

Организация структур данных многопользовательского текстового редактора

Р. Е. Спиридонов¹, С. А. Акентьев², К. С. Ильин³, П. С. Чигвинцев⁴

СПбГЭТУ «ЛЭТИ»

¹Rom@nSpiridonov.ru, ²Sergey@Akentev.com,
³kamushek.stone@gmail.com, ⁴pavel4ig@gmail.com

Аннотация. При подготовке систем и тренировочных комплексов для подготовки к работе операторов АСУ ТП, необходима совместная работа над проектом документации комплекса испытаний оператора, позволяющая охватить все возможные сценарии работы системы. Для более удобной разработки документации необходима совместная работа нескольких человек над одним набором документов, что приводит к ряду сложностей, связанных с одновременным редактированием одного и того же текста. В данной статье мы рассматриваем процесс онлайн работы с текстом, а также организацию структуры данных для обеспечения возможности одновременной работы с текстом для нескольких пользователей, сохранением полной истории редактирования, а также обеспечением дополнительного функционала по защите части текста от редактирования.

Ключевые слова: текстовый редактор; структуры данных; многопользовательский редактор; строки

I. ВВЕДЕНИЕ

Внедрение на предприятие со сложными техническими объектами АСУ ТП невозможно без обучения операторов. В последнее время, после законодательного подкрепления, необходима отработка действий операторов на учебных тренажерах, а затем уже работа непосредственно с самой АСУ ТП. Подготовка комплекса документации для тренажера выполняется не только разработчиками системы, но также представителями предприятия, которые дополняют сценарии тестирования операторов. В случае неудовлетворительной подготовки по одному из сценариев, автору этого фрагмента необходимо его доработать. Для этой задачи требуется сохранение авторства у каждого из фрагментов документа.

Для совместной работы над одним документом уже существует ряд инструментов, решающих вопросы одновременной работы коллектива авторов. Однако, в основном, редакторы нацелены на индивидуальную работу с файлами и совместная работа над одним документом в них проблематична.

Совместная работа над текстовым документом приводит к необходимости синхронизации действий пользователей между собой и объединения их в одном документе. На данный момент используется несколько подходов, получивших распространение среди текстовых редакторов:

- Автоматическое слияние изменений от разных пользователей с синхронизацией изменений в тексте по таймеру.
- Блокировка всего документа от редактирования, пока с ним работает один из пользователей.
- Непрерывное отображение онлайн всех действий пользователя: от движений курсора до набора отдельных символов.

Типовые структуры данных, которые используются для хранения текста, обычно оптимизированы по критерию сложности редактирования и вставки в середину текста. В своём большинстве, они обеспечивают независимость операций вставки от объема редактируемого текста.

Добавление мета информации об авторе фрагмента текста, времени его редактирования, а также других свойств, хоть и возможно, но работать при этом с такой информацией для использования её в различных режимах вывода текста накладывает существенные ограничения.

Для начала разработки структуры данных определим требования, которым должно удовлетворять предлагаемое решение.

II. ПОСТАНОВКА ЗАДАЧИ

Для составления требований к структуре данных определим требования к самому текстовому редактору.

Весь текст в редакторе формируется разными пользователями в определённой ими последовательности. При этом необходимо обеспечение работы ряда дополнительных функций:

- Возможность восстановить состояние текста на любой момент времени в прошлом.
- Сохранять информацию об авторе каждого фрагмента.
- Текст между разными фрагментами должен сохранять форматирование — размер шрифта, цвет шрифта, форматирование заголовков и т.п.
- Возможность сохранять черновик текста и обозначать место написания нового фрагмента текста для других пользователей.

- Желательна минимизация сетевого обмена при работе с текстом.

Помимо приведенных функций для текста, необходима возможность вывода документа в трех режимах: просмотр, изменение, история.

В режиме просмотр можно увидеть актуальную опубликованную версию документа на текущий момент времени.

В режиме правка в тексте присутствует текст моих черновиков, видны заблокированные другими авторами фрагменты текста, а также отображаются места редактирования другими авторами и фрагменты текста, заблокированные от редактирования.

III. ПРЕДСТАВЛЕНИЕ ТЕКСТА

Для обеспечения высокой скорости выполнения операций добавления или изменения отдельных фрагментов текста обратим наше внимание на структуру данных для хранения текста верёвки (Ropes), или как их ещё называют, быстрые строки.

Основная особенность верёвок в том, как выполняется хранение отдельных фрагментов текста. Для хранения разделённой строки создается отдельный узел, указывающий, что внутри него хранится конкатенация двух строк с указателями на эти строки.



Рис. 1. Хранение строки из 2 фрагментов

Такой способ объединения подстрок работает за $O(1)$, так как для конкатенации требуется создать новый узел, описывающий операцию. Этот узел также представляется в виде текстового узла, поэтому может участвовать в конкатенации с другими строками, образуя сложные древесные структуры из отдельных фрагментов и их объединений.

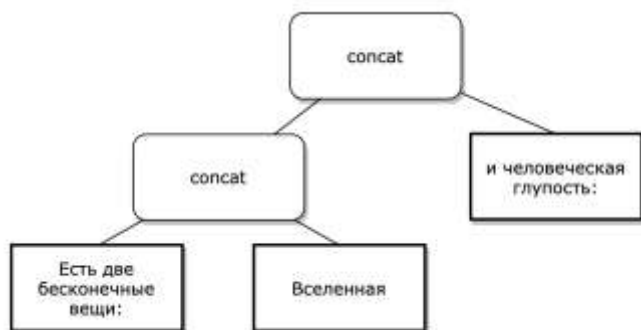


Рис. 2. Хранение строки из 3 фрагментов

Аналогичный подход позволяет решить вопрос разбиения строки на две подстроки. В этом случае, на

место исходной строки встаёт узел конкатенации, а подстроки уходят в его дочерние узлы. Но при этом также остаётся проблема с балансировкой строк, в случае частых малых редактирований.

Сами верёвки представляют структуру, которая позволяет при необходимости расширить отдельный фрагмент текста для хранения дополнительных полей со ссылкой на автора и время редактирования. Но при добавлении нового узла или изменении существующих, потребуется получить всё дерево из базы данных и выполнить необходимые операции. Затем произвести балансировку дерева, а затем положить обновлённые узлы обратно. Такой подход не оптимален с точки зрения используемых ресурсов сервера.

Отдельно стоит отметить необходимость преобразования всего дерева в другую структуру данных для вывода текста в режиме История.

IV. ЛИНЕЙНАЯ ТЕКСТОВАЯ ВЕРЕВКА

За основу идеи хранения текста в нашем редакторе возьмем подход с разбиением текста на фрагменты, как это выполнялось в структуре данных верёвки. Но, так как SQL базы данных больше предназначены для сохранения линейных списков, а не древесных структур, и обеспечение адресации текста на линейной структуре позволяет точно определять все входящие внутрь участка текста фрагменты, то для решения наших задач модифицируем структуру данных верёвки.

Для перехода от дерева к линейному списку, откажемся от промежуточных узлов, говорящих об операциях конкатенации дочерних узлов и добавим в текстовый фрагмент ссылку на предыдущий текстовый фрагмент. У первого фрагмента при этом будет указатель на NULL, что определит начальный фрагмент в документе.

Для упрощения работы с базой данных и ухода от серии рекурсивных запросов или необходимости написания встраиваемых функций, добавим в каждый фрагмент текста ссылку на документ, которому принадлежит этот фрагмент.

Для сохранения в линейном списке всех фрагментов, в том числе, утративших свою актуальность для текущей редакции, но требуемые для вывода в режиме История, в каждый текстовый фрагмент добавим поле для сохранения его статуса. Определим следующие возможные статусы для фрагмента текста:

1. Черновик

Фрагмент текста со статусом черновик будет виден только автору этого фрагмента. Остальные редакторы документа увидят только сам факт редактирования текста автором черновика в его позиции.

2. Опубликован

Все авторы, участвующие в редактировании документа, увидят текст из этого фрагмента. В режиме Просмотр выводятся только фрагменты текста со статусом Опубликован.

3. Заблокирован для редактирования

Данный статус используется для выделения части опубликованного текста, которая занята одним из авторов и связана с новым черновиком. Связь с черновиком выполняется с помощью указания во всех заблокированных фрагментах идентификатора фрагмента-черновика, что является реализацией связи многие-к-одному.

До момента публикации черновика в режиме Просмотр другие авторы видят заблокированный для редактирования текст точно таким же, как и текст в статусе Опубликован. Однако, в режиме Правка авторы видят, что фрагменты текста с этим статусом сейчас редактируются другим автором и не могут быть взяты другими на редактирование.

После публикации связанного черновика фрагменты со статусом Заблокирован для редактирования переходят в статус Неактуальных.

4. Заблокирован от редактирования

Данный статус позволяет защищать часть документа от редактирования другими авторами. Фрагмент с этим статусом не может быть разбит или объединён с другими фрагментами.

5. Неактуальный

Фрагменты с данным статусом в структуре данных используются для вывода документа в режиме История.

6. Отменённый черновик

В случае, когда автор решил не публиковать свой черновик, а отказаться от него, внутри структуры данных остаётся зафиксированным факт намерения изменения. Само содержимое черновика при этом не сохраняется.

Для сохранения авторства в каждом фрагменте будет указатель на идентификатор автора, а также метки времени создания, публикации и снятия с публикации фрагмента, позволяющие в дальнейшем организовать вывод текста в режиме История.



Рис. 3. Представление цитаты в линейной текстовой верёвке

Пример строки с цитатой Эйнштейна приведён на Рис. 3.

V. ОПЕРАЦИИ НА ЛИНЕЙНОЙ ТЕКСТОВОЙ ВЕРЁВКЕ

При создании нового документа для него автоматически создаётся пустой текстовый фрагмент, от которого дальше происходит формирование всей структуры данных. Все операции при работе с текстом сводятся к общему методу добавления нового фрагмента, который имеет несколько частных случаев.

A. Добавление нового фрагмента текста в конец документа

Добавляем новый фрагмент со статусом Черновик в конец линейной верёвки, связывая его с последним фрагментом. После редактирования нового фрагмента и смены его статуса на Опубликован, его содержимое становится видно всем редакторам документа.

B. Добавление текста в середину

Для добавления текста внутри документа следует знать координаты вставки нового фрагмента: идентификатор фрагмента и смещение от начала текста внутри фрагмента. В случае, когда смещение от начала текста фрагмента равно 0 или полной длине фрагмента, то этот случай становится частным и добавление сводится к вставке нового фрагмента в линейную верёвку, подобно добавлению нового узла в однонаправленный список.

Если смещение от начала фрагмента находится не на границах фрагмента, то необходимо исходный фрагмент текста разделить на два, скопировав начало и окончание текста исходного фрагмента в новые фрагменты. Атрибуты авторства и времени публикации разделяемых фрагментов сохраняют значения исходного фрагмента, в том числе, их статусы остаются в значении Опубликован, а оригинальный фрагмент переходит в статус Неактуального.

Новый черновик при этом помещается между двумя разделёнными фрагментами, первый из которых указывает на исходный фрагмент.

C. Редактирование текста

Выделив участок текста в документе, мы получаем координаты выделенной области. И начало, и конец представляем в виде двух координат: идентификатор фрагмента и смещение внутри текста от начала фрагмента.

Фрагменты, содержащие начало и конец выделенной области, делятся на два. Новый черновик при этом вставляется перед второй частью фрагмента окончания выделения и в него добавляется весь текст, содержавшийся в выделенной для редактирования области.

При такой операции конец начального фрагмента и начало конечного, а также все промежуточные фрагменты, попавшие между ними, получают статус Заблокирован для редактирования, а их указатели на черновик указывают на новый фрагмент с черновиком.

D. Удаление текста

Удаление текста происходит аналогично его редактированию с той лишь разницей, что после выделенного участка текста создается не новый черновик, а новый фрагмент с пустым текстом и статусом Неактуальный, сохраняющий данные об авторе изменений. Сами исходные фрагменты также помечаются статусом Неактуален и указывают с помощью указателя на черновик на новый созданный фрагмент.

Помимо приведённых выше операций, в редакторе также возможно взятие в редактирование всего фрагмента текста целиком, что, по сути, повторяет действие редактирования текста с автоматической подстановки координат точки начала и конца, равных координатам начала и конца фрагмента целиком. А также операции вырезания, копирования, вставки, которые реализуются на описанных выше методах, но добавляют в них функции по работе с буфером обмена операционной системы.

VI. КЛИЕНТ СЕРВЕРНЫЙ ОБМЕН ДАННЫМИ

В целях обеспечения целостности структуры данных для всех пользователей все операции по добавлению новых или разбиению текущих фрагментов выполняются на стороне сервера с обменом сообщениями с помощью Ajax или WebSocket с клиентом, в зависимости от реализации серверной части.

Для вывода документа целиком следует получить от сервера всю структуру данных линейной текстовой верёвки. Но при редактировании фрагментов, так как всё сводится к операциям над линейным списком, можно сократить обмен только до пересылки на сервер нового текста и необходимых координат изменения. А ответ от сервера выполнять только с содержимым изменённых блоков, после чего вносить изменения в локальную структуру в соответствии с пришедшим ответом.

В ходе начального получения текста в редактор от сервера происходит получение массива объектов со всеми фрагментами. После получения данных, на клиентской части формируются две вспомогательных структуры данных: последовательность вывода фрагментов в виде массива идентификаторов фрагментов, отражающих связь всех фрагментов друг с другом, а также объект с идентификаторами фрагментов в качестве ключей атрибутов — и объектами фрагментов текста в качестве их значений.

VII. ЗАКЛЮЧЕНИЕ

Совместная работа над документами позволит быстрее проводить подготовку комплекса документации, а внедрение подобного редактора в процессы подготовки и

пуско-наладки тренажеров для операторов АСУ ТП позволит сократить сроки разработки комплекса документации.

Как было показано в статье, задачи, поставленные перед текстовым редактором, решаются на уровне организации структуры данных, и позволило избежать написания дополнительных алгоритмов для редактора.

Структура линейных текстовых верёвок позволяет извлекать актуальную версию документа не только на текущий момент, но и на произвольный момент в прошлом с помощью запроса к базе данных и вывести её пользователю без дополнительных обработок.

Дальнейшие пути развития текстового редактора — в большей оптимизации сетевого обмена. Для режима Просмотр нет необходимости передавать все фрагменты текста, ведь достаточно получить только опубликованные и заблокированные фрагменты. Также как для режима Правка нет необходимости получать Неактуальные фрагменты.

Отдельным вопросом, опущенным в данной статье, является решение задачи сохранения форматирования текста между разными фрагментами. Для обеспечения целостности форматирования документа при разбиении на фрагменты необходим не только алгоритм, раскрывающий вопросы сохранения и дублирования HTML тегов при разбиении или объединении фрагментов, но и также дополнительная работа со стилями вывода блочных и строчных тегов.

СПИСОК ЛИТЕРАТУРЫ

- [1] Crowley C.P. (1998). Data Structures for Text Sequences. University of New Mexico. (<https://www.cs.unm.edu/~crowley/papers/sds.pdf>)
- [2] Boehm Hans-j., Atkinson Russ, Plass Michael. Ropes: an Alternative to Strings. Software—Practice and experience, vol. 25(12), 1315-1330 (December 1995)
- [3] “Html5 webmessaging,” www.w3.org/TR/webmessaging, Jan, 2014.
- [4] Clipboard API and events, Editor’s Draft, 22 August 2019 (<https://w3c.github.io/clipboard-apis/>)
- [5] Электронное руководство по Vue.js (<https://ru.vuejs.org/v2/guide/>)