«Системні технології» 1 (138) 2022 «System technologies»
DOI 10.34185/1562-9945-1-138-2022-04
УДК 004.5+ 004.78+ 004.823

О.С. Куроп'ятник

МОДЕЛЮВАННЯ СЦЕНАРІЮ ДІАЛОГУ ДЛЯ СИСТЕМИ ВИЯВЛЕННЯ ЗАПОЗИЧЕНЬ НА ОСНОВІ КОЛЬОРОВОЇ МЕРЕЖІ ПЕТРІ

Анотація. Виконання принципів простоти та дружності інтерфейсу користувача дозволяє зменшити час навчання роботі з програмою та виконання основних задач з її допомогою. Розробка та формалізація сценарію діалогу дозволяє реалізувати дані принципи в ході розробки програми. У роботі визначено основні засади побудови моделі сценарію діалогу з використанням апарату мережі Петрі. На їх основі виконано побудову сценарної моделі діалогу для системи виявлення запозичень та реалізовано інтерфейс користувача. Визначено основні переваги побудови моделі на основі мережі.
Ключові слова: сценарій діалогу, інтерфейс користувача, мережа Петрі.

Постановка проблеми. Академічна доброчесність є одним з ключових факторів забезпечення якості вищої освіти. В період пандемії COVID-19 стала поширеною дистанційна форма навчання, а тому зберігається тенденція несамостійного виконання навчальних завдань. Серед нових чинників, що сприяють цьому, – повна або часткова відсутність візуального контролю за процесом виконання завдань з боку викладача.

Актуальною залишається розробка, удосконалення та впровадження програмного забезпечення (ПЗ), що дозволяє виконувати пакетну перевірку навчальних з можливістю редагування бази та має простий та дружній інтерфейс користувача (ІК).

В даній роботі визначено основні засади моделювання сценарію діалогу як частини проекту ІК на основі кольорової мережі Петрі. Представлено результати розробки інтерфейсу користувача на основі побудованої сценарної моделі.

Аналіз останніх досліджень і публікацій. На сьогодні існує чимало програмних рішень для виявлення текстових запозичень [1, 2]. При їх розробці особливу увагу приділяють методам та алгоритм виявлення запозичень [3, 4], характеристикам їх роботи [5, 6]: часу, точності, відсутності хибних спрацювань тощо. Однак з огляду на застосування в академічному середовищі

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

важливими також є питання технології та зручності використання, що безпосередньо пов'язані з ІК.

Проектування ІК складається з багатьох етапів, серед яких розробка сценарію діалогу та ескізів форм, обрання структури діалогу, визначення засобів зворотного зв'язку тощо. Саме розробка та тестування сценарію діалогу дозволяють уникати складності у виборі операцій, станів програми без виходу та відсутності необхідних даних при вже виконаному переході до деякої операції.

Сценарій діалогу може бути представлено у словесному вигляді, проте така форма може містити неоднозначності. На противагу їй доцільним є використання певних формалізмів: графа діалогу [7], автоматів [8], мереж Петрі [8, 9, 10], UML-діаграм прецедентів, діяльності та станів.

Мета дослідження. Метою роботи є інтерпретація складових мережі Петрі для моделювання сценаріїв діалогу. Задачами є: визначення семантики компонентів кольорової мережі Петрі в термінах складових сценарію діалогу та процесу роботи програми; побудова сценарної моделі діалогу та реалізація на її основі ІК системи виявлення запозичень.

Проектування інтерфейсу користувача засобами кольорових мереж Петрі. Для побудови простого та дружнього інтерфейсу користувача доречним є:

– зменшення варіативності дій на кожному етапі роботи з програмою;

– уникнення станів у роботі програми, в яких відсутні переходи до наступних чи попередніх, при цьому вони не є заключними – «глухі кути»;

– контроль за введенням даних.

Для цього необхідно визначити послідовність дій при роботі з програмою та обрати відповідну структуру діалогу. Визначення дій виконується при створенні сценарію діалогу. Під сценарієм будемо розуміти опис станів та переходів, в яких перебуває система та користувач в рамках вирішення конкретного завдання.

Для моделювання сценарію визначимо кольорову мережу Петрі у загальному вигляді

$$N' = \langle P', T', F', S', G', M'_0 \rangle, \quad (1)$$

де P' – скінченна множина позицій, що відповідає станам програми і характеризується множиною значень вхідних та вихідних параметрів та діями з їх перетворення, T' – скінченна множина переходів, які позначають набір дій користувача у певному його стані, якими зумовлений перехід від стану p_i до p_j,

$p_i, p_j \in P'$, F' – функція переходів, S' – множина типів (аналог кольорів), $G' = \{g_k\}$ – множина охоронних умов на переходах, k – номер переходу, M'_0 – початкова розмітка.

При проектуванні ІК типи фішок позначають різні множини вхідних та вихідних даних. В графічному представлені моделі вони можуть бути позначені різними геометричними фігурами або символами, при описі розміток – буквами, які, наприклад, є першими у назві використаних фігур або типів даних, або цифрами.

$G' = \{g_k\}$ контролюють наявність даних у програмі для виконання операцій в певному стані, в моделі – вказують умову активації переходу.

Для визначення елементів множини G у загальному виді використаємо форму Бекуса-Наура:

$$g_k := e(p_i, s_j, n)\{\&e(p_l, s_k, m)\}, \tag{2}$$

де $p_i, p_l \in P'$, $s_j, s_k \in S'$, $n, m \in \mathrm{N}$ – необхідна кількість фішок заданого типу у позиції, $e(p, s, n)$ – функція, побудована за аналогією до *e(p)* [9], для перевірки наявності фішки типу $s \in S$ у позиції *p:*

$$e(p, s, n) = \begin{cases} true, якщо \exists M : \sum\limits_{s \in S} M(p, s) \geq n \\ false, якщо \exists M : \sum\limits_{s \in S} M(p, s) < n' \end{cases} \tag{3}$$

де M – деяка розмітка мережі, $\sum\limits_{s \in S} M(p, s) \geq n$ – вказує на наявність у розмітці M щонайменше *n* фішок типу s у позиції *p*.

Сценарій діалогу моделюється як мережа Петрі, в якій позиції позначають стани програми, що характеризуються визначенням даних, дії моделюються переходами. Завдяки використанню типів фішки не лише визначають наявність, а й специфікують набір даних, необхідний для переходу до наступної дії в сценарії. Функція переходів F' дозволяє відслідкувати наявність станів, з яких немає виходів, а також дані, доступні в кожному стані програми.

Сценарна модель діалогу для системи виявлення запозичень. Одним зі способів зменшення часових витрат на виявлення текстових запозичень є реалізація принципів дружності та простоти в ІК відповідної системи. Це дозволить покращити кількісні показники часу навчання роботі з програмою та часу, витраченого на виконання основних задач.

Оскільки складність організації ІК залежить від функціоналу, визначимо основні функціональні вимоги до системи виявлення запозичень:

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

– завантаження та попередня обробка пакету для перевірки. Під пакетом будемо розуміти набір файлів doc/docx формату на однакову тему (з подібним змістом), що мають інформативні назви;

– формування бази робіт: завантаження, попередня обробка текстів для порівняння та представлення їх у графовому вигляді [11];

– порівняння вхідних текстів пакету та текстів бази, визначення відсотку запозичень: за кожною роботою з бази та загального;

– встановлення параметрів:

– порівняння (мінімальна довжина фрагмента, який враховується як запозичений);

– виведення (мінімальний відсоток запозичень у роботі);

– сортування (робота, за спаданням відсотку запозичень у якій виконується сортування);

– перегляд результатів порівняння.

Основний сценарій діалогу передбачає такі дії з боку користувача:

1. обрати файли для перевірки – обрати пакет;

2. обрати файли, з якими буде вестися порівняння – обрати базу;

3. встановити параметри та порівняти роботи;

4. задати поріг запозичень та переглянути результати порівняння;

5. відсортувати та переглянути результати порівняння.

Допускаються такі варіації сценарію:

– обрання нових файлів для перевірки та/або бази;

– зміна параметрів порівняння або допустимого порогу запозичень;

– сортування за новим критерієм.

Побудуємо сценарну модель діалогу для реалізації функціональних вимог на основі мережі Петрі (рис. 1) та визначимо відповідність елементів сценарію (станів та переходів) елементам мережі:

$$N = \langle P, T, F, S, G, M_0 \rangle, \tag{4}$$

де P, T – скінченні множини позицій та переходів, F – функція переходів, задана деревом розміток (рис. 2), $S = \{c, d, s, t, r\}$ – множина типів, $G = \{g_k\}$ – множина охоронних умов на переходах, k – кількість переходів, $M_0 = \langle cd, 0, 0, s, 0, 0 \rangle$ – початкова розмітка, де буквами позначено наявність фішок відповідного типу, 0 – ознака відсутності фішок у позиції.

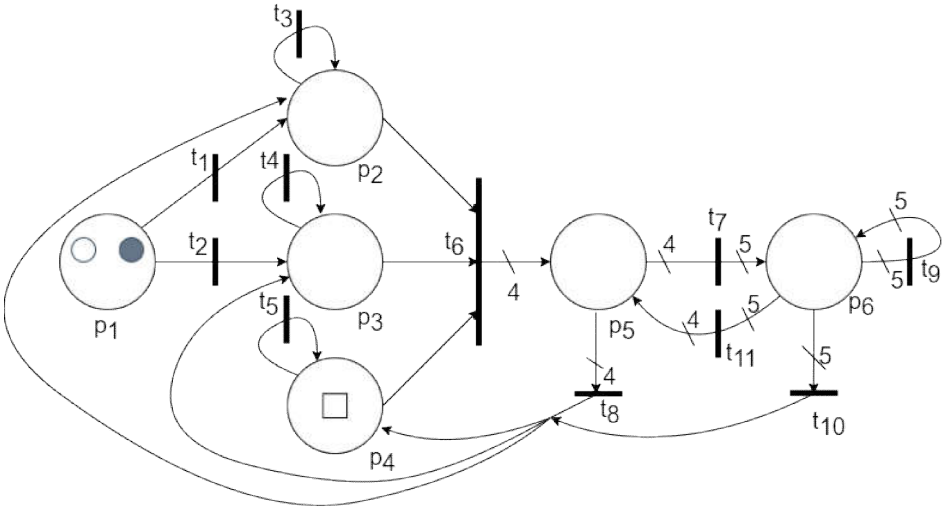

Рисунок 1 – Мережа Петрі для опису сценаріїв діалогу

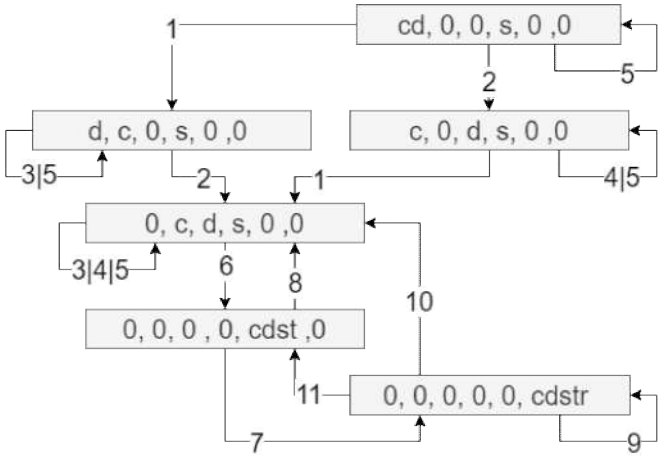

Рисунок 2 – Дерево розміток

Семантичну природу типів визначають множини вхідних та вихідних даних програми, так: *c* (коло)– вхідні тексти, що будуть перевірятися на наявність запозичень; *d* (диск, або круг) – тексти бази, з якими буде вестися порівняння; *s* (квадрат) – параметри порівняння; *t* (трикутник) – результати порівняння текстів у вигляді матриці $R = \{res_{ij}\}$ та вектора $UR = \{ur_i\}$, $i = \overline{1,n}$, $j = m$, res_{ij} – відсоток *i*-ого тексту, запозичений з *j*-ого, ur_i – загальний відсоток запозичень у *i*-ому тексті, n – кількість текстів, поданих на перевірку, m – кількість текстів у базі; *r* (ромб) – підмножини елементів матриці $\{res_{ij}\}$ та вектора $\{ur_i\}$, $ur_i \geq l$, l – мінімальний відсоток запозичень у тексті, та назви файлів, що містять відповідні тексти.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Вхідними даними також є поріг запозичень та критерій сортування, проте їх зміна призводить лише для формування підмножин вихідних даних, тому в даній моделі вони не враховуються як окремі фішки.

Оскільки у випадку, що розглядається, відсутня необхідність використання двох та більше однакових наборів даних, спростимо функцію (3), записавши її у вигляді:

$$e(p,s)=\begin{cases} true, якщо \exists M : \sum\limits_{s\in S} M(p,s) > 0 \\ false, якщо \exists M : \sum\limits_{s\in S} M(p,s) = 0' \end{cases} \quad (5)$$

де всі умовні позначення відповідають (3).

Тоді множина охоронних умов G має вигляд:

$$G=\{g_1=e(p_1,c);\ g_2=e(p_1,d);\ g_3=e(p_2,c);\ g_4=e(p_3,d);\ \ g_5=e(p_4,s);$$
$$g_6=e(p_2,c)\,\&\,e(p_3,d)\,\&\,e(p_4,s);\ g_7=g_8=e(p_5,c)\,\&\,e(p_5,d)\,\&\,e(p_5,s)\,\&\,e(p_5,t);$$
$$g_9=g_{10}=g_{11}=e(p_6,c)\,\&\,e(p_6,d)\,\&\,e(p_6,s)\,\&\,e(p_6,t)\,\&\,e(p_6,r)\},$$

де g_i є умовою спрацювання переходу t_i.

Специфікуємо позиції мережі та відповідні стани програми:

– p_1 – програму запущено, наявні множини вхідних даних, що позначають фішки в позиції;

– p_2, p_3 – зчитування, обробка та завантаження в оперативну пам'ять у форматі рядків вхідних текстів робіт, які будуть перевірятися на наявність запозичень, та побудова графів текстів, з якими буде вестися порівняння, відповідно;

– p_4 – завдання значень параметрів порівняння. Фішка в позиції присутня в M_0, оскільки в програмі задані значення за замовченням;

– p_5 – порівняння вхідних текстів та текстів у вигляді графів, обчислення відсоток запозичень;

– p_6 – виведення на форму відсортованого переліку робіт, у текстах яких відсоток запозичень не перевищує заданий поріг.

Специфікуємо переходи:

– t_1 – визначення джерела файлів робіт, тексти яких будуть перевірятися на наявність запозичень, та подання запиту на їх зчитування та попередню обробку;

– t_2 – визначення джерела файлів робіт, з текстами яких буде виконуватися порівняння, та подання запиту на їх зчитування, попередню обробку та побудову для них графового представлення;

– t_3, t_4 – подання запиту на оновлення (визначення нових) робіт для перевірки та бази відповідно. Аналогічно до t_1, t_2 з запитом на очищення оперативної пам'яті, яку займали рядки раніше зчитаних текстів;

– t_5 – введення параметрів порівняння;

– t_6 – подання запиту на порівняння та обчислення відсотку запозичень;

– t_7 – введення значення порогу та подання запиту на формування переліку робіт, в яких відсоток запозичень перевищує заданий поріг або рівний йому;

– t_8, t_{10} – зміна вхідних текстів або бази, або параметрів порівняння;

– t_9 – подання запиту на сортування переліку робіт;

– t_{11} – зміна порогу для виведення результатів.

Основний сценарій моделюється послідовним спрацюванням переходів t_1, t_2, t_6, t_7, t_9. Наявність петель, зворотних дуг та альтернативних переходів (рис. 1) дозволяє реалізовувати варіативність та повторне виконання будь-якого сценарію, передбаченого в моделі. Варіативність сценарію також проілюстровано в моделі наявністю розгалужень у дереві розміток (рис. 2), в тому числі позначених записами типу $i|j$, що вказує на альтернативність спрацювання переходів з відповідними номерами.

Аналіз мережі. Дослідимо властивості мережі (1) з інтерпретацією на процес роботи та взаємодії з програмою. Мережа є обмеженою, максимальна кількість фішок не перевищує п’яти. Кількість фішок обумовлена кількістю множин вхідних та вихідних даних програми. Мережа є безпечною відносно кожного типу $s \in S$. Оскільки фішки визначають наявність даних, то безпечність мережі вказує на відсутність дублікатів. Мережа не є зберігаючою за типами t, r, оскільки фішки вказують на наявність вихідних даних. Отримані результати R та UR втрачають свою актуальність при зміні вхідних даних, що пояснює невиконання даної властивості. Властивість досяжності виконується для розміток, зазначених на рис. 2, таким чином мережа моделює всі сценарії діалогу, описані вище. Властивість живості виконується, що говорить про відсутність взаємних блокувань та «глухих кутів».

Реалізація інтерфейсу користувача на основі сценарної моделі. За моделлю, описаною вище, розроблено ІК (рис. 3).

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

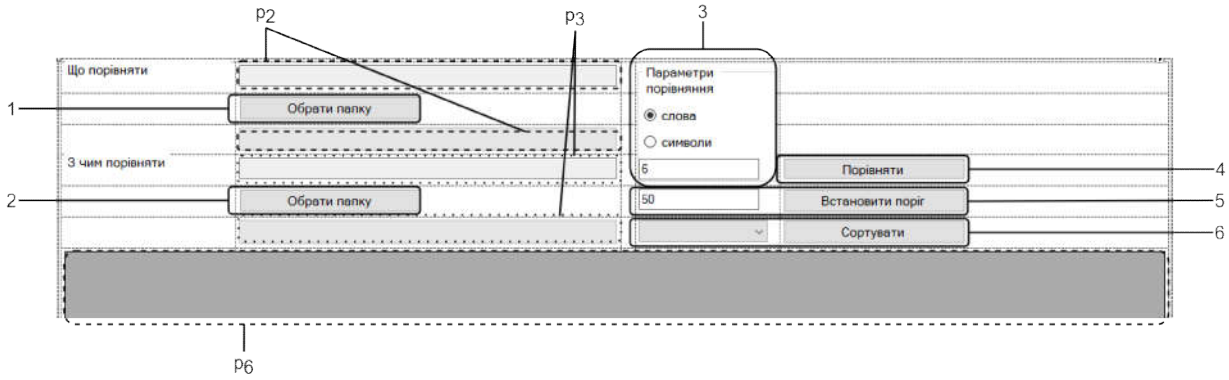

Рисунок 3 – Розроблена форма інтерфейсу користувача

Визначимо відповідність елементів реалізованого інтерфейсу елементам моделі (4):

– p_1 – відкрите вікно, для взаємодії доступні елементи форми 1 – 3;

– p_2, p_3 – поля та progress bars (елементи p2, p3) відображають місце розташування файлів з текстами та прогрес виконання операції зчитування та їх обробки;

– p_4 – задані параметри порівняння відображені у групі полів 3;

– p_5 – спливаюче повідомлення про завершення операції порівняння;

– p_6 – на формі в елемент p6 виведено відсортований перелік робіт, в яких відсоток запозичень не перевищує заданий поріг;

– t_1 та t_3, t_2 t_4 – кнопки 1 і 2 відповідно та їх обробники подій onclick;

– t_5 – введення з клавіатури параметрів порівняння у поля групи 3;

– t_6, t_{11} – кнопка 4 та її обробники події onclick;

– t_7 – група елементів форми 5 та їх обробники події onclick;

– t_8, t_{10} – доступність для взаємодії елементів форми 1 – 3;

– t_9 – група елементів форми 6 та їх обробники події onclick.

Висновки. Інтерпретація складових мережі Петрі в контексті проектування ІК та роботи програми дозволило виконати побудову сценарної моделі діалогу для системи виявлення запозичень. На основі моделі було реалізовано віконний інтерфейс програми. Побудова моделі дала формальне представлення сценарію, що зменшило неоднозначність його розуміння в процесі реалізації ІК.

Застосування апарату кольорової мережі при розробці ІК є корисним для визначення:

– даних, які необхідні для подальшої роботи програми, що дає можливість обґрунтувати перелік елементів інтерфейсу та їх доступність для регулювання дій користувача;

– даних, що присутні в програмі на кожному етапі її роботи. Аналіз мережі дозволяє виявити їх дублікати;

– дій програми, що виконуються паралельно, та необхідного моменту їх синхронізації;

– можливих станів, що є «глухими кутами».

ЛІТЕРАТУРА

1. Shynkarenko V. I. Plagiarism detection problems and analysis software tools for its solve / V. I Shynkarenko, O. S. Kuropiatnyk //Science and Transport Progress. Bulletin of Dnipropetrovsk National University of Railway Transport. – 2017. – №. 1 (67). – P. 131-142. DOI: https://doi.org/10.15802/stp2017/94034

2. Chowdhury H. A. Plagiarism: Taxonomy, tools and detection techniques / H. A. Chowdhury, D. K. Bhattacharyya //arXiv preprint arXiv:1801.06323. – 2018.

3. Foltýnek T. Academic plagiarism detection: a systematic literature review / T. Foltýnek, N. Meuschke, B. Gipp //ACM Computing Surveys (CSUR). – 2019. – Vol. 52. – №. 6. – P. 1-42. DOI: https://doi.org/10.1145/3345317

4. Sabeeh M. Plagiarism Detection Methods and Tools: An Overview / M. Sabeeh, F. Khaled //Iraqi Journal of Science. – 2021. – P. 2771-2783. DOI: https://doi.org/10.24996/ijs.2021.62.8.30

5. Belyy A. Improved evaluation framework for complex plagiarism detection / A. Belyy, M. Dubova, D. Nekrasov//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). – 2018. – P. 157-162.

6. Nichols L. et al. Syntax-based improvements to plagiarism detectors and their evaluations //Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education. – 2019. – P. 555-561. DOI: https://doi.org/10.1145/3304221.3319789

7. Зюзин А. В. и др. Методика формирования адаптивного сценария диалога при решении автоматизированных задач управления на рабочем месте комплекса средств автоматизации военного назначения //Наукоемкие технологии в космических исследованиях Земли. – 2021. – Т. 13. – №. 3. – С. 36-47.

8. Верлань А. Ф. Две модели вопросно-ответной «машины диалога» / А. Ф. Верлань, І. О. Чмир //Математичне та комп'ютерне моделювання. Серія: Технічні науки. – 2020. – С. 25-42.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

9. Алгазинов Э. К. Построение системы ведения диалога на основе расширения сетей Петри / Э. К. Алгазинов, А. А. Жижелев, А. Р. Нехаев //Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2011. – №. 2. – С. 136-143.

10. Пантелеев Е. Р. Моделирование сценариев действий пользователя в среде САПР на базе сетей Петри / Е. Р. Пантелеев, В. А Зуйко. //Вестник Ивановского государственного энергетического университета. – 2015. – №. 5.

11. Kuropiatnyk O. S. Constructive and object-oriented modeling text for detection of text borrowings / O. S. Kuropiatnyk //System technologies. – 2019. – Vol. 4. – №. 123. – P. 34-47. DOI: 10.34185/1562-9945-4- 123-2019-04

REFERENCES

1. Shynkarenko V. I. Plagiarism detection problems and analysis software tools for its solve / V. I Shynkarenko, O. S. Kuropiatnyk //Science and Transport Progress. Bulletin of Dnipropetrovsk National University of Railway Transport. – 2017. – №. 1 (67). – P. 131-142. DOI: https://doi.org/10.15802/stp2017/94034

2. Chowdhury H. A. Plagiarism: Taxonomy, tools and detection techniques / H. A. Chowdhury, D. K. Bhattacharyya //arXiv preprint arXiv:1801.06323. – 2018.

3. Foltýnek T. Academic plagiarism detection: a systematic literature review / T. Foltýnek, N. Meuschke, B. Gipp //ACM Computing Surveys (CSUR). – 2019. – Vol. 52. – №. 6. – P. 1-42. DOI: https://doi.org/10.1145/3345317

4. Sabeeh M. Plagiarism Detection Methods and Tools: An Overview / M. Sabeeh, F. Khaled //Iraqi Journal of Science. – 2021. – P. 2771-2783. DOI: https://doi.org/10.24996/ijs.2021.62.8.30

5. Belyy A. Improved evaluation framework for complex plagiarism detection / A. Belyy, M. Dubova, D. Nekrasov//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). – 2018. – P. 157-162.

6. Nichols L. et al. Syntax-based improvements to plagiarism detectors and their evaluations //Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education. – 2019. – P. 555-561. DOI: https://doi.org/10.1145/3304221.3319789

7. Zyuzin A. V. et al. Methodology for the formation of an adaptive script for dialogue in solving automated control tasks at the workplace of a complex of military automation equipment // Science-intensive technologies in space research of the Earth. – 2021. – Vol. 13. – No. 3. – P. 36-47.

8. Verlan A. F. Two models of question-answer "dialogue machine" / A. F. Verlan, I. O. Chmir // Mathematical and computer modeling. Series: Technical Sciences. – 2020. – P. 25-42.

9. Algazinov E. K. Building a dialogue system based on the expansion of Petri nets / E. K. Algazinov, A. A. Zhizhelev, A. R. Nekhaev // Bulletin of the Voronezh State University. Series: System Analysis and Information Technology. – 2011. – No. 2. – P. 136-143.

10. Panteleev E. R. Modeling of scenarios of user actions in a CAD environment based on Petri nets / E. R. Panteleev, V. A. Zuiko. // Bulletin of the Ivanovo State Power Engineering University. – 2015. – No. 5.

11. Kuropiatnyk O. S. Constructive and object-oriented modeling text for detection of text borrowings / O. S. Kuropiatnyk //System technologies. – 2019. – Vol. 4. – №. 123. – P. 34-47. DOI: 10.34185/1562-9945-4- 123-2019-04

Received 06.01.2022.
Accepted 12.01.2022.

Modeling of dialogue scenario for the text borrowing detection system based on the coloured petri net

There are many software solutions for detecting text borrowings. In their development, special attention is paid to methods and algorithms for detecting borrowings, the characteristics of their work. However, technology and usability issues that are directly related to the user inter face (UI) are also important.

UI design consists of many stages, including the development of a dialogue scenario. The development and testing of a dialogue scenario avoids the difficulty of choosing operations, the state of the program without exit and the lack of necessary data in the already completed transi tion to an operation.

The dialogue scenario can be presented in verbal or formalized form: dialogue graph, au tomata, Petri nets, UML diagrams of precedents, activities and states.

The aim of this paper is to interpret the components of the Petri net to model dialogue scenarios. The tasks are: to determine the essence and purpose of the components of the color Petri net in terms of the components of the dialogue scenario and the process of the program work; construction of a scenario model of dialogue and implementation user interface for the text borrowings detection system of on its basis.

To model the scenario, a general coloured Petri net is defined. Its set of positions corre sponds to the states of the program and is characterized by the set of values of input and output parameters and actions for their transformation. The set of transitions indicates the set of user actions in a certain state. The set of security conditions on transitions control the availability of data in the program to perform operations in a certain state. The rules of formation of these conditions are determined in the work. Chip types denote different sets of input and output data.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

This interpretation of the net was used to build a scenario model of the dialogue of the text borrowing detection system. Analysis of network properties showed its adequacy of the model to the object of modeling.

The results of the implementation of the user interface according to the developed model are presented.

The conclusions form the main advantages of using coloured Petri nets to model a dialogue scenario.

Куроп'ятник Олена Сергіївна – к.т.н., доцент кафедри комп'ютерних інформаційних технологій, Український державний університет науки та технологій.

Kuropiatnyk Olena – Candidate of Engineering Sciences, Associate Professor of the Computer Information Technology Department, Ukrainian State University of Science and Technologies.