

сучасного програмного забезпечення використовуються підходи, засновані на методах інспекції, які припускають перевірку відповідності ПЗ вимогам нормативних документів шляхом неформалізованого аналізу документації та процесів розробки; використанні спеціальних показників - метрик, що дозволяють побічно судити про рівень якості та надійності ПЗ на підставі аналізу ймовірно-вимірюваних характеристик програмного продукту і процесів його проектування; застосуванні математичних моделей для оцінки ймовірнісних показників надійності ПЗ.

Список використаних джерел

1. Дідковська М.В. Аналіз моделей оцінювання надійності програмного забезпечення // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка. - 2004. - № 41. - С. 103-120.
2. Рябишин И.А. Надежность и безопасность структурно-сложных систем. - Спб.: Политехника, 2000. - 248 с.

УДК 519.7:378.147

МЕТОД ТА АЛГОРИТМ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ WEB-КОНТЕНТУ

Шпінталь М.Я.¹⁾, Кирильчук А.Б.²⁾

Тернопільський національний економічний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

І. Постановка проблеми

Розвиток глобальної мережі Internet та інформаційних технологій дало можливість звичайним користувачам бути як споживачами інформації, але її генераторами і розповсюджувачами. Такий розвиток призвів до стрімкого зростання та хаотичної організації інформації в мережі. В Інтернеті присутня велика кількість знань та різної інформації, кількість не завжди означає якість. Це створює певні складності під час пошуку необхідної інформації. Для спрощення та підвищення ефективності процесу пошуку слід використовувати спеціальні технології для отримання корисних знань з мережі Internet. Важливими стають програми, які створені з використанням алгоритмів інтелектуального аналізу визначення релевантної інформації, достовірність якої може бути перевірена.

Переважна більшість сучасних технологій в своїй основі використовують статистичні та синтаксичні методи перевірки несуперечності та цілісності, розташованих в базі знань, це вказує на нездатність існуючих методів перевірки впоратися з неструктурованістю, різномірністю і суперечливістю інформації, представленої в Інтернеті.

II. Мета роботи

Метою роботи є дослідження алгоритмів інтелектуального аналізу контенту сайту та розробка засобів для отримання та перевірки знань у конкретній сфері діяльності.

Завдання:

- проаналізувати сучасні підходи до отримання даних та знань з глобальних сховищ;
- розробити алгоритм отримання релевантної інформації та знань;
- аналіз ефективності запропонованих алгоритмів.

III. Модель системи підвищення ефективності інтелектуального аналізу Web-контенту

У роботі був застосований нечіткий висновок для вирішення задачі класифікації сторінок web-сайту на підставі поточних даних про відвідування. Автор виходить з того, що по поведінці відвідувача на сайті можна судити про якість його навігаційної структури і контенту, а так саме відповідності представленої інформації запитам відвідувачів.

Для оцінки якості окремої сторінки були обрані два критерії (вхідні змінні): «відвідуваність» і «різниця між входами і виходами». Для змінної «відвідуваність» було вибрано три значення: «погана», «середня» і «хороша». Змінна «Входи і виходи» являє собою різницю між кількістю входів на сторінку на початку сесії і виходів з неї в кінці за період часу. Для її оцінки були вибрані три терма «погано», «середньо» і «добре». Терми задано відповідно Z, p і S образними функціями. Для вихідної змінної «якість сторінки» були обрані три терма: «хороше», «середнє» і «погане».

Була побудована база правил для вихідної змінної «якість сторінки». Для кожної конкретної сторінки виробляються вхідні змінні і обчислюються ступені їх приналежності до кожного з термів

відповідних лінгвістичних змінних. Обчислюється ступінь належності об'єкта класифікації класам з бази знань. Далі вибирається клас з максимальним ступенем приналежності.

В інтерфейсі системи результати роботи нечіткого висновку представлені у вигляді дерева сторінок, кожна сторінка забарвлюється в колір, відповідний її «якості». Таким чином, адміністратор сайту може відразу оцінити ситуацію за день або будь-який інший період часу.

Висновок

Запропонована методика є працездатною, вона дає можливість встановлювати якість Web-ресурсу залежно від поведінки користувачів на його сторінках.

Список використаних джерел

1. Головянко М. В. Методи і модель верифікації знань для інтелектуалізації Web-контенту; Автореф. дисс. канд. техн. наук. – Х., 2011. – 129 с.
2. Intelligent Miner for Text (IBM). [Електронний ресурс]. – Режим доступу: http://www.research.ibm.com/trl/projects/textmining/index_e.htm

УДК 004.4'232

ІНТЕЛЕКТУАЛЬНІ ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО КОДУ

Шпінталь М.Я.¹⁾, Піхо І.І.²⁾

Тернопільський національний економічний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

I. Постановка проблеми

У процесі написання програмного коду або скриптів, з метою реалізації певного алгоритму певною мовою програмування, насамперед потрібен зручний і практичний редактор коду. Багато редакторів коду надають функції, які особливо корисні для програмістів. Підсвічування синтаксису і автоматичні відступи – це, мабуть, найважливіші з інструментів такого роду. Вони дозволяють з одного погляду зрозуміти, що введений код в загальних рисах коректний, що всі блоки правильно вкладені один в один і не містять очевидних помилок. Тому кодування є частиною програмування, поряд з аналізом, проектуванням, компіляцією, тестуванням, налагодженням і супроводом.

II. Мета роботи

Метою магістерської роботи є дослідження можливостей і функцій сучасних популярних редакторів коду та проектування і розробка вдосконаленого, зручного редактора коду, який пристосовуватиме користувача до відомих інтегрованих середовищ розробки або Integrated Development Environments (IDE).

Завдання:

- Дослідити можливості і функції популярних редакторів коду.
- Проаналізувати сучасні технологічні засоби для максимальної відповідності функціональним вимогам.
- Розробити зручний і багатофункціональний редактор коду.

III. Структура текстового редактора

Код програми редактора розділений на декілька модулів: AdvTextBox.cs, AssemblyInfo.cs, Block.cs, BlockList.cs, Document.cs, Group.cs, GroupList.cs, Log.cs, MainForm.cs, Parser.cs.

У файлі AdvTextBox.cs описаний клас AdvTextBox, об'єктом якого є поле для редагування тексту програм. У цьому класі перевантажена процедура обробки повідомлень.

Основна інформація про програму зберігається у файлі AssemblyInfo.cs. У файлі Block.cs описаний клас Block, який використовується для зберігання блоків тексту. Файл BlockList.cs містить опис масиву об'єктів класу Block. У файлі Document.cs реалізований пошук блоків тексту, в яких були зроблені зміни. Потім ці блоки передаються автоматично Parser для отримання остаточної розмальовки. У файлі Group.cs описаний клас Group, який служить для зберігання ключових слів. Клас Log, який описаний у файлі Log.cs, використовується для протоколювання дій автомата. У файлі MainForm.cs описаний основний клас MainForm створення графічного інтерфейсу. Автомат