На правах рукописи

Луцив Дмитрий Вадимович

ПОИСК НЕТОЧНЫХ ПОВТОРОВ В ДОКУМЕНТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

05.13.11 — Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Санкт-Петербург
2018

Работа выполнена на кафедре системного программирования Санкт-Петербургского государственного университета

Научный руководитель:

КОЗНОВ Дмитрий Владимирович, доктор технических наук, доцент, профессор кафедры системного программирования

Официальные оппоненты:

ВОДЯХО Александр Иванович, доктор технических наук, профессор, профессор кафедры вычислительной техники, Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В.И. Ульянова (Ленина)»

ДРОБИНЦЕВ Павел Дмитриевич, кандидат технических наук, доцент, доцент высшей школы программной инженерии, Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский политехнический университет имени Петра Великого»

Ведущая организация:

Федеральное государственное учреждение «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук» (ИПМ им. М.В. Келдыша РАН)

Защита состоится 17 мая 2018 г. в 16:30 на заседании диссертационного совета Д 212.232.51 на базе Санкт-Петербургского государственного университета по адресу: 198504, Санкт-Петербург, Старый Петергоф, Университетский пр. 28, математико-механический факультет СПбГУ, ауд. 405.

С диссертацией можно ознакомиться в библиотеке Санкт-Петербургского государственного университета по адресу: 199034, Санкт-Петербург, Университетская наб., д. 7/9, а также на сайте: https://disser.spbu.ru/disser/soiskatelyu-uchjonoj-stepeni/dis-list/details/14/1598.html.

Автореферат разослан _____ 2018 года.

Учёный секретарь
диссертационного совета Демьянович Юрий Казимирович

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. Документация современного программного обеспечения (ПО) имеет значительные объёмы, сложную структуру и длительный жизненный цикл. При разработке и сопровождении документации в неё вносится большое количество точечных изменений, часто документация разрабатывается коллективом авторов, состав которого со временем меняется. Все это приводит к большому количеству ошибок и рассогласованностей, а также к нарушению единого стиля, то есть к деградации качества документации. В качестве примера можно указать описание ядра ОС Linux (Linux Kernel Documentation): проблемы качества этой документации в последнее время активно обсуждаются в Linux-сообществе. Одним из известных способов поддержки документации является стандартизация и повторное использование. Стандартизация означает создание и использование шаблонов и соглашений. Повторное использование документации подразумевает выделение переиспользуемых свойств программного обеспечения (тестов, требований, функциональности модулей, методов и пр.) и унификацию соответствующих фрагментов документации. Это, в свою очередь, облегчает сопровождение документации, так как изменения вносятся один раз в определение текстового фрагмента и далее автоматически попадают во все соответствующие места текста.

Следует отметить важность неточных повторов — текстовых фрагментов, незначительно отличающихся друг от друга. К их появлению приводит широко используемый при разработке документации приём copy/paste: вначале фрагмент текста копируется, затем каждая копия как-то по-своему изменяется. Учёт неточных повторов может предоставить новые возможности для стандартизации и повторного использования (работа E. Juergens с коллегами, работа M. Oumaziz с коллегами), в том числе для параметрического повторного использования (работы M. Oumaziz с коллегами, К. Романовского и Д. Кознова, M. Horie с коллегами, M. Nosál и J. Porubän).

Как при стандартизации существующей документации, так и при наладке повторного использования, важным оказывается поиск повторяющихся текстовых фрагментов. Такой поиск востребован также при формальной обработке

документации, например, в процессе выделения формальных требований в техническом задании для последующей генерации тестов. При этом, если для поиска точных повторов могут быть использованы различные существующие подходы, например, методы поиска клонов в ПО, то для поиска неточных повторов требуются специальные алгоритмы и методы. На настоящий момент такие подходы отсутствуют.

Таким образом, поиск неточных повторов в документации ПО является актуальной научно-практической задачей, решение которой способно существенно повысить качество сложной документации ПО путём предоставления различных сервисов по анализу и сопровождению документации. Также требуют изучения вопросы использования алгоритмов поиска неточных повторов в процессе реструктуризации и улучшения документации ПО.

Степень разработанности темы работы. Существуют работы, предлагающие методы разработки документации с применением повторного использования: M. Oumaziz с коллегами, M. Horie с коллегами, M. Nosál и J. Porubän, К. Романовский и Д. Кознов. Следует также отметить модульный способ организации документации в технологии DITA. Работы в этой сфере концентрируются вокруг форматов и языков разработки документации, разделяющих внутреннее и финальное представления документации – DITA, DocBook, JavaDoc, Tex, ReStructuredText, Markdown и др. Такие форматы активно используются на практике, поскольку позволяют решать задачу единого представления документации (single source) и автоматически порождать по этому представлению различные выходные документы, а также ввиду того, что технологии, основанные на этих форматах, обеспечивают надёжность при вёрстке больших документов. Методы повторного использования являются одним из основных источников требований к исследованиям в области автоматизированного поиска неточных повторов в документации ПО.

Существует значительное количество эмпирических исследований, посвящённых повторам в различной программной документации: E. Juergens с коллегами исследуют повторы в спецификациях требований, M. Oumaziz с коллегами — в API-документации, M. Nosál и J. Porubän — во внутренней документации (internal documentation). A. Wingkvist с коллегами исследовали повторы в документации ПО с точки зрения управления дальнейшей разработкой документации. Однако во всех этих

исследованиях рассматривались лишь точные повторы, хотя и отмечалась необходимость работы с неточными. Требуется также отметить отсутствие формализации понятия «неточный повтор»: общая концепция повторно используемого контента была разработана ещё в конце 90-х P. Basett, но с тех пор были предприняты лишь незначительные дальнейшие шаги — см. работы M. Nosál и J. Porubän, в которых, тем не менее, формальное определение неточных повторов отсутствует. Кроме того, осталась в стороне задача унификации большого класса документации программного обеспечения — так называемой reference-документации (различные справочники, руководства пользователей, API-документация и т.д.). Данный вид документации предложил в 2011 году D. Parnas.

Существующие методы поиска повторов в документации основываются преимущественно на технике поиска клонов в ПО: с этой целью используются готовые клон-детекторы, незначительно модифицированные для работы с документацией. Тем не менее следует отметить, что данный подход не позволяет эффективно искать неточные повторы в документации ПО, поскольку использует поиск по синтаксическому дереву программы. Кроме того, такой подход приводит к большому количеству ложноположительных срабатываний (false positives), а также низкому качеству найденных повторов (одна из основных проблем здесь — игнорирование структуры документа). S. Wagner, и D. Fernández пишут о преимуществах методов и средств обработки естественных языков (natural language processing) при формальной обработке текстовой информации программных проектов, но не приводят готовых методов для поиска повторов. Также следует отметить, что до сих пор остались не исследованными вопросы семантического анализа повторов в документации ПО.

Итак, необходима формальная модель нечетких повторов, ориентированная на разработку соответствующих алгоритмов, а также эффективные подходы для поиска неточных повторов в документации ПО, которые обеспечили бы учет семантической информации, а также были бы максимально свободны от ложноположительных срабатываний. Также требуется детально исследовать семантику повторов документации различного вида. Наконец, необходимы методы использования найденных повторов при улучшении документации, а также стыковка таких методов с подходами к повторному использованию документации.

Объектом исследования диссертационной работы являются алгоритмы поиска точных и неточных повторов в документации ПО, подходы к сопровождению и улучшению документации, модели и методы разработки программного обеспечения, языки разметки документации (markup languages), технологии и программные средства для разработки документации.

Целью данной работы является создание алгоритмов и методов для поиска неточных повторов в документации программного обеспечения с целью повышения качества документации в процессе её сопровождения. Для достижения этой цели был сформулирован ряд следующих **задач**.

1. Формализовать понятие неточного повтора в документации ПО и разработать эффективные алгоритмы поиска таких повторов.
2. Создать метод улучшения документации на основе поиска повторов.
3. Выполнить программную реализацию алгоритмов и методик, апробировать результаты исследования на реальной документации ПО.

Постановка цели и задач исследования соответствует следующим пунктам паспорта специальности 05.13.11: модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования (пункт 1); оценка качества, стандартизация и сопровождение программных систем (пункт 10).

Методология и методы исследования. Методология исследования базируется на идеях и подходах программной инженерии, нацеленных на разработку эффективных методов создания ПО и документации.

В работе использована концепция архетипа и дельт, а также концепция настраиваемых фреймов для повторного использования вариативного контента (P. Bassett). В качестве базового средства поиска повторов был применен метод поиска клонов в ПО, использована структура данных под названием интервальное дерево (interval tree) для быстрого поиска пересекающихся целочисленных интервалов, а также редакционное расстояние между строками для вычисления степени сходства тестовых фрагментов. Применена известная метафора тепловой карты (heat map) для визуализации информации о найденных повторах. В качестве средств программной реализации использовались языки Python и Java.

Положения, выносимые на защиту.

1. Предложена формальная модель неточных повторов в программной документации, разработан алгоритм поиска неточных повторов в документации ПО на основе компоновки точных повторов, найденных с помощью метода поиска точных клонов ПО. Доказана корректность алгоритма.
2. Создана методика интерактивного поиска неточных повторов, позволяющая учитывать заданную экспертом семантику повторов. Создан алгоритм поиска по образцу, доказана полнота данного алгоритма.
3. Создан метод улучшения документации на основе неточных повторов, включая автоматизированный рефакторинг документации в формате DocBook.

Научная новизна представленных результатов заключается в следующем.

1. Предложенная формальная модель неточных повторов является новой: в ней делается акцент на синтаксической структуре повторов, уточняется и формализуется концепция повторно используемого контента, предложенная P. Bassett. M. Nosál и J. Porubän дают определение неточного повтора, но оно является неформальным и оставляет в стороне синтаксическую структуру повтора. В работах E. Juergens с коллегами и M. Oumaziz с коллегами лишь говорится о необходимости поиска неточных повторов, но не предлагается соответствующих формальных моделей.
2. Предложенные алгоритмы поиска неточных повторов в документации ПО новы, основываются на формальной модели повторов, что позволяет формально доказывать их свойства. В свою очередь, существующие алгоритмы поиска неточных клонов ПО не могут быть применены для решения данной задачи, так как они производят поиск по синтаксическому дереву программы, а тексты на естественных языках не могут быть эффективно проанализированы таким способом. Алгоритмы из области информационного поиска, решая аналогичную задачу, созданы для другой модели использования, решая такие вопросы как ранжирование результатов при представлении больших выдач, решение проблемы производительности на больших коллекциях документов, определение сходства/различия документов целиком (а не поиск повторяющихся фрагментов) и т.д.

3. Предложенный метод улучшения документации на основе неточных повторов является новым. Выдвинута идея использовать повторы для унификации документации и не считать их плагиатом или ненужной избыточностью. Существующие инженерные подходы к использованию повторов ограничиваются повторным использованием (M. Nosál и J. Porubän, M. Oumaziz) и верхнеуровневым анализом качества документации (A. Wingkvist).

Теоретическая и практическая значимость работы. Полученные результаты обобщают исследования в области поиска повторов в документации ПО, впервые предлагая формальные определения и алгоритмы. При этом использовались современные методы поиска, такие как поиск клонов в ПО.

Практическая значимость работы заключается в создании метода улучшения документации на основе неточных повторов, а также в реализации предложенных алгоритмов в рамках программного инструмента Duplicate Finder, исходные коды которого, а также примеры использования, выложены в свободный доступ: https://github.com/spbu-se/pldoctoolkit/blob/master/doc-clone-miner.

Достоверность результатов работы подтверждается формальными доказательствами базовых свойств предложенных алгоритмов, а также инженерными экспериментами на документах реальных программных продуктов.

Результаты исследования были доложены на следующих научных конференциях и семинарах: 10th International Andrei Ershov Memorial Conference on Perspectives of System Informatics (PSI, 26 августа 2015 года, Казань), Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE, 31 марта 2017 года, Иннополис), 5th International Conference on Actual Problems of System and Software Engineering (APSSE, 15 ноября 2017 года, Москва), семинар в ИПМ им. М.В. Келдыша РАН (05 декабря 2017 года, Москва), семинар в СПбГТУ (08 декабря 2017 года, Санкт-Петербург).

Дополнительной апробацией результатов является поддержка исследований, представленных в диссертации, грантом РФФИ №16-01-00304 «Управление повторами при разработке и сопровождении документации программного обеспечения».

Публикации по теме диссертации. Все результаты диссертации опубликованы в 8-ми печатных работах, из них 5 зарегистрированы в РИНЦ, 3 статьи изданы в журналах из «Перечня российских рецензируемых научных журналов, в которых должны быть

опубликованы основные научные результаты диссертаций на соискание учёных степеней доктора и кандидата наук», 3 статьи опубликованы в изданиях, входящих в базы цитирования Scopus и Web of Science.

Работы [1–6, 8] написаны в соавторстве. Личный вклад автора в данных публикациях заключается в следующем. В работе [1] автор разработал алгоритма поиска неточных повторов, соавторы реализовали алгоритм и выполнили необходимые эксперименты. В работах [2, 3] автор разработал и реализовал алгоритм поиска неточных повторов на основе поиска клонов, соавторы участвовали в создании идеи алгоритма, разработке процесса и настройки инструмента поиска клонов. В работах [4, 5] автору принадлежит идея доработки алгоритма поиска неточных повторов на основе клонов, реализация этих доработок. Соавторы занимались экспериментами. В работе [6] автор разработал методику применения поиска неточных повторов для улучшения документации, соавторы участвовали в разработке идеи методики, сделали обзор литературы и выполнили эксперименты. В работе [8] автору принадлежат идеи визуальных метафор, предложенные для отображения результатов иерархического сравнения документов, соавторы предложили концепцию создания целевых сервисов на базе поиска повторов для разработки и сопровождения офисной документации.

Объем и структура работы. Диссертация состоит из введения, пяти глав, заключения и приложения. Полный объем диссертации — 122 страницы текста с 16 рисунками и 5 таблицами. Список литературы содержит 153 наименования.

СОДЕРЖАНИЕ РАБОТЫ

Во **введении** описывается проблематика и делается постановка задачи диссертационного исследования

В **главе 1** представлен обзор исследований в данной области и делаются выводы о необходимости проведения данного исследования.

В **главе 2** представлена формальная модель неточных повторов. Ниже приводятся основные определения из этой модели.

Определение 1. Определим двухместный предикат Before на множестве $D^* \times D^*$, (D^*— множество всех текстовых фрагментов документа D), который является истинным для двух текстовых фрагментов $g^1, g^2 \in D$ тогда и только тогда, когда $e^1 < b^2$, где $[g^1] = [b^1, e^1], [g^2] = [b^2, e^2]$, а b^1, e^1 и b^2, e^2 являются началами и концами от-

резков, содержащих текстовые фрагменты g^1 и g^2 (отрезки являются частью числовой оси от 1 до длины документа в символах). При этом операция [ ] выдает отрезок по текстовому фрагменту.

Определение 2. Рассмотрим упорядоченный набор точных групп $\langle G_1, \dots, G_N \rangle$ некоторого документа D (точная группа — это упорядоченное множество точных повторов, то есть вхождений в документ идентичного фрагмента текста), и каждая группа имеет одинаковое количество элементов: $\#G_1 = \cdots = \#G_N$. Также пусть текстовые фрагменты, имеющие в разных точных группах одни и те же порядковые номера, следуют в исходном тексте в одном и том же порядке, то есть $\forall g_i^k \in G_i \ \forall g_j^k \in G_j \ ((i < j) \Leftrightarrow \text{Before}(g_i^k, g_j^k))$. Кроме того, пусть $\forall\, k \in \{1, \dots, N-1\}$ будет истинно: $\text{Before}(g_N^k, g_1^{k+1})$. Наконец, пусть $\forall k \in \{1, \dots, \#G_1\}$ справедливо следующее:

$$\sum_{i=1}^{N-1} \text{dist}\left(g_i^k, g_{i+1}^k\right) \leq 0{,}15 * \sum_{i=1}^{N} |\, g_i^k|.$$

Тогда будем говорить, что набор точных групп $\langle G_1, \dots, G_N \rangle$ порождает ***группу неточных повторов***.

Константа 0,15 в этом определении взята из работы (Bassett, 1996). Данное определение используется в алгоритме компоновки неточных повторов (листинг 1). Следует отметить, что определение 2 фиксирует константу сходства (хотя наши эксперименты показали, что она слишком жёсткая и что на практике её удобно варьировать), а также не допускает вариаций по краям неточного повтора и в явном виде ориентируется на компоновку неточных повторов из точных. Ниже представлено обобщённое определение неточного повтора, свободное от этих ограничений.

Определение 3. Пусть у нас имеется набор текстовых фрагментов $fr_1, \dots, fr_M$ документа D. Пусть также истинно: $\forall\, i$ Before (fr_i, fr_{i+1}), и существует упорядоченный набор строк $I_1, \dots, I_N$ такой, что имеется вхождение этого набора в каждый текстовый фрагмент fr_j, то есть $\forall j \in \{1, \dots, M\}\, \forall i \in \{1, \dots, N\}(\, I_i \subset \text{str}(fr_j)) \ \wedge\ \forall\, i'\, i'' \in \{1, \dots, N\} \left(i' < i'' \Rightarrow \text{Before}\left(I_{i'}^j, I_{i''}^j\right)\right)$, где I_i^j — вхождение I_i в fr_j. Пусть также для некоторого k: $0 < k \leq 1$ выполнено следующее:

$$\forall j \in \{1, \dots, M\}: \ \frac{\sum_{i=1}^{N} |I_i|}{|fr_j|} \geq k.$$

Тогда будем говорить, что этот набор текстовых фрагментов $fr_1, \dots, fr_M$ является ***группой неточных повторов с мерой близости k.***

В представленном исследовании используется более узкий диапазон значений k, что подробно обсуждается в тексте диссертационной работы.

Утверждение 1. Группа неточных повторов в смысле определения 2 является также группой неточных повторов в смысле определения 3. Обратное не верно, даже если в определении 2 варьировать константу 0,15 так же, как в определении 3.

Алгоритм компоновки неточных повторов, реализованный на основе определения 2, представлен на листинге 1. Массовой операцией алгоритма является поиск для некоторых интервала и множества интервалов такого подмножества, в котором каждый интервал пересекается с данным. Для этой цели используется интервальное дерево; для неточного повтора $g = (g_1^k, \dots, g_N^k)$, составленного из точных $g_1^k, \dots, g_N^k$, соответствующий интервал выглядит так: $[b_1^k - x^k, e_N^k + x^k]$, где $x^k = 0{,}15 * \sum_{i=1}^{N} |g_i^k| - \sum_{i=1}^{N-1} \mathrm{dist}\,(g_i^k, g_{i+1}^k)$.

```
/* SetG — Входные данные
/* SetVG — Результат
SetVG ← ∅
Initiate()
repeat
  SetNew ← ∅
  for each G ∈ SetG ∪ SetVG
    SetCand ← NearBy(G)
    if SetCand ≠ ∅
      G' ← GetClosest(G, SetCand)
      Remove(G, G') /* Удаление G и G' из SetG и SetVG
      if Before(G, G')
        SetNew ← SetNew ∪ {⟨G, G'⟩}
      else
        SetNew ← SetNew ∪ {⟨G', G⟩}
      Join(SetVG, SetNew)
until SetNew = ∅
SetVG ← SetVG ∪ SetG
```

Листинг 1. Алгоритм компоновки неточных повторов

```
/* D, p, k — Входные данные
/* R — Результат
W1 ← ∅  /* начало фазы 1 (сканирование)
for ∀ w1: w1 ∈ D ∧ |w1| = Lw
    if d_di(w1, p) ≤ k_di
        add w1 to W1
W2 ← ∅    /* начало фазы 2 («усушка»)
for w ∈ W1
  w2' ← w
  for l ∈ I
    for ∀w2: w2 ⊆ w ∧ |w2| = l
      if Compare(w2, w2', p)
        w2' ← w2
      add w2' to W2
W3 ← Unique(W2)  /* начало фазы 3 (фильтрация)
for w3 ∈ W3
  if ∃w3' ∈ W3: w ⊂ w3'
    remove w3 from W3
R ← W3
```

Листинг 2. Алгоритм поиска по образцу

Теорема 1. Алгоритм компоновки неточных повторов является корректным относительно определения 2, то есть его результаты соответствуют данному определению.

Доказательство опирается на формальное описание функции NearBy, выдающей для произвольной группы повторов список ближайших к ней. Наилучший кандидат используется для компоновки соответствующей неточной группы. Понятие близости групп повторов также формально определяется в диссертации.

Алгоритм не обладает полнотой, поскольку скомпоновать исходные точные повторы в неточные можно многими альтернативными способами. Кроме того, исходное множество точных повторов также не является полным.

Далее предложен ряд оптимизаций, уменьшающих объем выдачи алгоритма.

В **главе 3** представлена методика интерактивного поиска неточных повторов, реализующая семантический поиск посредством вовлечения пользователя в процесс (то есть семантику задаёт и контролирует пользователь); в методику входит предложенный автором диссертации алгоритм поиска по образцу, сформулирован критерий полноты и доказана полнота предложенного алгоритма.

Определение 4. Пусть для некоторого документа D у нас имеется группа неточных повторов G с точностью k в смысле определения 3, и пусть имеется p — текстовый фрагмент документа D. Если $p \in G$, то будем говорить, что G является ***группой неточных повторов фрагмента $\mathbf{p}$ с точностью $\mathbf{k}$.***

Критерий полноты для нашего алгоритма определим следующим образом. Пусть для произвольного документа D, для любого его текстового фрагмента p, а также для R — некоторой выдачи алгоритма и любой группы неточных повторов G фрагмента p с точностью k выполнено следующее условие:

$$\forall g \in G\ \exists w \in R\colon\ |g \cap w| \geq \frac{|p|}{2}\left(3k - \frac{1}{k}\right).$$

Смысл данного критерия применительно к алгоритму поиска по образцу заключается в том, что если этот критерий выполнен для любой выдачи алгоритма R, то для любого неточного повтора образца p, содержащегося в документе D, множество R будет содержать текстовый фрагмент, который *существенно пересекает* данный повтор. Таким образом, этот неточный повтор оказывается покрыт выдачей, и пользователь, просматривая результаты работы алгоритма, сможет, при желании, выполнить редактирование границ соответствующего элемента выдачи с тем, чтобы данный повтор был включен в результирующую выдачу полностью. На практике наилучшие результаты

достигаются для $k \geq 0{,}77$, т.к. в этом случае критерий полноты дает нижнюю оценку $\frac{|p|}{2}$, то есть все элементы R «цепляют» неточные повторы минимум на половину длины образца. Но следует отметить, что результаты экспериментов показывают бóльшее пересечение выдачи алгоритма и неточных повторов в документе.

Алгоритм поиска по образцу на фазе 1 использует определение редакционного расстояния по наибольшей общей подпоследовательности (монография J. Leskovec с коллегами «Leskovec, J. Mining of Massive Datasets», 2014 год). Спецификация алгоритма представлена на листинге 2.

Теорема 2. Алгоритм поиска по образцу обладает полнотой, то есть любая его выдача (множество R) удовлетворяет критерию полноты.

Доказательство ведётся по основным фазам алгоритма: показывается, что каждая из них не нарушает полноты. Доказательство также опирается на ряд дополнительных свойств неточных повторов, сформулированных и доказанных в тексте диссертации.

Далее приводятся четыре оптимизации алгоритма, предназначенные для увеличения быстродействия и уменьшения объёма выдачи. Для трёх из них доказывается, что они не нарушают полноты.

В **главе 4** представлен метод улучшения документации на основе неточных повторов, включая автоматизированный рефакторинг документации в формате DocBook. Общая схема метода представлена на рис. 1.

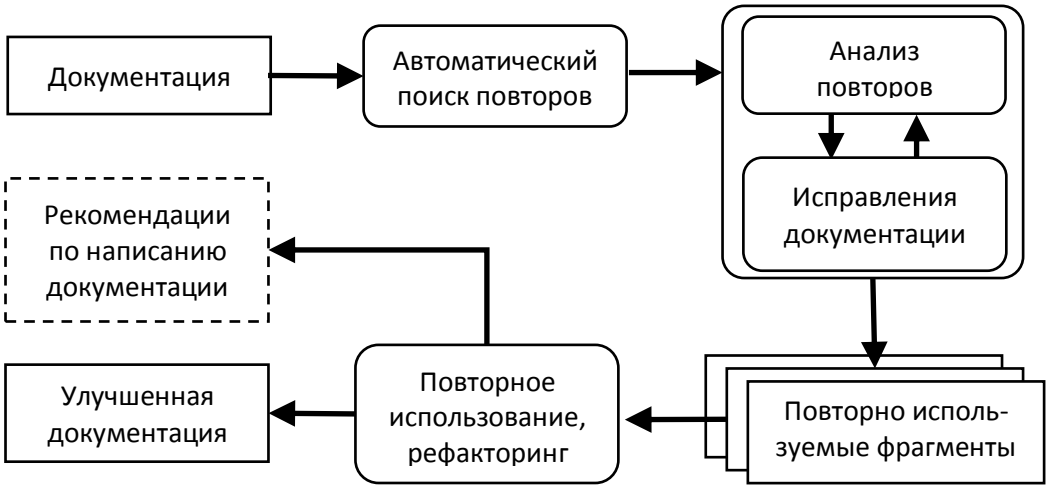

Рис. 1. Схема метода улучшения документации ПО
на основе поиска неточных повторов

Глава 5 посвящена реализации предложенных подходов, а также экспериментам на реальной документации ПО. Представлен программный инструмент Duplicate Finer,

разработанный автором диссертационной работы. Инструмент работает в двух режимах: автоматическом (см. рис. 2) и интерактивном (см. рис. 3). Первый режим позволяет сделать экспресс оценку наличия повторов в документации на основе алгоритма компоновки неточных повторов. Второй режим позволяет учесть семантику повторов через интерактивное взаимодействие с пользователем (то есть пользователь задаёт шаблон поиска, «замыкая» его семантически); данный режим использует алгоритм поиска по образцу. Инструмент работает с документацией в формате DocBook и «плоским» текстом. Практически, любой другой формат конвертируется в DocBook/«плоский» текст с помощью известной утилиты Pandoc.

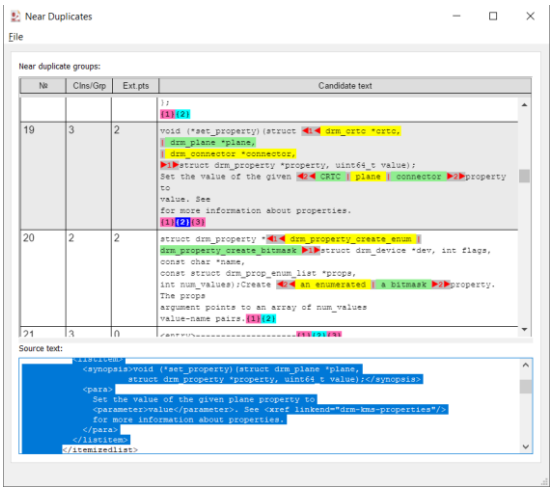

Рис. 2. Пример отображения неточного повтора (автоматический режим)

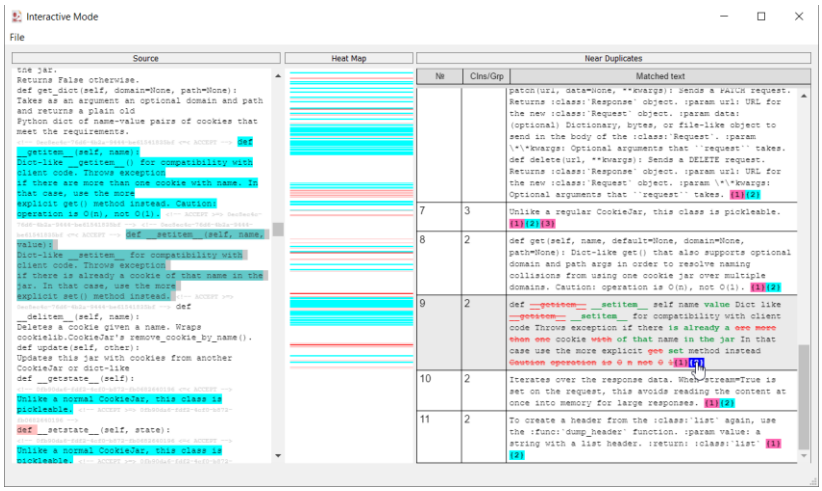

Рис. 3. Пример тепловой карты и окна просмотра найденных повторов (интерактивный режим)

В **заключении** формулируются итоги диссертационного исследования.

1. Предложена формальная модель неточных повторов в программной документации, разработан алгоритм поиска неточных повторов в документации ПО на основе компоновки точных повторов, найденных с помощью метода поиска точных клонов ПО. Доказана корректность алгоритма.
2. Создана методика интерактивного поиска неточных повторов, позволяющая учитывать заданную экспертом семантику повторов. Создан алгоритм поиска по образцу, доказана полнота данного алгоритма. Выполнены оптимизации алгоритма для увеличения быстродействия и уменьшения количества ложноположительных срабатываний.

3. Создан метод улучшения документации ПО на основе неточных повторов, включая автоматизированный рефакторинг документации в формате DocBook.

В качестве **рекомендаций для использования полученных результатов** в промышленности, образовании и научных исследованиях указывается, что предложенные алгоритмы могут быть реализованы в составе более сложных целевых сервисов по разработке промышленной документации ПО. Модульная архитектура реализации предложенных алгоритмов допускает замену отдельных элементов (алгоритма вычисления редакционного расстояния и др.), а также эффективное распараллеливание. Предложенная в работе методика может быть уточнена в соответствии с особенностями процесса разработки документации в конкретной компании и эффективно применена при сопровождении больших пакетов долгоживущей документации, а также для выработки корпоративного стандарта документации. Предложенные средства разработки могут быть применены на практике — как непосредственно, так и в составе более сложных средств разработки и поддержки документации ПО.

Сформулированы также следующие **перспективы дальнейшей разработки представленной в работе тематики**: разработка алгоритмов автоматического поиска неточных повторов, учитывающих структуру документа, автоматизированное извлечение из документации иерархии объектов, которые документация описывает (с использованием машинного обучения); классификация повторов в зависимости от типа документации, дальнейшее совершенствование инструментов поиска неточных повторов.

СПИСОК ПУБЛИКАЦИЙ АВТОРА ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи из «Перечня российских рецензируемых научных журналов, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней доктора и кандидата наук», сформированного согласно требованиям, установленным Министерством образования и науки Российской Федерации

1. Луцив, Д.В. Обнаружение неточно повторяющегося текста в документации программного обеспечения / Л.Д. Кантеев, Ю.О. Костюков, Д.В. Луцив, Д.В. Кознов, М.Н. Смирнов // Труды Института системного программирования РАН. — 2017. — № 4. — С. 303–314.

2. Луцив, Д.В. Задачи поиска нечётких повторов при организации повторного использования документации / Д.В. Луцив, Д.В. Кознов, Х.А. Басит, А.Н. Терехов // Программирование. — 2016. — № 4. — С. 39–49.
3. Луцив, Д.В. Метод поиска повторяющихся фрагментов текста в технической документации / Д.В. Луцив, Д.В. Кознов, Х.А. Басит, О.Е. Ли, М.Н. Смирнов, К.Ю. Романовский // Научно-технический вестник информационных технологий, механики и оптики. — 2014. — 4 (92). — С. 106–114.

Публикации по теме диссертации в Scopus и Web of Science

4. Koznov, D.V. Clone detection in reuse of software technical documentation / D.V. Koznov, D.V. Luciv, H.A. Basit, O.E. Lieh, M.N. Smirnov // Lecture Notes in Computer Science. — 2016. — Vol. 9609. — P. 170–185.
5. Luciv, D.V. On Fuzzy Repetitions Detection in Documentation Reuse / D.V. Luciv, D.V. Koznov, A.N. Terekhov, H.A. Basit // Programming and Computer Software. — 2016. — 4 (42). — P. 216–224.
6. Koznov, D.V. Duplicate management in software documentation maintenance / D.V. Koznov, D.V. Luciv, G.A. Chernishev // Proceedings of the 5th International Conference on Actual Problems of System and Software Engineering (APSSE 2017). CEUR Workshops proceedings. — Vol. 1989. — 2017. — P. 195–201.

Публикации по теме диссертации в других изданиях

7. Luciv, D.V. Detecting and Tracking Near Duplicates in Software Documentation. / D.V. Luciv // Preliminary Proceedings of the 11th Spring/Summer Young Researchers' Colloquium on Software Engineering. — 2017. — P. 125–129.
8. Луцив, Д.В. Иерархический алгоритм DIFF при работе со сложными документами / Д.В. Луцив, Д.В. Кознов, В.С. Андреев // Системное программирование. — 2012. — 7 (1). — С. 57–68.