. 2022.