

І.Ю. Гришанова, Ю.В. Рогушина

РОЗРОБКА МЕТОДІВ КЕРУВАННЯ ДОСТУПОМ ДО ІНФОРМАЦІЇ WIKI-РЕСУРСАХ

В роботі аналізуються технологічні засади розробки Wiki-ресурсів на основі програмного забезпечення MediaWiki, розглядаються проблеми доступу до інформації у цих ресурсах та наводяться методи та засоби вирішення цих проблем. Значна увага приділяється архітектурі MediaWiki та складовим цієї архітектури. Розроблені в результаті аналізу рішення апробовано у розробці порталу Великої української енциклопедії (е-ВУЕ).

Ключові слова: MediaWiki, Wiki-ресурс, керування доступом, Велика українська енциклопедія.

Вступ

Wiki-технології є основою для створення структурованих інформаційних ресурсів великого обсягу, орієнтованих на одночасне використання через Web великою кількістю користувачів. Таке подання інформації є зручним та інтуїтивно зрозумілим для широких кіл користувачів, має достатню виразну здатність для відображення досить складних змістових відношень між окремими елементами ресурсу.

Нині Wiki-технології широко застосовуються для створення різноманітних відкритих енциклопедійних ресурсів. Наприклад, портальна версія Великої української енциклопедії (е-ВУЕ) базується на програмному забезпеченні MediaWiki [1], дозволяє подавати авторський та рецензований мультимедійний контент із складною системою зв'язків та навігації [2]. Процес створення контенту є централізованим та контролюється модераторами наукових напрямків, що виключає безпосередню участь користувачів у редагуванні даних. Частина даних у е-ВУЕ призначена для забезпечення семантичного пошуку та навігації і не орієнтована на безпосереднє використання кінцевими користувачами, що призводить до необхідності керувати доступом до інформації відповідно до ролей користувачів цього порталу. Але технологія MediaWiki не підтримує безпосередньо такий функціонал, і тому виникає потреба у розробці таких спеціалізованих програмних засобів, що дозволяють вирішувати ці задачі.

Wiki-технологія створення відкритих інформаційних ресурсів

Wiki-ресурси являють собою перспективний шлях розвитку приватних і публічних баз знань. Оригінальна система Wiki була винайдена в 1995 р. У. Канінгемом [3] для Web-вузла Pattern Languages Community, щоб спростити спільне створення й документування програмного забезпечення. Всесвітньо відомий приклад застосування технології Wiki – це Вікіпедія, найбільша з безкоштовних онлайн-енциклопедій [4].

Під *Wiki-технологією* зазвичай розуміють таку технологію побудови Wiki-ресурсу, яка дає змогу відвідувачам брати участь у редагуванні його вмісту – виправляти помилки, додавати нові матеріали, не використовуючи спеціальні програми, не реєструючись на сайті й не вивчаючи мову HTML [5].

Кожна окрема сторінка Wiki-ресурсу називається **Wiki-сторінка** (*Wiki page*). Створення й редагування Wiki-сторінок не потребує від користувачів знання мови HTML: механізм Wiki надає можливість написання документів за допомогою простої мови розмітки й Web-браузеру. Всі Wiki-сторінки мають унікальні назви, на які можна посилатися з інших статей, і вміст. Інформація, представлена у Wiki, має нелінійну навігаційну структуру: кожна сторінка, зазвичай, містить велику кількість гіперпосилань на інші сторінки.

У 2006 році термін Wiki додано до онлайнового Оксфордського Словника Англійської мови (OxfordEnglish Dictionary, OED) як позначення моделі сайтів, контент яких може змінювати сам користувач.

Формат Wiki-сторінок (Wiki-текст) – це спрощена мова розмітки, що використовується для того, щоб виділити в тексті різні структурні й візуальні елементи або вказати на них. Кожна Wiki-система має власний стиль і синтаксис залежно від реалізації. У багатьох реалізаціях Wiki гіперпосилання показується таким, яким воно є насправді, на відміну від HTML, де невидиме гіперпосилання може мати довільний видимий текст або зображення.

Для створення та підтримки Wiki-ресурсу необхідно спеціальне програмне забезпечення (ПЗ) – *Wiki-рушій*, який забезпечує роботу відповідного Web-сайту. Рушій Wiki-сайту робить запис змін так, щоб у будь-який час сторінку можна було повернути до кожного з її попередніх станів. Wiki-система може також містити різні інструментальні засоби для простого контролювання стану Wiki, що постійно змінюється, а також для обговорювання й розв'язання проблем, що виникають через різні погляди на зміст Wiki-сторінок.

Популярність технологій Wiki сприяла появі великої кількості реалізацій практично для всіх можливих платформ і конфігурацій ПЗ. Крім того, різні реалізації Wiki неоднаково розширюють основні можливості технології, додаючи, наприклад, убудовані графічні редактори чи сервіси WAP. Але загальна орієнтованість на створення відкритих інформаційних ресурсів (IP) призводить до певних проблем, що пов'язані з обробкою персональних даних та із захистом інформації від несанкціонованого доступу (приміром, до службової інформації, що не призначається для кінцевого користувача, але необхідна для ефективної організації Wiki-ресурсу).

MediaWiki як технологічна основа побудови Wiki-ресурсів

Широко вживаним рішенням для створення Wiki-ресурсів є MediaWiki [6]. Це програмне забезпечення розроблено

для створення найбільш відомого Wiki-ресурсу – Вікіпедії. Воно враховує розподілену розробку незалежними розробниками додаткових плагінів та функціоналу без зміни архітектури.

MediaWiki розроблено для підтримки спільноти, яка створює та керує вільне повторно використовуване знання на відкритій платформі. Тому MediaWiki не включає стандартний функціонал корпоративних систем керування контентом (CMS), натомість пропонує різноманітні інструменти для боротьби зі спамом та вандалізмом.

Архітектура MediaWiki повинна враховувати не тільки розподілене використання ПЗ, постійно зростаючу кількість користувачів та авторів контенту, патрулювання для збереження цілісності контенту, боротьби зі спамом та вандалізмом, а також і розподілений процес розробки ПЗ як афілійованими розробниками, так і окремими сторонніми користувачами, надавши їм можливість додавати власні додатки до загальної бази програмного коду.

Для організації спільного відкритого процесу розробки з використанням зусиль різних розробників та волонтерів, було обрано такі найбільш поширені та оптимальні інструменти: LAMP платформа – мова PHP (з подальшим використанням фреймворків Symfony та інших, як Zend Framework), база даних MySQL/MariaDB (використовується як базова з можливістю зміни на PostgreSQL, SQLite тощо), jQuery як клієнтська бібліотека JavaScript. Розробка відбувається в стилі відкритого коду [7] і координується спеціальними спільнотами та групами Wikimedia Foundation. Документація коду генерується автоматично, загальна інформація публікується на сторінках Вікіпедії.

В ретроспективі визнано, що багато рішень прийняті не зовсім оптимально, а деякі були помилковими. Але зважаючи на невелику кількість розробників і волонтерський характер участі у проекті, швидкість розробки і невеликий обсяг коду, важко критикувати засновників проекту. Проект зростає, розвивається і постійно викорис-

товується, тому базові рішення (навіть ті, які нині визнано не оптимальними) здебільшого кардинально не змінюються, а ведеться пошук рішень, які зберігають існуючу архітектуру.

На даний час нам не вдалося знайти більш-менш сталого і змістовного визначення архітектури MediaWiki. Ресурси, пов'язані із розширенням функціоналу MediaWiki, містять базові рекомендації для розробників, опис змінних, опис таблиць БД, але не надають формалізований опис архітектури. Для забезпечення безпеки, розробники ядра і ревізійники коду розробили жорсткі правила кодування, а також надали набори обгортки (wrappers) для екранування вхідних html-даних та ескейпування sql-запитів (класи WebRequest, Sanitizer).

MediaWiki може мати багато можливих варіацій конфігурацій, параметри яких зберігаються в глобальних php-змінних. Значення за замовчуванням встановлені в DefaultSettings.php, але адміністратор може змінити їх, вказавши нові або додаткові значення в файлі LocalSettings.php. З точки зору безпеки та можливості виникнення конфліктів, це не є добрим рішенням, але процес розвитку ПЗ MediaWiki має довгу і ще не закінчену історію руху від глобальних змінних до об'єктів. Для збереження наступності версій і підтримки старих, конфігурація зберігається за допомогою змінних.

MediaWiki використовує реляційну БД (за замовчуванням це MySQL), і Вікіпедія використовує саме цю БД, але спільнота розробила підтримку інших БД, таких як PostgreSQL, Oracle, SQLite. Під час первинної процедури встановлення адміністратор обирає необхідну БД. ПЗ MediaWiki забезпечує рівень абстракції БД і рівень абстракції запитів до БД, що полегшує розробникам доступ до неї.

В процесі розвитку MediaWiki схема БД багаторазово змінювалася, найбільш значною зміною було розділення місця зберігання текстів і відслідкування ревізій в MediaWiki 1.5. На даний час БД складається з десятків таблиць, які містять дані про контент (наприклад page, revision,

category, recentchanges), про користувачів (user, user_groups), про медіа файли (image, filearchive), кешування (objectcache, 110n_cache, queuycache), внутрішні засоби (job – для виконання черги задач), а також може містити інші додаткові дані, що встановлені додатково та є необхідними для певних плагінів.

Абстрактна модель архітектури MediaWiki

Архітектура MediaWiki на абстрактному рівні містить (рис. 1):

- базу даних;
- базове ядро MediaWiki;
- набір плагінів, які розширюють функціональні можливості ПЗ;
- набір шкінів (шаблонів відображення контенту сторінок та засобів навігації).

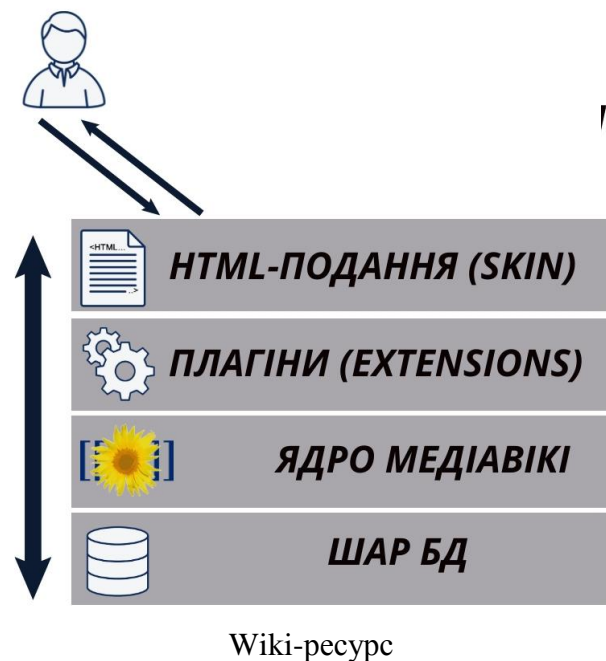


Рис. 1. Абстрактна модель архітектури MediaWiki

Архітектурно відображення контенту Web-сторінок (навігації та контенту сторінок) максимально відокремлено відповідно до MVC моделі (model-view-control) і винесено на рівень шкінів (шаблонів відображення), хоча деякі невеликі частини інтерфейсу, які історично створені раніше, не вдалось повністю відокремити від основного коду бізнес-логіки.

Набір стилів оформлення (так звані skin) дозволяють користувачам модифікувати зовнішнє оформлення MediaWiki [8]. Зазвичай базовий комплект програмного забезпечення містить 4 базових набори: Vector, Monobook, Cologne Blue та Modern. Стиль, встановлюваний за замовчуванням – Vector. Деякі набори стилів не підтримують мобільної версії. Додаткові компоненти, такі як JavaScript, також функціонують тільки в певних наборах стилів.

На даний час в e-VUE прийнятий для використання як базовий додатковий набір стилів оформлення Foreground, який фокусується на виділенні контенту, підтримує адаптивні макети та має спеціальні задалегідь визначені класи для підтримки функціоналу Semantic MediaWiki. Цей набір побудовано на Zurb's Foundation Framework (v4.3.2), який орієнтований у першу чергу на підтримку мобільних пристроїв та розширеного адаптивного фронт-енд фреймворку.

Skin (skin) – це набір елементів, що визначають дизайн інтерфейсу (стили, меню, шрифти тощо). У MediaWiki скіни – це PHP-класи, кожен з яких розширює батьківський клас Skin; вони містять функції, які збирають інформацію, необхідну для генерування HTML. На відміну від плагінів, розробка або налаштування скінів – тяжка і нестандартна задача. Наприклад, стандартний скін "MonoBook" довгий час було важко налаштувати під нову версію, оскільки він містив багато стилів CSS, які підтримують старі браузері; редагування шаблону або CSS вимагало багатьох наступних змін, щоб відобразити зміни для всіх браузерів і платформ.

Скін задає стиль оформлення сторінок (з використанням CSS та JavaScript):

- розташування елементів навігації, інформаційних блоків, кнопок тощо;
- фон сторінок та фон елементів сторінки;
- типи та розміри шрифтів для різних типів абзаців;
- розміри відступів від блоків та елементів сторінки.

Набір стилів оформлення розроблюється і задається один раз і поширюється на всі сторінки.

Зазвичай Web-сторінка може мати стандартний набір зон: навігаційний блок зверху, зони меню навігації ліворуч та праворуч, основний зміст сторінки в середині, і внизу сторінки – зона підвалу. Набір стилів оформлення Foreground повністю змінює макет зовнішнього відображення Вікі-сторінки та дозволяє додавати нові елементи інтерфейсу. В e-VUE всі функціональні меню перенесені в горизонтальний навігаційний рядок у вигляді випадаючих меню, що знаходиться наверху сторінки.

Стандартна сторінка MediaWiki за замовчуванням містить певний обмежений набір компонентів (рис. 2):

- елементи персональних інструментів та налаштування для зареєстрованого користувача;
- меню набору дій, які можливо робити з сторінкою (редагувати, обговорення, перегляд історії тощо);
- форма пошуку по сайту;
- системні повідомлення (site notice);
- гасло сторінки;
- рядок сторінки, вищої за таксономічним поділом та джерело перепосилання (redirected from);
- блок контенту сторінки, який може містити текст, мультимедійну інформацію, формули, таблиці, підзаголовки, блок змісту. Контент сторінки може розділятися на частини, кожна частина може редагуватися окремо, для чого з'являється елемент «Редагувати» над кожною виділеною частиною;
- блок категорій, до яких належить сторінка;
- нижній колонтитул з піктограмами та посиланнями;
- ліва бічна панель (sidebar), яка містить посилання на певні головні сторінки, сторінки довідок, блок посилань на інструменти та блок посилань на версії сторінки іншими мовами;
- індикатор статусу сторінки.

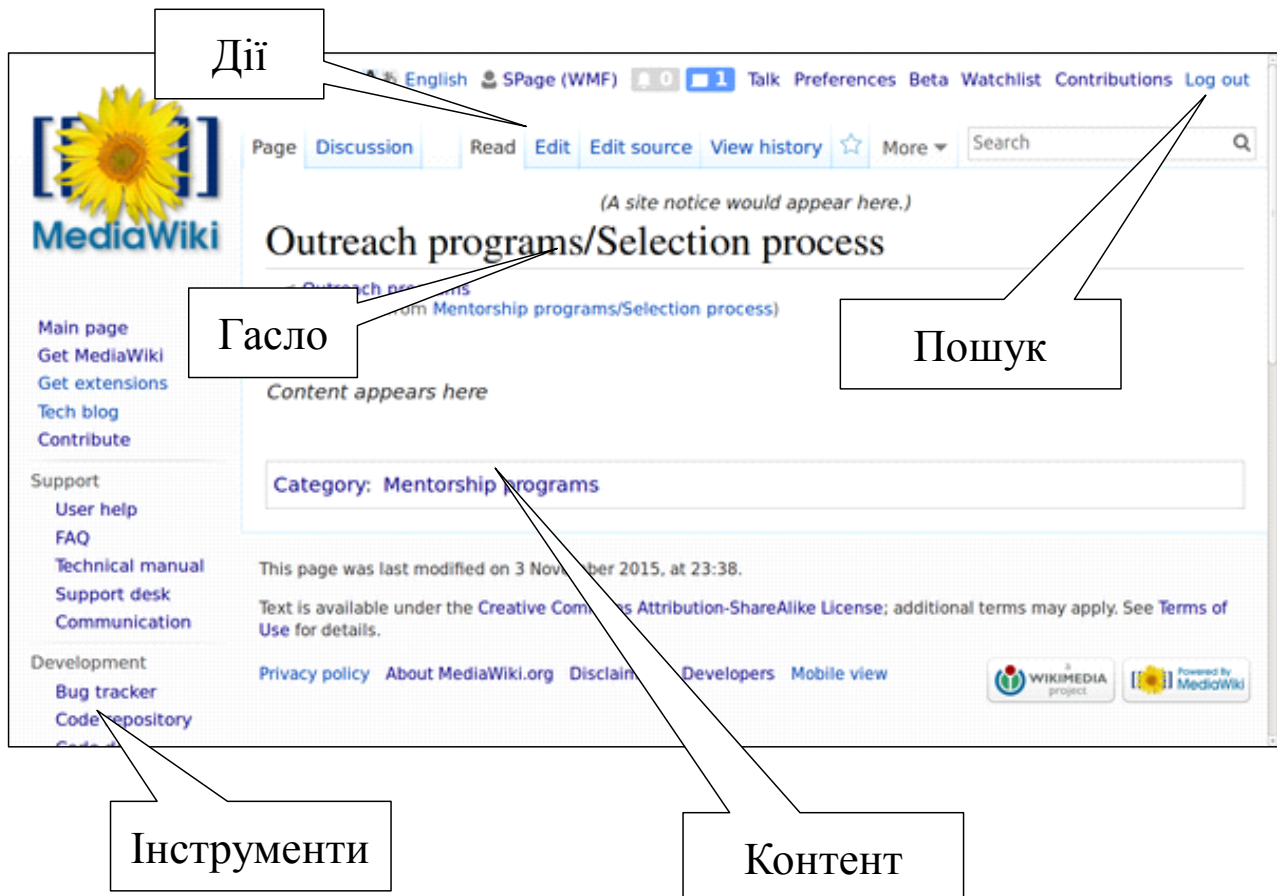


Рис. 2. Основні елементи Wiki-сторінки

Процес виконання Web-запиту в MediaWiki

Головною точкою доступу в MediaWiki є `index.php`, який керує більшістю запитів, поданих до серверу. Код цього модулю виконує перевірку безпеки, завантажує встановлені конфігураційні значення з `includes/DefaultSettings.php`, виконує конфігурацію за допомогою `includes/Setup.php`, після чого встановлює конфігураційні змінні, що відповідають конкретному серверу і прописані в файлі `LocalSettings.php`. Потім він створює об'єкт MediaWiki (`$mediawiki`) та створює об'єкт Title (`$wgTitle`) в залежності від заголовка та параметрів дії, вказаних в URL-запиті.

`index.php` може приймати різні параметри дії, вказані в URL-запиті; дією за замовчуванням є перегляд (*view*), яка показує звичайний вигляд вмісту статті. Наприклад, запит `https://en.wikipedia.org/w/index.php?title=Apple&action=view` відображає зміст статті "Apple" в англійській Вікіпедії. Також часто застосовують на-

ступні дії: *edit* (відкриття статті для редагування), *submit* (для попереднього перегляду редагування або збереження статті), *history* (для відображення історії змін статті) та *watch* (для додавання статті до списку спостереження користувача). Адміністративні дії включають *delete* (для видалення статті) та *protect* (захист для запобігання редагуванню статті).

Для обробки більшості URL-запитів викликається метод `MediaWiki::performRequest()`. Він перевіряє наявність поганих заголовків, обмеження читання, локальні переспрямування та цикли переспрямування та визначає для якого типу сторінки запит – для звичайної чи спеціальної.

Запити звичайної сторінки передаються методу `MediaWiki::InitializeArticle()`, який створює для сторінки об'єкт `Article` (`$wgArticle`), а потім методу `MediaWiki::performAction()`, який обробляє "стандартні" дії. Після завершення дії виконується `MediaWiki::finalCleanup()`, який

доопрацьовує запит, здійснюючи закінчення транзакцій БД, виводячи згенерований HTML і запускаючи відкладені оновлення, які поставлені в чергу завдань. `MediaWiki::restInPeace()` закінчує виконання відкладених оновлень та закриває завдання.

Якщо запитувана сторінка є спеціальною (тобто не звичайна сторінка Wiki-ресурсу, а спеціалізована сторінка, пов'язана з відповідним програмним забезпеченням або виконанням певних спеціальних операцій, наприклад, Статистика), замість `InitializeArticle()` викликається `SpecialPageFactory::ExecutePath` і виконується відповідний PHP скрипт. Спеціальні сторінки можуть робити всілякі операції і кожна має певну мету, як правило, незалежну від будь-якої статті чи її змісту. Спеціальні сторінки включають різного роду звіти (останні зміни, журнали, категорії без категорій) та засоби адміністрування Wiki (блоки користувачів, зміни прав користувача). Послідовність їх виконання залежить від їх функції.

Багато функцій містять код профілювання, який дає можливість, якщо профілювання включено, слідкувати за робочим процесом виконання, що корисно для відлагодження. Профілювання здійснюється за допомогою виклику функцій `wfProfileIn` та `wfProfileOut`, які відповідно запускають та зупиняють профілювання; обидві функції приймають ім'я функції як параметр. На сайтах WikiMedia (таких як Вікіпедія, Вікідата тощо), щоб зберегти продуктивність, профілювання зазвичай відключене або проводиться на один відсоток від усіх запитів.

Під час перегляду сторінки HTML-код може бути взято з кешу. Якщо ж кешування відключене або сторінка кожен раз генерується заново, то спочатку розгортаються шаблони, функції парсеру та змінні. В результаті генерується розширений Wiki-текст, що є проміжним результатом, який можна побачити за допомогою *Special:ExpandTemplates*, і який залежить від:

- Wiki-тексту;
- шаблонів, на які прямо чи опосередковано є посилання;

- функцій і параметрів парсеру, на які прямо чи опосередковано є посилання;

- значень змінних, на які прямо чи опосередковано є посилання.

Далі цей розширений Wiki-текст перетворюється в HTML-код; він надсилається користувачеві та містить посилання на CSS, JavaScript та файли зображень. Користувач може побачити цей проміжний результат, застосувавши в браузері опцію «view source». HTML-код для певної сторінки залежить від:

- розширеного Wiki-тексту;
- режиму, перегляд чи редагування (див. далі);

- наявності внутрішньо пов'язаних сторінок (дає посилання для перегляду чи редагування);

- скіну та інших налаштувань користувача;

- імені користувача;

- статусу користувача (більше посилань, якщо адміністратор тощо);

- простору імен (визначає посилання на сторінку Talk або у випадку Talk-сторінки – відповідну сторінку);

- чи відслідковує користувач сторінку (надає посилання для включення чи відключення відслідковування за змінами);

- чи була нещодавно відредагована Talk-сторінка користувача (видає повідомлення).

Нарешті, браузер рендерить HTML з використанням файлів, на які є посилання. Результат, який користувач бачить на екрані, залежить від:

- HTML-коду;

- файлів, на які посилається код HTML, таких як вставлені зображення, CSS-файли на стороні сервера та файли JavaScript;

- браузера та його налаштувань, включаючи, можливо, локальний файл CSS та роздільну здатність екрана.

Якщо JavaScript реагує на подію, таку як клацання миші, сторінка на екрані також залежить від цих подій. Це може виникати, наприклад, у разі виконання сортування таблиці.

Якщо користувач обирає вкладку “Редагувати”, клієнту надсилається тільки Wiki-текст (контент сторінки або окремого розділу цієї сторінки з Wiki-розміткою, але без навігаційних елементів). Якщо користувач натискає «Попередній перегляд», на сервер для обробки надсилається нова версія Wiki-тексту, і після обробки сервер надсилає нову версію HTML-коду. Якщо користувач натискає кнопку "Зберегти", надсилаючи нову версію статті на сервер, система записує здійснені редагування та надсилає браузеру HTML-код нової версії (знову). У деяких випадках на цьому етапі також відбувається автоматичне перетворення Wiki-тексту.

Кешування в MediaWiki

Оскільки MediaWiki є базовим ПЗ усіх сайтів Wikimedia, він максимально оптимізований для реалізації високої продуктивності.

На сайтах Wikimedia більшість запитів обробляються зворотним кешуванням проксі-серверів (Squids) і ніколи не роблять це на серверах застосунків MediaWiki. Проксі-сервери містять статичні версії цілих відтворених сторінок, які служать для простого читання для незареєстрованих користувачів. MediaWiki спочатку підтримує Squid і Varnish та інтегрується з цим шаром кешування, наприклад, сповіщаючи їх про очищення сторінки з кеша, коли вона була змінена. Для зареєстрованих користувачів та виконання інших запитів, які не можуть обслуговувати проксі, Squid пересилає запити на Web-сервер (Apache).

Другий рівень кешування використовується, коли MediaWiki надає та збирає сторінку з декількох об'єктів, багато з яких можна кешувати, щоб мінімізувати майбутні звернення. Такі об'єкти включають інтерфейс сторінки (бічна панель, меню, текст інтерфейсу користувача) та власне вміст, проаналізований з Wiki-тексту. Кеш об'єктів у пам'яті доступний в MediaWiki з ранньої версії 1.1 (2003), і це особливо важливо, тому що дозволяє уникнути повторного розбору сторінок із складною структурою та великим обсягом.

Дані сесій також можуть зберігатися у *memcached*. Починаючи з версії 1.16, MediaWiki використовує виділений кеш-об'єкт для локалізованого тексту інтерфейсу; це було додано після того, як помітили, що значна частина об'єктів, кешованих у пам'яті, складалася з повідомлень інтерфейсу, поданих мовою користувача.

Останній шар кешування складається з кешування коду PHP (PHP opcode cache), яке зазвичай використовується для прискорення роботи PHP програм. Компіляція може бути тривалим процесом; щоб уникнути компіляції PHP-скриптів у код виконання кожного разу, коли вони викликаються, для зберігання зібраного коду і виконання його безпосередньо без компіляції може використовуватися PHP-акселератор. MediaWiki може працювати з багатьма акселераторами, такими як APC, PHP-акселератор та eAccelerator.

Також шар кешування абстрактних об'єктів MediaWiki дозволяє зберігати кешовані об'єкти в декількох місцях, включаючи файлову систему, базу даних або opcode cache.

Структура контенту MediaWiki

Простори імен (namespaces), так само як і спеціальні Wiki-сторінки, були вперше використані в PHP-скрипті для Нупедії (попередниці Вікіпедії) розробником MediaWiki Г.М. Манське [9]. Впродовж багатьох років namespaces застосовувалися у Wiki-ресурсах, виконуючи лише одну функцію – розділяти різні види контенту. За написом namespaces є префіксом, який відокремлений від назви сторінки двокрапкою (наприклад, *Talk.*, *File:* та *Template:*); в основному просторі імен вмісту префікса немає. Простори імен виявились важливою особливістю MediaWiki, оскільки вони створюють необхідні передумови для спільноти Wiki та встановлюють дискусії на метарівні, процеси спільноти, портали, профілі користувачів тощо.

Простори імен дозволяють структурувати контент Wiki-ресурсів, розділити контент за функціональним призначенням. В межах одного простору імен сторінки можуть бути впорядковані за темами за допомогою категорій. Існує певний список

вже зарезервованих просторів імен: *User* – для розділення користувачів, *Mediawiki* – для службових сторінок ядра MediaWiki, *File* – для зберігання інформації про файли, *Special* – для службових сторінок, *Template* – для сторінок шаблонів.

Як було зазначено вище, в загальному вигляді Web-серверна архітектура розподіляє процес отримання результату на етапи: серверну обробку запиту, підготовку відповіді, відправку та обробку отриманих даних на клієнті і відображення їх в браузері (або іншому клієнтському ПЗ).

Відповідно до цього підходу, процес обробки сервером запиту користувача складається з наступних етапів:

- виконання запиту користувача і підготовка необхідних для відсилання даних;
- перетворення даних у кінцевий формат шляхом трансформації та обгортання в html-формат, додавання елементів навігації, інтерфейсу та дизайну;
- відправлення згенерованої Web-сторінки клієнту (браузеру).

Архітектура MediaWiki пропонує різні способи налаштування та розширення програмного забезпечення. Це можна зробити на різних рівнях доступу:

- системні адміністратори можуть встановлювати розширення, скіни, налаштовувати окремі допоміжні програми wiki (наприклад, для відображення мініатюр для зображень та генерування TeX) та глобальні налаштування;
- користувачі Wiki з правами sysops (іноді їх також називають "адміністраторами") можуть редагувати гаджети, налаштування JavaScript та CSS;
- будь-який зареєстрований користувач може налаштувати власний інтерфейс, використовуючи свої уподобання (для існуючих налаштувань, скінів та гаджетів) або внести зміни до власної моделі інтерфейсу (використовуючи свої особисті сторінки JS та CSS).

Зовнішні програми також можуть спілкуватися з MediaWiki через його API, якщо він включений, в основному роблячи будь-яку функцію та дані доступними для користувача.

MediaWiki забезпечує систему гачків (hooks) – перехвату, що дозволяють стороннім розробникам запускати користувацький PHP-код до, після або замість коду MediaWiki для конкретних подій. Розширення MediaWiki використовують такі гачки для підключення свого коду.

До появи в MediaWiki цих засобів перехвату процес додавання спеціального коду PHP викликав зміну основного коду. За допомогою гачків можливо розширити мову Wiki-розмітки додатковими можливостями, вводячи нові теги.

Екосистема MediaWiki дозволяє розробникам створювати плагіни, які називаються розширеннями (extensions) і використовують гачки. Система розширень має низку недоліків: реєстрація розширень базується на виконанні додаткового коду при кожному запуску системи, а не на кешованих даних, що обмежує абстрагування та оптимізацію та шкодить продуктивності MediaWiki. Але в цілому сучасна архітектура розширень є досить гнучкою інфраструктурою, яка допомагає зробити спеціалізований код більш модульним, утримуючи основне програмне забезпечення від занадто великого збільшення та дозволяючи стороннім користувачам будувати власну функціональність на основі стандартної версії MediaWiki.

Використання авторизації для визначення прав доступу до інформації

В MediaWiki реалізовано можливість розподілу функціоналу за допомогою авторизації. В момент встановлення Wiki-системи є можливість одразу визначити, закритий чи відкритий тип матиме ця система. Якщо система встановлюється як відкрита, то кожен незареєстрований користувач має повний доступ до контенту всіх сторінок.

Авторизація – керування рівнями та засобами доступу до певного захищеного ресурсу, як у фізичному розумінні (доступ до кімнати готелю за карткою), так і в галузі цифрових технологій (наприклад, автоматизована система контролю доступу) та ресурсів системи залежно від ідентифікатора і пароля користувача або надання

певних повноважень (особі, програмі) на виконання деяких дій у системі обробки даних [10].

З позицій інформаційної безпеки авторизація – частина процедури надання доступу для роботи в інформаційній системі, після ідентифікації і автентифікації.

Авторизація контролює доступ легальних користувачів до ресурсів системи після успішного проходження ними автентифікації. Найчастіше процедури автентифікації й авторизації поєднують. За допомогою авторизації встановлюються й реалізуються права доступу до ресурсів.

Створення облікового запису користувача у MediaWiki означає, що цей користувач вказує ім'я облікового запису (логін) і пароль. Для наступного входу користувачу потрібно вказати логін і знову підтвердити пароль.

Для перегляду будь-якого відкритого Wiki-ресурсу на основі MediaWiki авторизація не потрібна. Як правило, вона не потрібна й для того, щоб редагувати контент. Проте доступ з авторизацією надає користувачу кілька важливих переваг. Інші користувачі зможуть довідатися про діяльність користувача (зміни, внесені у сторінки) за логіном користувача. Це зручніше, ніж використання для цього IP-адреси користувача, яка може змінюватися, якщо користувач заходить до Wiki-ресурсу з різних пристроїв та місць. Спрощується спілкування з іншими користувачами Wiki-ресурсу, простіше відслідковувати зміни в цікавлячих сторінках, використовуючи список спостереження, та виникає можливість отримувати автоматичні повідомлення про важливі події.

Незважаючи на те, що технологія Wiki орієнтована на розробку відкритих джерел інформації, в процесі функціонування таких ресурсів виникає потреба у захисті деяких фрагментів даних від несанкціонованого доступу. Зазвичай виникає необхідність закрити деякі певні сторінки, які містять конфіденційну бізнес-інформацію або є частиною робочого процесу установи, що розробляє відповідний Wiki-ресурс.

Це викликано як потребою у захисті персональних даних, так і доцільністю

закриття технічних деталей реалізації від кінцевих користувачів.

Зважаючи на відкриту ідеологію Wiki-систем, відповідні механізми в базовій версії MediaWiki відсутні. Існуючі плагіни, розроблені для цього сторонніми розробниками або мають лише бета-версію, або взагалі вже не підтримуються і є застарілими. Тому необхідно, використовуючи існуючі технологічні засоби середовища MediaWiki, створити механізм, який забезпечить захист елементів контенту Wiki-ресурсу для різних груп користувачів цього ресурсу. Додатковою умовою є максимальна незалежність такого рішення від оновлення версій ПЗ, гнучкість і простота його використання.

Постановка задачі

Враховуючи існуючу архітектуру системи і принцип відкритості Wiki-ресурсу, необхідно створити технічне рішення – надбудову над системою, яка не змінить кардинально код системи, буде логічним і простим у використанні, але дозволить диференціювати доступ до інформації для різних категорій користувачів відкритого Wiki-ресурсу відповідно до їх ролей, повноважень та інформаційних потреб.

Для вирішення такої задачі пропонується: виконати семантичний поділ контенту (сторінок, що складають Wiki-ресурс) на окремі множини, що не перетинаються, відповідно до їх функцій та типу вмісту, використовувати механізм розподілу контенту за просторами імен, який існує в WikiMedia; розробити механізм перехвату інформації, що згенерована на сервері системою, яка буде враховувати знання про цей розподіл та пов'язані з ним правила обробки для того, щоб передавати на клієнт тільки ті відомості, які є дозволеніми для цієї групи користувачів.

Алгоритм керування доступом до контенту Wiki-ресурсу

У найбільш узагальненому розумінні, для рішення проблеми керування доступу до контенту містить такі підзадачі:

- розподіл користувачів на групи доступу відповідно до їх прав доступу до окремих елементів контенту (наприклад, на зареєстрованих і незареєстрованих);
- розподіл контенту на групи (з використанням таких механізмів MediaWiki, як простори імен та категорії, а також семантичного розширення Semantic MediaWiki – семантичних властивостей сторінок та їх значень);
- створення шаблонів типових інформаційних об'єктів (ТІО), що описують ті дані, доступ до яких має бути диференційованим для користувачів з різних груп доступу;
- розробка правил співставлення належності користувача до певної групи доступу з тим, яку саме інформацію він має право отримувати;
- реалізація механізму для отримання набору даних, що доступні для конкретного користувача, що надає можливість аналізувати згенерований контент перед відправкою його на клієнт;
- створення шаблонів для подання інформації відповідно до аналізу, що виконано у попередньому пункті.

Розглянемо окремі випадки реалізації цього підходу. Для таких випадків пропонується використовувати механізм розподілу контенту за просторами імен, що існує в WikiMedia, і розробити механізм перехоплення інформації, що згенерована на сервері системою, яка буде передавати на клієнт тільки те, що дозволено.

У загальному випадку з урахуванням специфіки MediaWiki, що проаналізована вище, вхідними даними задачі є:

- розподіл користувачів на зареєстрованих і незареєстрованих;
- поділ контенту на різні простори імен та категорії (та можливість додавати до сторінок певні шаблони);
- можливість додавати програмний код, здатний аналізувати контент, згенерований MediaWiki, перед відправкою даних на клієнт.

Архітектура Wiki побудована таким чином, що на останньому етапі перед відправкою згенерованої сторінки вона оброблюється на рівні шкіни. На цьому етапі мо-

жливо додати до інформації певні html-теги, а також зробити більш глибокий аналіз контенту. Найбільш простим є аналіз простору імен, до якого відноситься сторінка, що аналізується.

Скін Foreground програмно реалізовано двома класами: class Skinforeground, який розширює базовий клас ядра SkinTemplate і клас формату подання – class foregroundTemplate, який розширює BaseTemplate. Разом такий формат відображення і весь програмний код шкінів подано в файлі типу Назва Шкіну.skin.php. [11]. Розробка власного шкіну надає можливість вносити такі операції аналізу, не впливаючи на ядро MediaWiki. Внаслідок цього виникає можливість аналізувати контент відповідно до набору обраних умов та згенерувати новий контент відповідно до власних потреб.

Узагальнений алгоритм роботи такого шкіну містить наступні етапи:

- формування сторінки контентом згідно отриманих параметрів ядром Wiki в змінній *\$body*;
- аналіз змісту змінної *\$body* і внесення змін відповідно до прав користувача і правил доступу;
- відображення змінної *\$body* в браузері.

Перевірка, чи зареєстрований користувач, здійснюється методом – *\$isLoggedIn = \$wgUser->isLoggedIn()*.

Отримання простору імен поточної сторінки, здійснюється методом: *\$namespace = \$this->getSkin()->getTitle()->getNsText()*; Інші параметри сторінки отримуються відповідно.

Виконання перевірки та отримання дозволу виконується функцією *\$allow_to_show = check_allow_to_show_content(\$isLoggedIn, \$namespace, \$isArticle, \$title, \$content)*.

Ця функція працює за наступною логікою:

```
if ($allow_to_show == 1),
```

зміна контенту згідно з правилами показу

```
else
```

зміна контенту згідно з правилами.

Функція визначення дозволу має наступний вигляд:

```
function
check_allow_to_show_content($isLoggedIn,
$namespace, $isArticle, $title, $content) {
    $allow_to_show = 1;
    if ($isLoggedIn != 1) {
        if ($namespace=="XXXX")
            $allow_to_show
= 0;
        elseif ($isArticle) {
            $allow_to_show =
check_page_edition_status($vue_content);
        }
    }
    return $allow_to_show;
}
```

Функція *check_page_edition_status* (*\$content*) аналізує контент статті (сторінки) і визначає дозвіл на показ чи редагування поточної сторінки для певного користувача.

Таким чином, підключення аналізатору до скіну дозволяє проаналізувати сторінку і надати користувачеві тільки ту інформацію, яку можна показувати саме йому. Таке рішення не змінює основного ядра ПЗ, не залежить від версії ПЗ і дозволяє безболісно встановлювати оновлення ядра і додаткових плагінів.

Можливість MediaWiki семантично розділяти контент шляхом використання механізму категорій та просторів імен надає багато інших можливостей і переваг для створення енциклопедичного контенту та корпоративних інформаційних ресурсів, тому що це забезпечує не тільки відображення контенту, але й інтеграцію цієї системи в робочий процес підприємства. Запропоноване рішення надає більше можливостей для використання MediaWiki не тільки як базу знань, але й як допоміжну довідкову систему для керування внутрішніх корпоративних бізнес-процесів документообігу.

Важливо зазначити, що запропоноване рішення не стосується захисту доступу до API MediaWiki та від різних типів хакерських атак тощо.

Використання методу керування доступом в е-ВУЕ

Запропонований вище підхід використовується для розширення функціоналу портальної версії Великої української енциклопедії (е-ВУЕ). Наразі структура БЗ е-ВУЕ формально зафіксована за допомогою відповідної Wiki-онтології, яка є результатом співпраці експертів Про (у даному випадку – модераторів галузей знань, що представлені в енциклопедії) з інженерами із знань. Для складної структури знань, що характерна для е-ВУЕ, це дозволяє значно чіткіше описувати знання та запобігати повторного використання імен категорій з різними значеннями [12] та імпортувати знання з інших онтологій [13].

Проаналізувавши структуру БЗ, ми виділили наступні простори імен, які потребують спеціалізованої обробки з точки зору доступу до параметрів сторінок:

- шаблон;
- подія;
- артефакт;
- зображення;
- медіафайли;
- аудіосупровід;
- модератор.

Кожен з цих просторів імен має специфічне призначення в е-ВУЕ і має використовуватися тільки розробниками ресурсу у службових цілях. Наприклад, простір імен Артефакт забезпечує виведення на першу сторінку порталу відомостей про ті артефакти, створення яких пов'язано з поточною датою.

Інший приклад – шаблони виведення інформації про типові елементи контенту е-ВУЕ. Доступ до таких шаблонів мають отримувати тільки розробники цього ресурсу, автори та модератори наукових напрямків (рис. 3).

Цей шаблон “Персоналія” призначений для опису типових інформаційних об'єктів, пов'язаних з конкретними особами, – персоналій. За допомогою цього шаблону забезпечується уніфіковане подання інформації про осіб на тих сторінках Wiki-ресурсу, що їм відповідають.

Користувачі е-ВУЕ, що не авторизовані, не отримують доступ до цього шаблону. Тому контент таких сторінок буде схованим від них за допомогою розглянутих вище засобів (рис. 4).

Шаблон

Персоналія

Це шаблон «Персоналія». Він повинен викликатися наступним чином (після знаку = можна вводити значення параметрів):

```

{{Персоналія
|Прізвище=
|Ім'я=
|По батькові=
|Прізвище та ім'я мовою оригіналу=
|Псевдонім=
|Справжнє ім'я=
|Справжнє прізвище=
|День народження=
|Місяць народження=
|Рік народження=
|Місце народження=
|День смерті=
|Місяць смерті=
|Рік смерті=
|Місце смерті=
|Місце поховання=
|Стать=
|Alma mater=
|Місце діяльності=
|Напрями діяльності=

```

Абрикосов, Олексій Іванович	
Рік народження	1875
Місце народження	Москва, Росія
Рік смерті	1955
Місце смерті	Москва, Росія
Alma mater	Московський державний університет, Москва
Напрями діяльності	медицина, патологічна анатомія

Рис. 3. Шаблон е-ВУЕ “Персоналія” та приклад його використання

Шаблон

Персоналія

Службова сторінка. Доступ тільки для адміністрації порталу е-ВУЕ

Відмова у доступі для незареєстрованих користувачів

Рис. 4. Відмова у доступі до контенту сторінки шаблон е-ВУЕ “Персоналія”

Висновки

Розглянуто перспективи та область застосування Wiki-технологій, проаналізовано структуру подання контенту та програмне забезпечення MediaWiki. Запропоновано абстрактну модель архітектури MediaWiki, яка формалізує відношення між основними складовими цього технологічного середовища. Визначено проблему, що пов'язана з необхідністю керування доступом до контенту Wiki-ресурсів відповідно до специфіки інформації, що міститься у таких ресурсах. Результати аналізу показали, що базові засоби MediaWiki не дозволяють отримати задовільне рішення цієї задачі. Тому виникає потреба у створенні спеціалізованого програмного забезпечення, яке базується на класифікації контенту з використанням окремих просторів імен, категорій, шаблонів та семантичних властивостей, які здобуваються з MediaWiki, та є незалежним від ядра MediaWiki, а базується на програмуванні аналізу контенту на рівні шкіни.

Наведене рішення апробовано у розробці порталу e-VUE [14] і використовує знання щодо структури бази знань цього порталу.

Література

1. MediaWiki. <https://www.mediawiki.org/wiki/MediaWiki>
2. Гришанова І.Ю., Рогушина Ю.В. Адаптація технологічних засад semantic mediawiki до потреб онлайн-версії великої української енциклопедії ВУЕ. *Традиції та сучасні концепти енциклопедичної справи в Україні*: колективна монографія / За ред. Киридон А.М. К.: Державна наукова установа «Енциклопедичне видавництво», 2018. С. 240–253. <https://ev.vue.gov.ua/wp-content/uploads/2019/11/Traditions.pdf>
3. Leuf B., Cunningham W. The Wiki way: collaboration and sharing on the Internet, 2001, <http://www.citeulike.org/group/13847/article/7659081>.
4. Wikipedia – <https://www.wikipedia.org>.
5. Рогушина Ю.В., Прийма С.М., Строкань О.В. Створення та використання семантичних Wiki-ресурсів: навчальний довідник. Мелітополь, ФОП Однорог Т.В., 2017. 169 с.

6. Manual: What is MediaWiki? https://www.mediawiki.org/wiki/Manual:What_is_MediaWiki%3F.
7. Brown A., Wilson G. The Architecture of Open Source Applications: Elegance, Evolution, and a Few Fearless Hacks, V. 1, 2011. <ftp://188.235.129.151/incoming/books/Wilson%20G.%20-%20The%20Architecture%20Of%20Open%20Source%20Applications%20-%202011.pdf>.
8. Manual: Skins. <https://www.mediawiki.org/wiki/Manual:Skins>.
9. Magnus Manske. https://en.wikipedia.org/wiki/Magnus_Manske
10. Авторизація. <https://uk.wikipedia.org/wiki/Авторизація>
11. Документація для розробників шкінів https://www.mediawiki.org/wiki/Manual:Skinning_Part_2
12. Рогушина Ю.В., Гришанова І.Ю. Онтологічна модель бази знань онлайн-версії «Великої української енциклопедії» та методи її застосування для семантичного пошуку та навігації. Енциклопедичний контент і виклики сучасного світу: Збірник матеріалів наукової конференції / За ред. Киридон А.М. К.: Державна наукова установа «Енциклопедичне видавництво», 2019. С. 69–74.
13. Гладун А.Я., Рогушина Ю.В. Репозитории онтологий как средство повторного использования знаний для распознавания информационных объектов. *Онтология проектирования*. 2013. № 1 (7). С. 35–50.
14. Велика українська енциклопедія. <https://vue.gov.ua/>

References

1. MediaWiki. <https://www.mediawiki.org/wiki/MediaWiki>
2. Grishanova I.Y., Rogushina J.V. (2018) Adaptation of technological means of Semantic Mediawiki for needs of online version of Great Ukrainian Encyclopedia // *Encyclopaedias in Ukraine: people, ideas, steps: collective monograph* / Ed. Kyrydon A.M., Kyiv, P.240-253. [in Ukrainian]
3. Leuf B.,Cunningham W. (2001) The Wiki way: collaboration and sharing on the Internet, 2001, <http://www.citeulike.org/group/13847/article/7659081>.
4. Wikipedia. <https://www.wikipedia.org>.
5. Rogushina Y.V., Priyma S.M, Strokany O.V. (2017) Creating and use of the Semantic Wiki

- resources: tutorial. - Melitopol, FOP Odinorog T.V. – 169 p. [in Ukrainian]
6. Manual: What is MediaWiki? https://www.mediawiki.org/wiki/Manual:What_is_MediaWiki%3F.
 7. Brown A., Wilson G. (2011). The Architecture of Open Source Applications: Elegance, Evolution, and a Few Fearless Hacks (V. 1). – <ftp://188.235.129.151/incoming/books/Wilson%20G.%20-%20The%20Architecture%20Of%20Open%20Source%20Applications%20-%20202011.pdf>.
 8. Manual: Skins. <https://www.mediawiki.org/wiki/Manual:Skins>.
 9. Magnus Manske. https://en.wikipedia.org/wiki/Magnus_Manske.
 10. Avtorization. <https://uk.wikipedia.org/wiki/Авторизація>.
 11. Documentation for skin developers. – https://www.mediawiki.org/wiki/Manual:Skinning_Part_2.
 12. Grishanova I.Y., Rogushina J.V. (2019) Ontological model of online version of Great Ukrainian Encyclopedia knowledge base and methods of its use for semantic search // Encyclopaedic content and challenges of modern world / Ed. Kyrydon A.M., Kyiv. P. 64–74. [in Ukrainian]
 13. Gladun A., Rogushina J. Ontology repositories as a means of knowledge reusing for recognition of information objects // Ontology for Design. 2013. N 1 (7). P. 35–50. [in Russian]
 14. Great Ukrainian Encyclopedia, <https://vue.gov.ua/>

Про авторів:

Гришанова Ірина Юріївна,
науковий співробітник.
Кількість наукових публікацій в українських виданнях – 19.
Кількість наукових публікацій в зарубіжних виданнях – 3.
<http://orcid.org/0000-0003-4999-6294>,

Рогущина Юлія Віталіївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 155.
Кількість наукових публікацій в зарубіжних виданнях – 31.
<http://orcid.org/0000-0001-7958-2557>.

Місце роботи авторів:

Інститут програмних систем
НАН України,
03181, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 066 550 1999.

E-mail: ladamandraka2010@gmail.com,
i26031966@gmail.com

Одержано 05.02.2020