



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2011149273/08, 01.06.2010

(24) Дата начала отсчета срока действия патента:
01.06.2010

Приоритет(ы):

(30) Конвенционный приоритет:
05.06.2009 US 12/479,189

(43) Дата публикации заявки: 10.06.2013 Бюл. № 16

(45) Опубликовано: 27.06.2015 Бюл. № 18

(56) Список документов, цитированных в отчете о
поиске: US 2009/0125518 A1, 14.05.2009. US
2006/0136472 A1, 22.06.2006. US 2006/0136387
A1, 22.06.2006. US 7299404 B2, 20.11.2007. RU
2357280 C2, 27.05.2009(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 02.12.2011(86) Заявка РСТ:
US 2010/036968 (01.06.2010)(87) Публикация заявки РСТ:
WO 2010/141517 (09.12.2010)

Адрес для переписки:

129090, Москва, ул. Б. Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

КЛАРК Саймон Питер (US),
БОСЕ Мико Арнаб С. (US),
СУНЬ Сюэлэй (US)

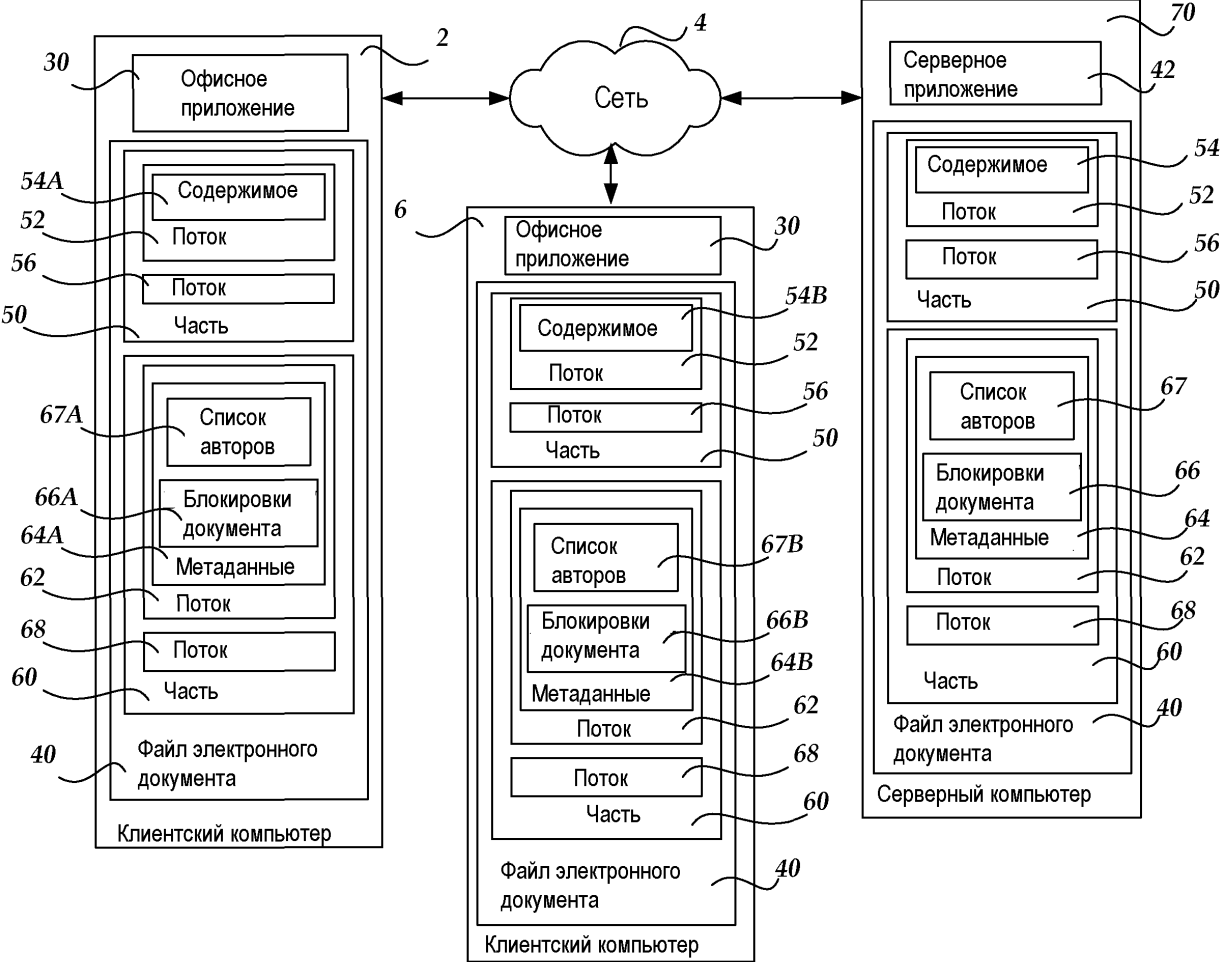
(73) Патентообладатель(и):

МАЙКРОСОФТ ТЕКНОЛОДЖИ
ЛАЙСЕНСИНГ, ЭлЭлСи (US)(54) СИНХРОНИЗАЦИЯ ЧАСТЕЙ ФАЙЛА С ИСПОЛЬЗОВАНИЕМ СЕРВЕРНОЙ МОДЕЛИ
ХРАНЕНИЯ ИНФОРМАЦИИ

(57) Реферат:

Изобретение относится к области синхронизации частей файла с помощью серверной модели хранения информации в клиент-серверной компьютерной сети. Техническим результатом является повышение защищенности данных при синхронизации. Изменения в содержимом электронного документа могут быть приняты на клиентском компьютере. Содержимое может быть включено в первую часть файла, сохраненного на серверном компьютере. Первая часть может включать в себя первый поток, который включает в себя содержимое

электронного документа. Файл может включать в себя множество частей, причем каждая часть включает в себя один или более потоков. Клиентский компьютер может формировать метаданные во второй части файла. Метаданные могут быть ассоциированы с изменениями в содержимом электронного документа в первой части. Вторая часть может затем быть отдельно синхронизирована с серверным компьютером, чтобы сохранять метаданные. Первая часть может затем быть отдельно синхронизирована с серверным компьютером, чтобы сохранять



Фиг. 1

RU 2 5 5 4 8 4 5 C 2

RU 2 5 5 4 8 4 5 C 2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION

(21)(22) Application: 2011149273/08, 01.06.2010

(24) Effective date for property rights:
01.06.2010

Priority:

(30) Convention priority:
05.06.2009 US 12/479,189

(43) Application published: 10.06.2013 Bull. № 16

(45) Date of publication: 27.06.2015 Bull. № 18

(85) Commencement of national phase: 02.12.2011

(86) PCT application:
US 2010/036968 (01.06.2010)

(87) PCT publication:
WO 2010/141517 (09.12.2010)

Mail address:

129090, Moskva, ul. B. Spasskaja, 25, stroenie 3,
OOO "Juridicheskaja firma Gorodisskij i Partnery"

(72) Inventor(s):

**KLARK Sajmon Piter (US),
BOSE Miko Arnab S. (US),
SUN' Sjuehlejh (US)**

(73) Proprietor(s):

**MAJKROSOFT TEKNOLODZhI
LAJSENSING, EhIEhIhSi (US)**

(54) SYNCHRONISING PARTS OF FILE USING SERVER INFORMATION STORAGE MODEL

(57) Abstract:

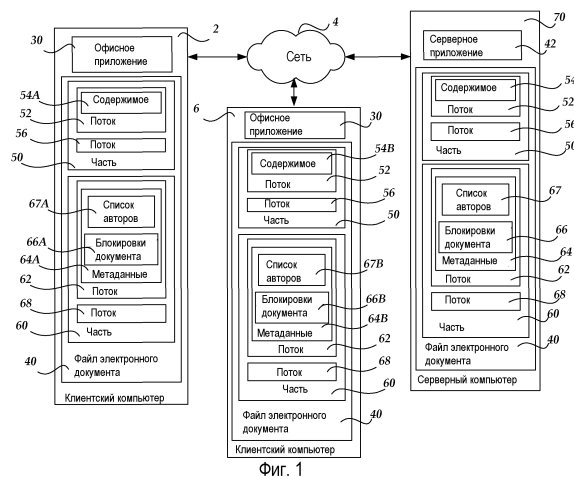
FIELD: physics, computer engineering.

SUBSTANCE: invention relates to synchronisation of parts of a file using a server information storage model in a client-server computer network. Changes to content of an electronic document may be applied on a client computer. The content may be included in a first part of a file stored on a server computer. The first part may include a first stream which includes content of the electronic document. The file may include a plurality of parts, each part including one or more streams. The client computer may generate metadata in a second part of the file. The metadata may be associated with changes in the content of the electronic document in the first part. The second part may then be separately synchronised with the server computer in order to store the metadata. The first part may then be separately synchronised with the server computer in order to save the changes made to the content of the

electronic document.

EFFECT: improved data security during synchronisation.

14 cl, 7 dwg



УРОВЕНЬ ТЕХНИКИ

В клиент-серверных компьютерных сетях файл электронного документа может редактироваться множеством пользователей на различных клиентских компьютерах. Файл электронного документа может включать в себя ассоциированные данные сеанса редактирования (например, метаданные), которые являются различными и не зависят от основного содержимого файла документа. Файл электронного документа синхронизируется между клиентским компьютером и центральным сервером для того, чтобы сохранять любые изменения, сделанные в файле электронного документа, которые могут затем быть просмотрены пользователями других клиентских компьютеров в сети. В существующих способах синхронизации любые ассоциированные метаданные должны быть синхронизированы вместе с изменениями, сделанными в файле электронного документа. В частности, для составных документов на языке разметки или по технологии связывания и внедрения объектов ("OLE") метаданные могут быть расположены в расширяемом формате файла для синхронизации. В результате синхронизации метаданные, ассоциированные с файлом электронного документа, автоматически раскрываются на сервере, даже если пользователь, синхронизирующий файл электронного документа, не желает, чтобы метаданные были просмотрены или загружены (в файле электронного документа) другими пользователями. Принимая во внимание эти факторы и другие причины, были осуществлены различные варианты осуществления настоящего изобретения.

РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

Эта раскрытие предоставлено для того, чтобы представить выбор концепций в упрощенной форме, которые дополнительно описаны ниже в «Осуществлении изобретения». Это раскрытие не предназначено ни для того, чтобы идентифицировать ключевые признаки или существенные признаки заявленного объекта изобретения, ни для того, чтобы использоваться как помощь в определении рамок заявленного объекта изобретения.

Варианты осуществления предоставляются для синхронизации частей файла с помощью серверной модели хранения информации в клиент-серверной компьютерной сети. Изменения в содержимом электронного документа могут быть приняты на клиентском компьютере. Содержимое может быть включено в первую часть файла, сохраненного на серверном компьютере. Первая часть может включать в себя первый поток, который включает в себя содержимое электронного документа. Файл может включать в себя множество частей, причем каждая часть включает в себя один или более потоков. Клиентский компьютер может формировать метаданные во второй части файла. Метаданные могут быть ассоциированы с изменениями в содержимом электронного документа в первой части. Вторая часть может затем быть отдельно синхронизирована с серверным компьютером, чтобы сохранять метаданные. Первая часть может затем быть отдельно синхронизирована с серверным компьютером, чтобы сохранять изменения, сделанные в содержимом электронного документа.

Эти и другие признаки и преимущества станут понятны из прочтения последующего подробного описания и просмотра связанных с ним чертежей. Следует понимать, что как вышеуказанное общее описание, так и последующее детальное описание являются только иллюстративными, а не ограничивающими изобретение согласно формуле изобретения.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Фиг.1 - это блок-схема, иллюстрирующая клиент-серверную сетевую архитектуру, которая может использовать серверную модель хранения информации для

синхронизации частей файла, в соответствии с различными вариантами осуществления;

Фиг.2 - это блок-схема, иллюстрирующая клиентскую вычислительную среду, которая может использовать серверную модель хранения информации для синхронизации частей файла, в соответствии с различными вариантами осуществления;

5 Фиг.3 - это блок-схема, иллюстрирующая алгоритм синхронизации частей файла с помощью серверной модели хранения информации, в соответствии с вариантом осуществления;

Фиг.4 - это блок-схема, иллюстрирующая алгоритм синхронизации одной части файла с помощью серверной модели хранения информации, в соответствии с вариантом осуществления, иллюстрированным на Фиг.3;

Фиг.5 - это блок-схема, иллюстрирующая алгоритм синхронизации другой части файла с помощью серверной модели хранения информации, в соответствии с вариантом осуществления, иллюстрированным на Фиг.3;

15 Фиг.6 - это блок-схема, иллюстрирующая алгоритм синхронизации частей файла с помощью серверной модели хранения информации, в соответствии с другим вариантом осуществления; и

Фиг.7 - это блок-схема, иллюстрирующая алгоритм синхронизации части файла на втором клиентском компьютере с помощью серверной модели хранения информации, в соответствии с вариантом осуществления, иллюстрированным на Фиг.6.

20 ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

Варианты осуществления предоставляются для синхронизации частей файла с помощью серверной модели хранения информации в клиент-серверной компьютерной сети. Изменения в содержимом электронного документа могут быть приняты на клиентском компьютере. Содержимое может быть включено в первую часть файла, 25 сохраненного на серверном компьютере. Первая часть может включать в себя первый поток, который включает в себя содержимое электронного документа. Файл может включать в себя множество частей, причем каждая часть включает в себя один или более потоков. Клиентский компьютер может формировать метаданные во второй части файла. Метаданные могут быть ассоциированы с изменениями в содержимом 30 электронного документа в первой части. Вторая часть может затем быть отдельно синхронизирована с серверным компьютером, чтобы сохранять метаданные. Первая часть может затем быть отдельно синхронизирована с серверным компьютером, чтобы сохранять изменения, сделанные в содержимом электронного документа.

В последующем подробном описании сделаны ссылки на сопровождающие чертежи, 35 которые формируют часть данного документа и на которых показаны посредством иллюстраций конкретные варианты осуществления или примеры. Эти варианты осуществления могут комбинироваться, могут быть использованы другие варианты осуществления и могут быть сделаны структурные изменения без отступления от сущности и объема настоящего изобретения. Последующее детальное описание поэтому 40 дано не в ограничивающем смысле, и объем настоящего изобретения определен приложенной формулой изобретения и ее эквивалентами.

Ссылаясь теперь на чертежи, на которых одинаковые номера ссылок представляют собой одинаковые элементы для нескольких чертежей, описаны различные аспекты настоящего изобретения. Фиг.1 - это блок-схема, иллюстрирующая клиент-серверную 45 сетевую архитектуру, которая может использовать серверную модель хранения информации для синхронизации частей файла, в соответствии с различными вариантами осуществления. Сетевая архитектура включает в себя клиентский компьютер 2 и клиентский компьютер 6, каждый из которых выполнен с возможностью связи с

серверным компьютером 70 через сеть 4. Сеть 4 может включать в себя локальную сеть или глобальную вычислительную сеть (например, Интернет).

Клиентский компьютер 2 может хранить офисное приложение 30, которое может быть использовано для редактирования файла 40 электронного документа (также сохраненного на клиентском компьютере 2). В соответствии с различными вариантами осуществления офисное приложение 30 может содержать программное обеспечение обработки текста WORD, программу презентационной графики POWERPOINT и программное обеспечение GROOVE от корпорации MICROSOFT из города Рэдмонд, штат Вашингтон. Следует принимать во внимание, тем не менее, то, что другие офисные/прикладные программы от других производителей могут быть использованы в соответствии с различными вариантами осуществления, описанными в данном документе.

Файл 40 электронного документа может быть принят на клиентском компьютере 2 с помощью операции загрузки по сети 4 с серверного компьютера 70 или сформирован посредством офисного приложения 30 на клиентском компьютере 2. Файл 40 электронного документа может включать в себя часть 50. Часть 50 может включать в себя потоки 52 и 56. Поток 52 может дополнительно включать в себя содержимое (т.е. содержимое электронного документа) 54А. Файл 40 электронного документа может также включать в себя часть 60. Часть 60 может включать в себя потоки 62 и 68. Поток 62 может дополнительно включать в себя метаданные 64А. В соответствии с вариантом осуществления метаданные 64А могут быть ассоциированы с содержимым 54А, сохраненным в части 50. В частности, метаданные 64А могут включать в себя список 67А авторов, описывающий одного или более авторов содержимого 54А, и блокировки 66А документа, описывающие текущего пользователя файла 40 электронного документа (например, пользователя, редактирующего в настоящий момент содержимое 54А).

В соответствии с различными вариантами осуществления "часть" определяется как "файл" в файле (таком, как файл 40 электронного документа) и может содержать один или более потоков. "Поток", который содержится в части, содержит данные, ассоциированные с файлом электронного документа (такие, как содержимое 54А или метаданные 64А). Следует понимать, что каждая из частей в файле может быть синхронизирована независимо друг от друга и с различными скоростями синхронизации (т.е. части в одном файле могут синхронизироваться в различные моменты времени). Кроме того, множественные потоки в одной и той же части связаны друг с другом, в то же время взаимосвязь может не существовать между потоками в различных частях. Например, поток 56 содержит данные, ассоциированные с содержимым электронного документа (и, таким образом, связанные с потоком 52), но может не содержать метаданные, ассоциированные с файлом 40 электронного документа. Кроме того, множественные потоки в данной части синхронизируются одновременно, таким образом, гарантируя согласованность между потоками в одной и той же части.

Клиентский компьютер 6 может хранить офисное приложение 30, которое также может быть использовано для редактирования файла 40 электронного документа (также сохраненного на клиентском компьютере 6). Файл 40 электронного документа может быть принят на клиентском компьютере 6 с помощью операции загрузки по сети 4 с серверного компьютера 70 или сформирован посредством офисного приложения 30 на клиентском компьютере 6. Файл 40 электронного документа может включать в себя часть 50. Часть 50 может включать в себя потоки 52 и 56. Поток 52 может дополнительно включать в себя содержимое (т.е. содержимое электронного документа) 54В. Файл 40 электронного документа может также включать в себя часть 60. Часть 60 может

включать в себя потоки 62 и 68. Поток 62 может дополнительно включать в себя метаданные 64В. Следует понимать, что в соответствии с различными вариантами осуществления "метаданные" представляют содержимое, содержащееся с потоком в части. В соответствии с вариантом осуществления метаданные 64В могут быть

5 ассоциированы с содержимым 54В, сохраненным в части 50. В частности, метаданные 64В могут включать в себя список 67В авторов, описывающий одного или более авторов содержимого 54В, и блокировки 66В документа, описывающие текущего пользователя файла 40 электронного документа (например, пользователя, редактирующего в

настоящий момент содержимое 54В).

10 Серверный компьютер 70 может хранить серверное приложение 42. В соответствии с различными вариантами осуществления серверный компьютер 70 может функционировать в качестве системы хранения документов для электронных файлов, созданных на клиентских компьютерах 2 и 6. Серверное приложение 42 может содержать прикладную программу совместных служб, такую как приложение служб SHAREPOINT

15 SERVER от корпорации MICROSOFT. Как известно специалистам в области техники, технология служб SHAREPOINT позволяет пользователям создавать, сохранять и представлять совместную среду, чтобы совместно использовать информацию. С помощью технологии пользователь или организация может создавать один или более веб-сайтов, чтобы предоставлять и совместно использовать информацию (например,

20 документы на веб-сервере или в веб-папке и т.д.) для других пользователей, ассоциированных с веб-сайтами. Следует понимать, что варианты осуществления, описанные в данном документе, не должны истолковываться как ограниченные технологией служб SHAREPOINT и что другие технологии совместного обслуживания от других разработчиков и/или производителей могут также быть использованы.

25 Серверный компьютер 70 может также хранить файл 40 электронного документа. Файл 40 электронного документа может включать в себя часть 50. Часть 50 может включать в себя потоки 52 и 56. Поток 52 может дополнительно включать в себя содержимое (т.е. содержимое электронного документа) 54. Файл 40 электронного документа может также включать в себя часть 60. Часть 60 может включать в себя

30 потоки 62 и 68. Поток 62 может дополнительно включать в себя метаданные 64. В соответствии с вариантом осуществления метаданные 64 могут быть ассоциированы с содержимым 54, сохраненным в части 50. В частности, метаданные 64 могут включать в себя список 67 авторов, описывающий одного или более авторов содержимого 54, и блокировки 66 документа, описывающие текущего пользователя файла 40 электронного

35 документа (например, пользователя, редактирующего в настоящий момент содержимое 54). Следует понимать, что в соответствии с вариантом осуществления файл 40 электронного документа может быть загружен с серверного компьютера 70 на клиентские компьютеры 2 и 6 для редактирования, таким образом, потенциально создавая различные версии содержимого 54 и метаданных 64 в файле 40 электронного

40 документа, когда изменения вносятся перед обратной синхронизацией с серверным компьютером 70. Таким образом, в настоящем подробном описании версии содержимого 54 и метаданных 64 на клиентских компьютерах 2 и 6 идентифицируются буквами "А" и "В", присоединенными после ссылочных номеров для этих компонентов файла.

В соответствии с различными вариантами осуществления, как будет описано более

45 подробно ниже со ссылкой на Фиг.3-7, серверное приложение 42 и офисное приложение 30, могут быть сконфигурированы, чтобы упрощать проведение синхронизации частей 50 и 60 на клиентских компьютерах 2 и 6 с серверным компьютером 70. В соответствии с вариантом осуществления серверное приложение 42 может быть сконфигурировано,

чтобы использовать протокол (включающий в себя, но не только, протокол, не использующий информацию о состоянии), чтобы проводить инкрементальную синхронизацию документа и добавлять поддержку серверных файлов, чтобы предоставлять множественные независимые или зависимые потоки данных. Таким образом, каждая из частей в электронном файле может быть синхронизирована независимо друг от друга и с различными скоростями синхронизации (т.е. части в одном файле могут быть синхронизированы в различные моменты времени). Например, серверное приложение 42 вместе с офисным приложением 30 могут быть сконфигурированы, чтобы синхронизировать часть 60 (содержащую метаданные 64) на клиентском компьютере 2 с серверным компьютером 70 перед частью 50 (содержащей содержимое 54А). Следует принимать во внимание, что, синхронизируя части с различными скоростями, объем данных, передаваемых серверному компьютеру 70 по сети 4, уменьшается, таким образом, улучшая масштабируемость сервера. Иллюстративный протокол, не использующий информацию о состоянии, который может быть использован в соответствии с различными вариантами осуществления, описывается в родственной патентной заявке США порядковый номер 12/113,975, озаглавленной "Document Synchronization over Stateless Protocols", поданной 2 мая 2008 года, раскрытие которой содержится в данном документе во всей своей полноте посредством ссылки.

Следует понимать, что различные части и потоки, обсуждаемые в отношении клиентских компьютеров 2 и 6 и серверного компьютера 70, являются просто иллюстративными и не ограничиваются числом частей или потоков, которые могут содержаться в файле электронного документа в соответствии с различными вариантами осуществления. Например, в соответствии с вариантом осуществления файл электронного документа может содержать только одну часть и один поток, в то время как в соответствии с другими вариантами осуществления файл электронного документа может содержать множество частей и множество потоков. Следует дополнительно понимать, что в соответствии с различными вариантами осуществления содержимое потоков, содержащихся в файле электронного документа на клиентских компьютерах 2 и 6 и серверном компьютере 70, не ограничивается метаданными, а может также включать в себя общие данные (т.е. данные, которые не ассоциированы с содержимым, сохраненным в части файла электронного документа) равным образом. Следует дополнительно понимать, что варианты осуществления, описанные в данном документе, не должны истолковываться как ограничиваемые вышеупомянутыми приложениями системы программного обеспечения и что другие приложения системы программного обеспечения от других разработчиков и/или производителей также могут быть использованы. Следует дополнительно понимать, что сетевая архитектура на Фиг.1 и ее компоненты включают в себя функциональность связи с другими вычислительными устройствами, устройствами связи и/или другими системами и не предполагают ограничения вариантами осуществления и примерами, описанными в данном документе. Таким образом, например, сетевая архитектура на Фиг.1 может включать в себя дополнительные клиентские компьютеры, выполненные с возможностью связи с серверным компьютером 70, в соответствии с различными вариантами осуществления.

Примерная операционная среда

Ссылаясь теперь на Фиг.2, нижеследующее пояснение предназначено для предоставления краткого общего описания подходящей вычислительной среды, в которой могут быть реализованы различные иллюстративные варианты осуществления. В то время как различные варианты осуществления будут описаны в общем контексте

программных модулей, которые выполняются вместе с программными модулями, которые работают в операционной системе на персональном компьютере, специалисты в данной области техники поймут, что различные варианты осуществления могут также быть реализованы в комбинации с другими типами компьютерных систем и программных модулей.

Как правило, программные модули включают в себя алгоритмы, программы, компоненты, структуры данных и другие типы структур, которые выполняют конкретные задачи или реализуют конкретные абстрактные типы данных. Более того, специалисты в данной области техники должны принимать во внимание, что различные варианты осуществления могут быть реализованы на практике с другими конфигурациями компьютерных систем, включающими в себя "карманные" устройства, многопроцессорные системы, основанную на микропроцессорах или программируемую бытовую электронную аппаратуру, мини-компьютеры, мэйнфреймы и т.п. Различные варианты осуществления также могут быть реализованы на практике в распределенных вычислительных средах, причем задачи выполняются удаленными обрабатывающими устройствами, которые связаны через сеть передачи данных. В распределенной вычислительной среде программные модули могут размещаться и на локальных, и на удаленных устройствах хранения данных.

Фиг.2 показывает клиентский компьютер 2, который может включать в себя настольный компьютер общего назначения, портативный, карманный, планшетный или другой тип компьютера, допускающего выполнение одной или более прикладных программ. Клиентский компьютер 2 включает в себя, по меньшей мере, один центральный процессор 8 ("CPU"), системную память 12, включающую в себя оперативное запоминающее устройство 18 ("RAM") и постоянное запоминающее устройство ("ROM") 20, и системную шину 10, которая связывает память с CPU 8. Базовая система ввода/вывода, содержащая базовые процедуры, которые помогают передавать информацию между элементами в компьютере, поскольку во время начальной загрузки, хранится в ROM 20. Клиентский компьютер 2 дополнительно включает в себя запоминающее устройство 14 большой емкости, хранящее операционную систему 32, офисное приложение 30 и файл 40 электронного документа, который включает в себя часть 50, потоки 52, 56, 62 и 68, содержимое 54А, метаданные 64А, список 67А авторов и блокировки 66А документа, описанные выше относительно Фиг.1.

В соответствии с различными вариантами осуществления операционная система 32 может быть подходящей для управления работой подключенного в сеть персонального компьютера, такой как операционные системы WINDOWS от корпорации MICROSOFT из города Рэдмонд, штат Вашингтон. Устройство 14 хранения большой емкости подключено к CPU 8 посредством контроллера устройства хранения большой емкости (не показан), подключенного к шине 10. Устройство 14 хранения большой емкости и ассоциированные с ним машиночитаемые носители предоставляют энергонезависимое запоминающее устройство для клиентского компьютера 2. Хотя описание машиночитаемых носителей, содержащееся в данном документе, ссылается на устройство хранения большой емкости, такое как жесткий диск или накопитель CD-ROM, специалистам в данной области техники следует понимать, что машиночитаемые носители могут быть любыми доступными носителями хранения информации, к которым можно осуществлять доступ или использовать их посредством клиентского компьютера 2. В качестве примера, но не для ограничения, машиночитаемые носители могут представлять собой компьютерные носители хранения и среду связи.

Компьютерные носители хранения данных включают в себя энергозависимые и энергонезависимые, съемные и стационарные аппаратные носители хранения данных, реализованные любым физическим способом или технологией для хранения информации, такой как машиночитаемые инструкции, структуры данных, программные модули или другие данные. Компьютерные носители хранения информации включают в себя, но не только, RAM, ROM, EPROM, EEPROM, флэш-память или другую технологию твердотельной памяти, CD-ROM, универсальные цифровые диски ("DVD") или другое оптическое запоминающее устройство, магнитные кассеты, магнитную ленту, накопитель на магнитном диске или другие магнитные устройства хранения информации, которые могут быть использованы для хранения желаемой информации и к которым может быть осуществлен доступ посредством клиентского компьютера 2. Среда передачи данных типично осуществляет машиночитаемые инструкции, структуры данных, программные модули или другие данные в модулированном сигнале данных, таком как несущая волна или другой механизм передачи, и включает в себя любую среду доставки информации. Термин "модулированный сигнал данных" означает сигнал, который имеет одну или более своих характеристик, заданных или изменяемых таким образом, чтобы кодировать информацию в сигнале. В качестве примера, а не ограничения, среда связи включает в себя проводную среду, такую как проводная сеть или прямое проводное соединение, и беспроводную среду, такую как акустическая среда, RF, инфракрасное излучение и другая беспроводная среда. Комбинации любого из вышеприведенных элементов также должны быть включены в объем машиночитаемых носителей. Машиночитаемые носители также могут упоминаться как компьютерный программный продукт.

Согласно различным вариантам осуществления клиентский компьютер 2 может работать в сетевой среде с помощью логических соединений с удаленными компьютерами по сети 4, которая может содержать, например, локальную сеть или глобальную вычислительную сеть (например, Интернет). Клиентский компьютер 2 может подключаться к сети 4 через сетевой интерфейсный блок 16, подключенный к шине 10. Следует понимать, что сетевой интерфейсный блок 16 может также использоваться для подключения к другим типам сетей и удаленным вычислительным системам. Клиентский компьютер 2 может также включать в себя контроллер 22 ввода/вывода для приема и обработки входных данных из множества типов входных данных, включающих в себя клавиатуру, мышь, перо, стилус, палец и/или другое средство. Подобным образом, контроллер 22 ввода/вывода может обеспечивать вывод на устройство 82 отображения, принтер или другой тип устройства вывода. Дополнительно, сенсорный экран может служить в качестве механизма ввода и вывода. Следует принимать во внимание, что клиентский компьютер 6 и серверный компьютер 70, показанные на Фиг.1, могут включать в себя множество традиционных компонентов, показанных относительно клиентского компьютера 2 на Фиг.2.

Фиг.3 - это блок-схема, иллюстрирующая алгоритм 300 синхронизации частей файла с помощью серверной модели хранения информации, в соответствии с вариантом осуществления. При прочтении обсуждения представленных в данном документе алгоритмов следует принимать во внимание, что логические операции различных вариантов осуществления настоящего изобретения реализованы (1) как последовательность реализуемых компьютером действий или программных модулей, выполняющихся в вычислительной системе, и/или (2) как взаимосвязанные машинные логические схемы или модули схем в вычислительной системе. Реализация выбирается и зависит от требований к производительности вычислительной системы, реализующей

изобретение. Соответственно, логические операции, проиллюстрированные на Фиг.3-7 и составляющие различные варианты осуществления, описанные в данном документе, упоминаются по-разному как операции, структурные устройства, этапы или модули. Специалистам в данной области техники следует принимать во внимание, что эти

5 операции, структурные устройства, этапы и модули могут быть реализованы в программном обеспечении, аппаратно-программном обеспечении, цифровой логике специального назначения и любой их комбинации без отступления от сущности и объема настоящего изобретения, указанных в формуле изобретения, изложенной в данном документе.

10 Алгоритм 300 начинается на этапе 305, где офисное приложение 30, выполняющееся на клиентском компьютере 2, принимает изменения в содержимом электронного документа (содержащемся в потоке 52) в части 50 файла 40 электронного документа. В соответствии с вариантом осуществления файл 40 электронного документа может быть загружен с серверного компьютера 70, включающего в себя все части и потоки,

15 содержащиеся в нем. Часть 50 может затем быть открыта для того, чтобы редактировать содержимое 54, содержащееся в ней, таким образом, создавая содержимое 54А.

От этапа 305 алгоритм 300 переходит к этапу 310, где офисное приложение 30 формирует метаданные 64А в части 60 файла 40 электронного документа. В частности, офисное приложение 30 может формировать метаданные, чтобы обновлять список 67

20 авторов в части 60 файла 40 электронного документа, чтобы добавлять пользователя клиентского компьютера 2 в качестве автора (таким образом, формируя список 67А авторов). Офисное приложение 30 может также создавать метаданные, чтобы добавлять блокировку 66А документа в часть 60 файла 40 электронного документа, чтобы указывать, что файл 40 электронного документа в настоящий момент редактируется

25 пользователем клиентского компьютера 2.

От этапа 310 алгоритм 300 переходит к этапу 315, где офисное приложение 30 отправляет запрос серверному приложению 42, чтобы синхронизировать часть 60 с серверным компьютером 70 для того, чтобы сохранять изменения, сделанные в метаданных 64 (т.е. метаданные 64А, содержащие список 67А авторов и блокировку

30 66А документа), на серверном компьютере 70. Следует понимать, что часть 60 может быть синхронизирована независимо от части 50, так что только измененные метаданные 64А сохраняются на серверном компьютере 70. Иллюстративный алгоритм синхронизации части 60 с серверным компьютером будет описан более подробно ниже со ссылкой на Фиг.4.

От этапа 315 алгоритм 300 переходит к этапу 320, где офисное приложение 30 отправляет запрос серверному приложению 42, чтобы синхронизировать часть 50 с серверным компьютером 70 для того, чтобы сохранять отредактированное содержимое

35 54А на серверном компьютере 70. Следует понимать, что часть 50 может быть синхронизирована независимо от части 60, так что только отредактированное содержимое 54А сохраняется на серверном компьютере 70. Следует дополнительно понимать, что порядок синхронизации частей 60 и 50 может быть обратным, так что часть 50 синхронизируется независимо с серверным компьютером 70 перед частью 60. Иллюстративный алгоритм синхронизации части 50 с серверным компьютером 70 будет описан более подробно ниже относительно Фиг.5. От этапа 320 алгоритм 300 переходит

40 к этапу 325, где он затем заканчивается. Следует понимать, что этапы 310-320, обсужденные выше, могут также выполняться офисным приложением 30, выполняющимся на клиентском компьютере 6, в соответствии с вариантом осуществления.

Фиг.4 - это блок-схема, иллюстрирующая алгоритм 400 синхронизации одной части файла с помощью серверной модели хранения информации, в соответствии с вариантом осуществления, иллюстрированным на Фиг.3. Алгоритм 400 начинается на этапе 315 на Фиг.3 и переходит к этапу 405, где серверное приложение 42 определяет, существует ли конфликт с синхронизацией части 60 на клиентском компьютере 2 с серверным компьютером 70. В частности и в соответствии с вариантом осуществления, серверное приложение 42 может сравнивать метаданные 64А, сохраненные в части 60 на клиентском компьютере 2, с метаданными 64, сохраненными на серверном компьютере 70, чтобы определять какие-либо различия. Если метаданные 64А и метаданные 64 различны, тогда серверное приложение 42 определяет, что существует конфликт, и алгоритм 400 переходит к этапу 410. Если различия между метаданными 64А и метаданными 64 не существуют, тогда алгоритм 400 возвращается к этапу 320 на Фиг.3.

На этапе 410 серверное приложение 42 определяет, как разрешить конфликт, на основе того, являются ли метаданные 64А в части 60 на клиентском компьютере более новыми, чем метаданные 64, сохраненные на серверном компьютере 70. В частности, запрос от офисного приложения 30, чтобы синхронизировать часть 60, может включать в себя временную метку, указывающую, когда метаданные 64А были сохранены в части 60 на клиентском компьютере 2. Серверное приложение 42 может сравнивать временную метку с временной меткой, указывающей, когда метаданные 64 были сохранены в части 60 на серверном компьютере 70. Если определяется, что метаданные 64А являются более новыми, чем метаданные 64, тогда алгоритм 400 продолжается на этапе 415. Если определяется, что метаданные 64 являются более новыми, чем метаданные 64А, тогда алгоритм 400 ответвляется к этапу 420. Специалистам в данной области техники следует понимать, что в соответствии с другим вариантом осуществления запрос от офисного приложения 30, чтобы синхронизировать часть 60, может включать в себя тег объекта ("ETAG") на языке разметки гипертекста ("НТТР"), чтобы идентифицировать (и разрешать) конфликты.

На этапе 415 серверное приложение 42 автоматически синхронизирует часть 60 на клиентском компьютере 2 с серверным компьютером 70, чтобы разрешать конфликт в пользу более новых метаданных 64А, присутствующих на клиентском компьютере 2. Таким образом, серверное приложение 42 может обновлять метаданные 64 в части 60 на серверном компьютере 70 с помощью метаданных 64А. От этапа 415 алгоритм 400 возвращается к этапу 320 на Фиг.3.

На этапе 420 серверное приложение 42 автоматически инструктирует прекращать синхронизацию части 60 на клиентском компьютере 2 с серверным компьютером 70 для того, чтобы разрешать конфликт в пользу метаданных 64, являющихся более новыми, чем метаданные 64А. Таким образом, серверное приложение 42 может сохранять метаданные 64 в части 60 на серверном компьютере 70. От этапа 420 алгоритм 400 возвращается к этапу 320 на Фиг.3.

Фиг.5 - это блок-схема, иллюстрирующая алгоритм 500 синхронизации другой части файла с помощью серверной модели хранения информации, в соответствии с вариантом осуществления, иллюстрированным на Фиг.3. Алгоритм 500 начинается на этапе 320 на Фиг.3 и переходит к этапу 505, где серверное приложение 42 определяет, существует ли конфликт с синхронизацией части 50 на клиентском компьютере 2 с серверным компьютером 70. В частности и в соответствии с вариантом осуществления, серверное приложение 42 может сравнивать содержимое 54А, сохраненное в части 50 на клиентском компьютере 2, с содержимым 54, сохраненным в части 50 на серверном компьютере 70. Если содержимое 54 содержит изменения, которые не присутствуют в

отредактированном содержимом 54А в части 50 на клиентском компьютере 2, тогда серверный компьютер 70 определяет, что существует конфликт, и алгоритм 500 переходит к этапу 510. Если содержимое 54 не содержит каких-либо изменений вовсе, тогда алгоритм 500 продолжается на этапе 515.

5 На этапе 510 серверное приложение 42 формирует сообщение об ошибке при конфликте для пользователя клиентского компьютера 2. Например, сообщение об ошибке при конфликте, сформированное серверным приложением 42 на клиентском компьютере 2, может включать в себя невидимые изменения в содержимом 54 (сделанные, например, пользователем клиентского компьютера 6), которые были ранее
10 выгружены на сервер 70. В ответ на сообщение об ошибке при конфликте пользователь клиентского компьютера 2 может вручную разрешать конфликт, объединяя отредактированное содержимое 54А с изменениями в содержимом 54 или альтернативно позволяя предыдущим изменениям в содержимом 54 оставаться неизменными на сервере 70. От этапа 510 алгоритм 500 возвращается к этапу 325 на Фиг.3.

15 На этапе 515 серверное приложение 42 синхронизирует отредактированное содержимое 54А в части 50 с сервером 70, чтобы обновлять содержимое 54 на сервере 70 отредактированным содержимым 54А. От этапа 515 алгоритм 500 возвращается к этапу 325 на Фиг.3.

Фиг.6 - это блок-схема, иллюстрирующая алгоритм 600 синхронизации частей файла
20 с помощью серверной модели хранения информации, в соответствии с другим вариантом осуществления. Алгоритм 600 начинается на этапе 605, где офисное приложение 30, выполняющееся на клиентском компьютере 2, принимает изменения в содержимом электронного документа (содержащемся в потоке 52) в части 50 файла 40 электронного документа. В соответствии с вариантом осуществления файл 40 электронного документа
25 может быть загружен с серверного компьютера 70, включающий в себя все части и потоки, содержащиеся в нем. Часть 50 может затем быть открыта для того, чтобы редактировать содержимое 54, содержащееся в ней, таким образом, создавая содержимое 54А.

От этапа 605 алгоритм 600 переходит к этапу 610, где офисное приложение 30
30 формирует метаданные 64А в части 60 файла 40 электронного документа, чтобы идентифицировать редактора содержимого 54А в части 50. В частности, офисное приложение 30 может формировать метаданные, чтобы добавлять имя редактора содержимого 54А в список 67 авторов в части 60 файла 40 электронного документа (таким образом, создавая список 67А авторов).

35 От этапа 610 алгоритм 600 переходит к этапу 615, где офисное приложение 30 отправляет запрос серверному приложению 42, чтобы синхронизировать часть 60 с серверным компьютером 70 для того, чтобы сохранять изменения, сделанные в метаданных 64 (т.е. метаданные 64А, содержащие список 67А авторов), на серверном компьютере 70. Следует понимать, что часть 60 может быть синхронизирована
40 независимо от части 50, так что только измененные метаданные 64А сохраняются на серверном компьютере 70.

От этапа 615 алгоритм 600 переходит к этапу 620, где офисное приложение 30, выполняющееся на клиентском компьютере 6, принимает части 50 и 60 файлов (включающие в себя потоки 42, 56, 62 и 68, содержащиеся в них) от серверного
45 компьютера 70 посредством загрузки электронного файла 40. Следует понимать, что клиентский компьютер 6, принимая часть 60 файла, также принимает метаданные 64А, идентифицирующие редактора содержимого 54А с клиентского компьютера 2, в результате синхронизации части 60 на клиентском компьютере 2 с серверным

компьютером 70 на этапе 615. Таким образом, пользователь клиентского компьютера 6 имеет возможность определять, что пользователь клиентского компьютера 2 также загрузил содержимое 54 с серверного компьютера 70 для редактирования.

От этапа 620 алгоритм 600 переходит к этапу 625, где офисное приложение 30, выполняющееся на клиентском компьютере 6, принимает изменения в содержимом электронного документа (содержащемся в потоке 52) в части 50 файла 40 электронного документа. Часть 50 может затем быть открыта для того, чтобы редактировать содержимое 54, содержащееся в ней, таким образом, создавая содержимое 54В. Следует понимать, что отредактированное содержимое 54В не включает в себя изменения, сделанные на клиентском компьютере 2 (т.е. отредактированное содержимое 54А).

От этапа 625 алгоритм 600 переходит к этапу 630, где офисное приложение 30, выполняющееся на клиентском компьютере 6, отправляет запрос серверному приложению 42, чтобы синхронизировать часть 60 с серверным компьютером 70 для того, чтобы сохранять изменения, сделанные в метаданных 64 (т.е. метаданные 64В, содержащие список 67В авторов), на серверном компьютере 70. Следует понимать, что часть 60 может быть синхронизирована независимо от части 50, так что только измененные метаданные 64В сохраняются на серверном компьютере 70.

От этапа 630 алгоритм 600 переходит к этапу 635, где клиентский компьютер 2 принимает метаданные 64В, идентифицирующие редактора содержимого 54В с клиентского компьютера 6, в результате синхронизации части 60 с серверным компьютером 70 на этапе 630. В частности, офисное приложение 30, выполняющееся на клиентском компьютере 2, может загружать часть 60 с серверного компьютера 70, которая была ранее выгружена с метаданными 64В с клиентского компьютера 6. Таким образом, пользователь клиентского компьютера 2 имеет возможность определять, что пользователь клиентского компьютера 6 также загрузил содержимое 54 с серверного компьютера 70 для редактирования.

От этапа 635 алгоритм 600 переходит к этапу 640, где офисное приложение 30, выполняющееся на клиентском компьютере 2, отправляет запрос серверному приложению 42, чтобы синхронизировать часть 50 с серверным компьютером 70 для того, чтобы сохранять отредактированное содержимое 54А на серверном компьютере 70. Следует понимать, что часть 50 может быть синхронизирована независимо от части 60, так что только отредактированное содержимое 54А сохраняется на серверном компьютере 70. Следует дополнительно понимать, что порядок синхронизации частей 60 и 50 может быть обратным, так что часть 50 синхронизируется независимо с серверным компьютером 70 перед частью 60.

От этапа 640 алгоритм 600 переходит к этапу 645, где офисное приложение 30, выполняющееся на клиентском компьютере 6, отправляет запрос серверному приложению 42, чтобы синхронизировать часть 50 с серверным компьютером 70 для того, чтобы сохранять отредактированное содержимое 54В на серверном компьютере 70. Следует понимать, что часть 50 может быть синхронизирована независимо от части 60, так что только отредактированное содержимое 54В сохраняется на серверном компьютере 70. Следует дополнительно понимать, что порядок синхронизации частей 60 и 50 может быть обратным, так что часть 50 синхронизируется независимо с серверным компьютером 70 перед частью 60. Иллюстративный алгоритм синхронизации части 50 с серверным компьютером 70 будет описан более подробно ниже относительно Фиг.7. От этапа 645 алгоритм 600 переходит к этапу 650, где он затем заканчивается.

Фиг.7 - это блок-схема, иллюстрирующая алгоритм 700 синхронизации части файла на втором клиентском компьютере с помощью серверной модели хранения информации,

в соответствии с вариантом осуществления, иллюстрированным на Фиг.6. Алгоритм 700 начинается на этапе 645 на Фиг.6 и переходит к этапу 605, где офисное приложение 30, выполняющееся на клиентском компьютере 6, проверяет с серверным компьютером 70 какие-либо обновления, сделанные в части 50. В частности, офисное приложение 30 может отправлять запрос серверному приложению 42 относительно того, было ли содержимое 54 в части 60 на серверном компьютере 70 обновлено клиентским компьютером 2. Как обсуждалось выше на этапе 620 на Фиг.6, пользователь клиентского компьютера 6 может быть осведомлен о том, что содержимое 54 в части 50 серверного компьютера 70 редактируется другим пользователем на клиентском компьютере 2 при приеме частей 50 и 60 файла в электронном файле 40, загруженном с серверного компьютера 70.

После этапа 705 алгоритм 700 продолжается на этапе 710, где клиентский компьютер 6 принимает подтверждение от серверного приложения 42, что часть 50 на серверном компьютере 70 была обновлена (т.е. клиентским компьютером 2), и, таким образом, отредактированное содержимое 54А с клиентского компьютера 2 доступно для объединения с отредактированным содержимым 54В в части 50 клиентского компьютера 6. От этапа 710 алгоритм переходит к этапу 715, где клиентский компьютер 6 принимает отредактированное содержимое 54А от серверного компьютера 70.

От этапа 715 алгоритм 700 переходит к этапу 720, где офисное приложение 30, выполняющееся на клиентском компьютере 6, запрашивает серверное приложение 42 на серверном компьютере 70, чтобы обновлять часть 50 отредактированным содержимым 54А (с клиентского компьютера 2) и отредактированным содержимым 54В (с клиентского компьютера 6). От этапа 720 алгоритм 700 возвращается к этапу 650 на Фиг.6.

Хотя изобретение описано в связи с различными иллюстративными вариантами осуществления, специалисты в данной области техники должны понимать, что множество модификаций может быть выполнено в них в рамках объема нижеследующей формулы изобретения. Соответственно, нет намерения ограничения объема изобретения каким-либо образом вышеуказанным описанием, и вместо этого он должен полностью определяться согласно нижеследующей формуле изобретения.

Формула изобретения

1. Способ синхронизации частей файла с помощью серверной модели хранения информации в клиент-серверной компьютерной сети, содержащий этапы, на которых:

принимают, на клиентском компьютере, изменения в содержимом электронного документа, причем содержимое содержит первый поток первой части в файле, при этом файл содержит первую часть и вторую часть, причем первая часть и вторая часть включают в себя, каждая, по меньшей мере, один поток;

формируют, во второй части на клиентском компьютере, метаданные, ассоциированные с изменениями в содержимом электронного документа в первой части;

синхронизируют вторую часть на клиентском компьютере с серверным компьютером, чтобы сохранять метаданные во второй части файла на серверном компьютере, причем этап синхронизации второй части содержит этапы, на которых:

сравнивают метаданные, сформированные во второй части на клиентском компьютере, с метаданными, сохраненными во второй части файла, сохраненного на серверном компьютере;

определяют конфликт, когда метаданные, сформированные во второй части на клиентском компьютере, отличаются от метаданных, сохраненных во второй части

файла на серверном компьютере,

автоматически синхронизируют вторую часть на клиентском компьютере с серверным компьютером, когда метаданные во второй части на клиентском компьютере более новые, чем метаданные, сохраненные во второй части файла, сохраненного на серверном компьютере, чтобы разрешать конфликт; и

автоматически инструктируют прекращать синхронизацию второй части на клиентском компьютере с серверным компьютером, когда метаданные, сохраненные во второй части файла, сохраненного на серверном компьютере, являются более новыми, чем метаданные, сохраненные во второй части на клиентском компьютере, чтобы разрешать конфликт; и

синхронизируют первую часть на клиентском компьютере с серверным компьютером, чтобы сохранять изменения, сделанные в содержимом электронного документа в первой части файла, на серверном компьютере.

2. Способ по п.1, в котором синхронизация первой части на клиентском компьютере с серверным компьютером, чтобы сохранять изменения, сделанные в содержимом электронного документа в первой части файла, на серверном компьютере, содержит этапы, на которых:

сравнивают отредактированное содержимое электронного документа в первой части на клиентском компьютере с содержимым электронного документа, сохраненным в первой части файла, сохраненного на серверном компьютере; и

определяют конфликт, когда содержимое электронного документа, сохраненное в первой части файла, сохраненного на серверном компьютере, содержит изменения, которые не присутствуют в отредактированном содержимом электронного документа в первой части на клиентском компьютере.

3. Способ по п.2, дополнительно содержащий

прекращение синхронизации первой части между клиентским компьютером и серверным компьютером и формирование, на клиентском компьютере, сообщения об ошибке при конфликте.

4. Способ по п.1, в котором формирование, во второй части на клиентском компьютере, метаданных, ассоциированных с изменениями в содержимом электронного документа в первой части, содержит формирование, в потоке второй части, метаданных, содержащих список авторов документа для содержимого электронного документа.

5. Машиночитаемый носитель хранения информации, содержащий выполняемые компьютером инструкции, которые, когда выполняются на компьютере, будут инструктировать компьютер выполнять способ синхронизации частей файла с помощью серверной модели хранения информации, содержащий этапы, на которых:

принимают, на клиентском компьютере, изменения в содержимом электронного документа, причем содержимое содержит первый поток первой части в файле, при этом файл содержит, по меньшей мере, первую часть и вторую часть, причем каждая, по меньшей мере, из первой и второй частей содержит, по меньшей мере, один поток;

формируют, во второй части на клиентском компьютере, метаданные, ассоциированные с изменениями в содержимом электронного документа в первой части;

синхронизируют вторую часть на клиентском компьютере с серверным компьютером, чтобы сохранять метаданные во второй части файла на серверном компьютере, причем этап синхронизации второй части содержит этапы, на которых:

сравнивают метаданные, сформированные во второй части на клиентском компьютере, с метаданными, сохраненными во второй части файла, сохраненного на серверном компьютере;

определяют конфликт, когда метаданные, сформированные во второй части на клиентском компьютере, отличаются от метаданных, сохраненных во второй части файла на серверном компьютере,

автоматически синхронизируют вторую часть на клиентском компьютере с серверным компьютером, когда метаданные во второй части на клиентском компьютере более новые, чем метаданные, сохраненные во второй части файла, сохраненного на серверном компьютере, чтобы разрешать конфликт; и

автоматически инструктируют прекращать синхронизацию второй части на клиентском компьютере с серверным компьютером, когда метаданные, сохраненные во второй части файла, сохраненного на серверном компьютере, являются более новыми, чем метаданные, сохраненные во второй части на клиентском компьютере, чтобы разрешать конфликт; и

синхронизируют первую часть на клиентском компьютере с серверным компьютером, чтобы сохранять изменения, сделанные в содержимом электронного документа в первой части файла, на серверном компьютере (сохраняют изменения, сделанные в содержимом электронного документа на клиентском компьютере).

6. Машиночитаемый носитель хранения информации по п.5, причем этап синхронизации первой части на клиентском компьютере с серверным компьютером, чтобы сохранять изменения, сделанные в содержимом электронного документа в первой части файла, на серверном компьютере, содержит этапы, на которых:

сравнивают измененное содержимое электронного документа в первой части на клиентском компьютере с измененным содержимым электронного документа, сохраненным в первой части файла, сохраненного на серверном компьютере; и

определяют конфликт, когда содержимое электронного документа, сохраненное в первой части файла, сохраненного на серверном компьютере, содержит изменения, которые не присутствуют в измененном содержимом электронного документа в первой части на клиентском компьютере.

7. Машиночитаемый носитель хранения информации по п.5, причем формирование, в потоке второй части на клиентском компьютере, метаданных, ассоциированных с изменениями в содержимом электронного документа в первой части, содержит формирование списка авторов документа для содержимого электронного документа.

8. Способ синхронизации частей файла с помощью серверной модели хранения информации в клиент-серверной компьютерной сети, содержащий этапы, на которых: принимают, на первом клиентском компьютере, изменения в содержимом

электронного документа, причем содержимое содержит первый поток первой части в файле, при этом файл содержит, по меньшей мере, первую часть и вторую часть, причем первая часть и вторая часть включают в себя, каждая, по меньшей мере, один поток;

формируют, в потоке второй части на первом клиентском компьютере, данные, идентифицирующие редактора содержимого электронного документа в первой части на первом клиентском компьютере;

синхронизируют вторую часть на первом клиентском компьютере с серверным компьютером, чтобы сохранять данные на серверном компьютере;

принимают, на втором клиентском компьютере, файл, сохраненный на серверном компьютере, включающий в себя, по меньшей мере, первую часть и вторую часть, причем вторая часть содержит данные, идентифицирующие редактора содержимого электронного документа на первом клиентском компьютере и идентифицирующие, что содержимое электронного документа первой части в настоящий момент загружается на первом клиентском компьютере для редактирования;

принимают, на втором клиентском компьютере, изменения в содержимом электронного документа в первой части, причем изменения, принятые в содержимом электронного документа на втором клиентском компьютере, не включают в себя изменения, принятые в содержимом электронного документа на первом клиентском

5 компьютере; и

синхронизируют первую часть на первом клиентском компьютере с серверным компьютером, чтобы сохранять изменения, сделанные в содержимом электронного документа, на серверном компьютере.

9. Способ по п.8, в котором этап формирования, в потоке второй части на первом клиентском компьютере, данных, идентифицирующих редактора содержимого электронного документа в первой части на первом клиентском компьютере, содержит этап, на котором формируют, в потоке второй части на первом клиентском компьютере, метаданные, идентифицирующие редактора содержимого электронного документа в первой части на первом клиентском компьютере.

10. Способ по п.9, в котором этап синхронизации второй части на первом клиентском компьютере с серверным компьютером, чтобы сохранять данные на серверном компьютере, содержит этап, на котором синхронизируют вторую часть на первом клиентском компьютере с серверным компьютером, чтобы сохранять метаданные на серверном компьютере.

11. Способ по п.10, дополнительно содержащий этап, на котором синхронизируют вторую часть на втором клиентском компьютере с серверным компьютером, чтобы обновлять метаданные, идентифицирующие редактора содержимого электронного документа на первом клиентском компьютере, идентификационной информацией редактора содержимого электронного документа на втором клиентском компьютере.

12. Способ по п.8, дополнительно содержащий этап, на котором синхронизируют первую часть на втором клиентском компьютере с серверным компьютером, чтобы добавлять, в первую часть на серверном компьютере, изменения, сделанные в содержимом электронного документа на втором клиентском компьютере, к изменениям, сделанным в электронном документе на первом клиентском компьютере.

13. Способ по п.12, в котором этап синхронизации первой части на втором клиентском компьютере с серверным компьютером, чтобы добавлять, в первую часть на серверном компьютере, изменения, сделанные в содержимом электронного документа на втором клиентском компьютере, к изменениям, сделанным в электронном документе на первом клиентском компьютере, дополнительно содержит этапы, на которых:

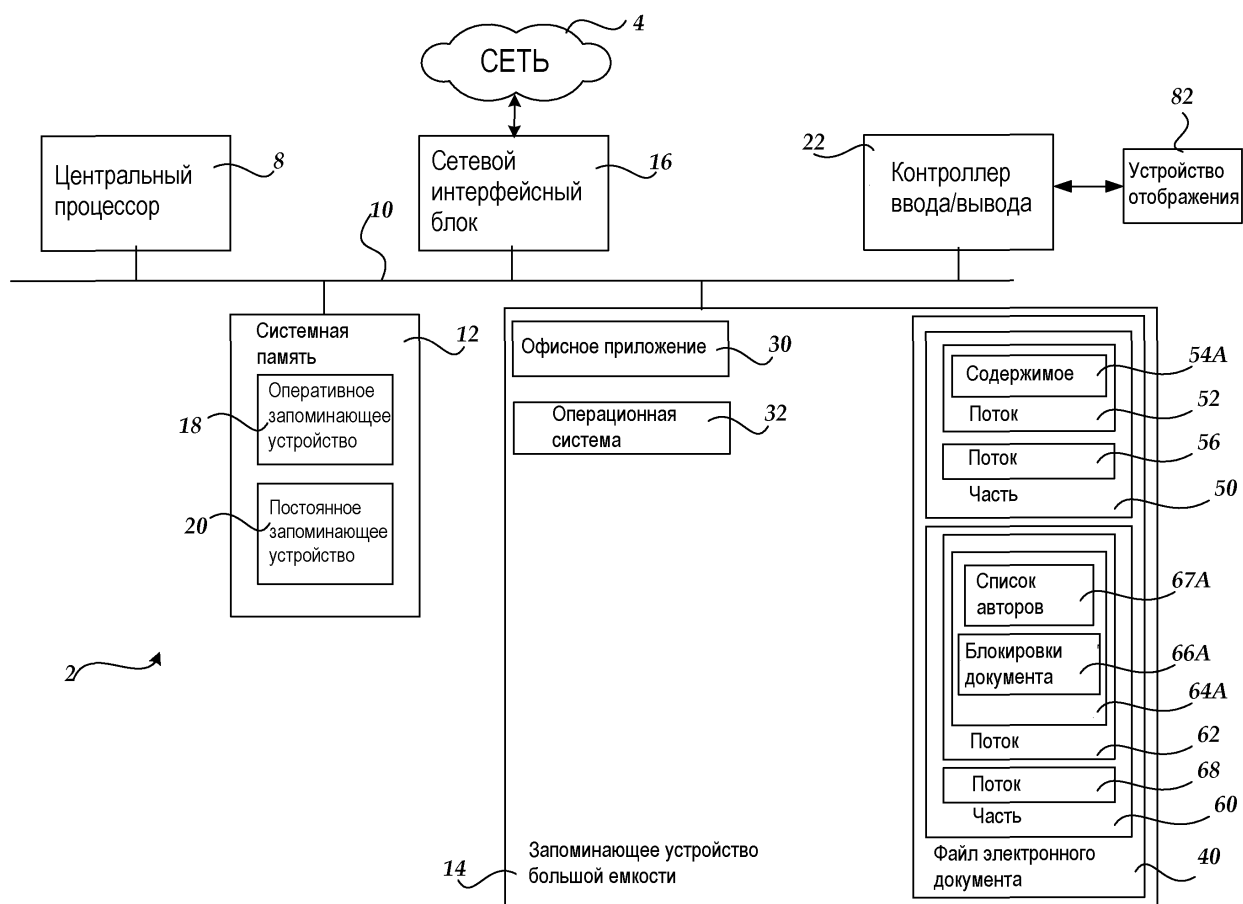
35 проверяют серверный компьютер на обновления в первой части; и

принимают подтверждение от серверного компьютера, что первая часть была обновлена.

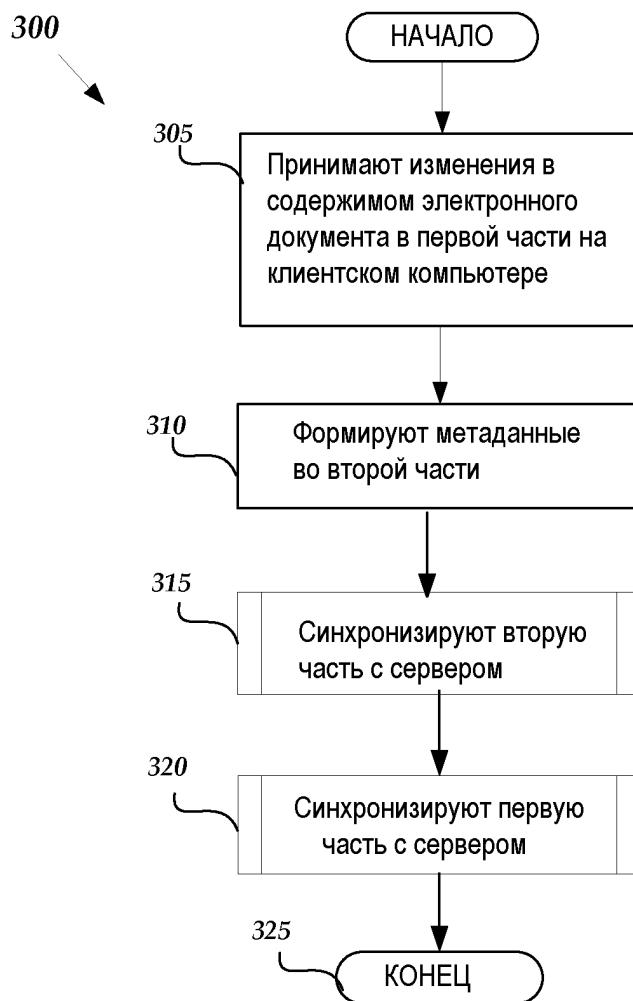
14. Способ по п.12, в котором этап синхронизации первой части на втором клиентском компьютере с серверным компьютером, чтобы добавлять, в первую часть на серверном компьютере, изменения, сделанные в содержимом электронного документа на втором клиентском компьютере, к изменениям, сделанным в электронном документе на первом клиентском компьютере, дополнительно содержит этапы, на которых:

принимают, от серверного компьютера, изменения, сделанные в содержимом электронного документа на первом клиентском компьютере; и

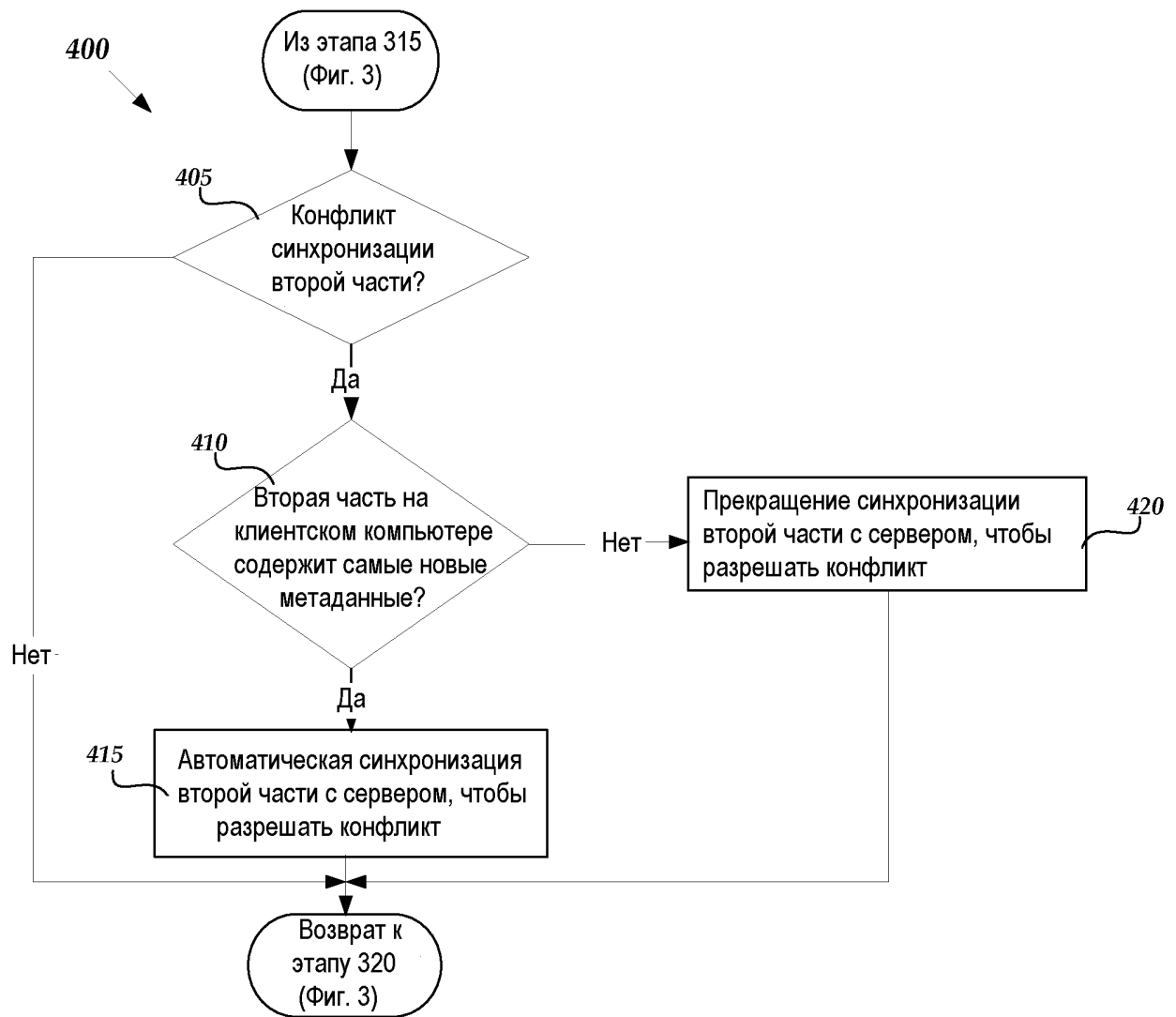
45 обновляют первую часть на серверном компьютере изменениями, сделанными в содержимом электронного документа на втором клиентском компьютере и первом клиентском компьютере.



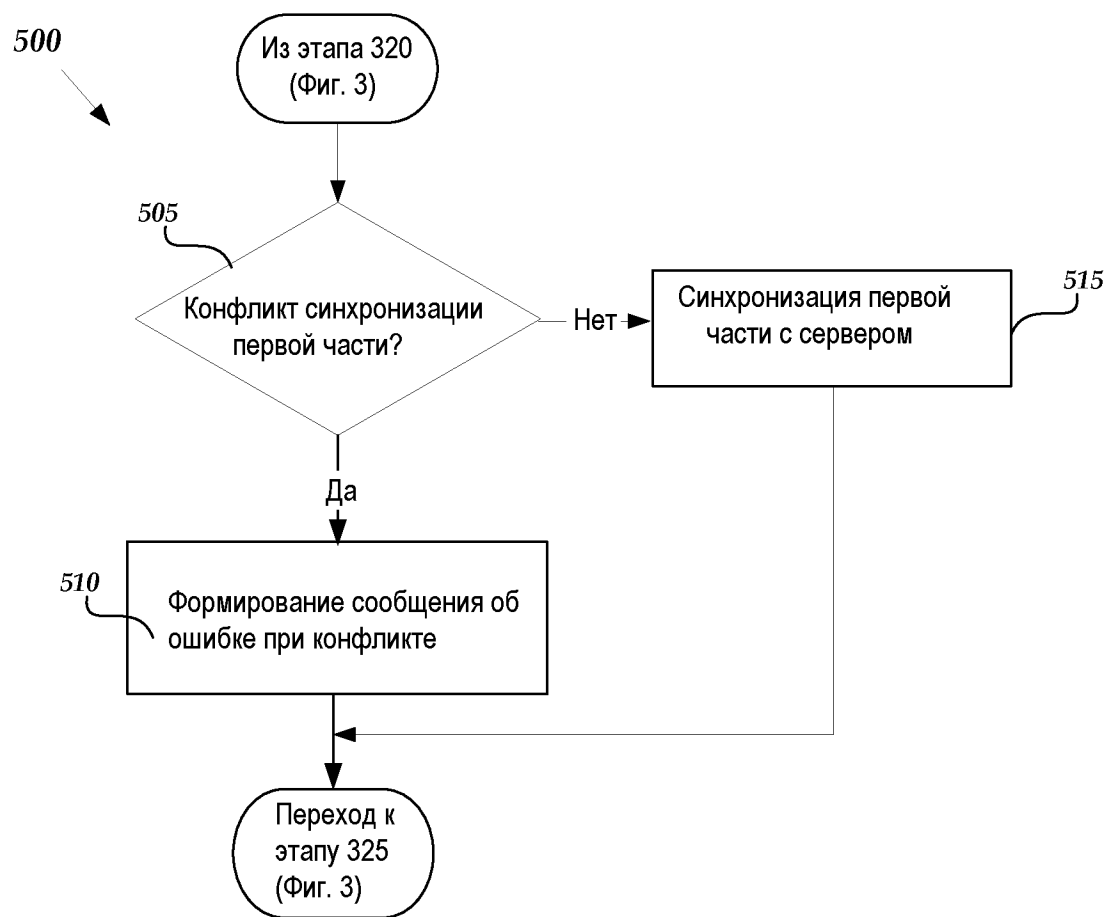
Фиг. 2



Фиг. 3

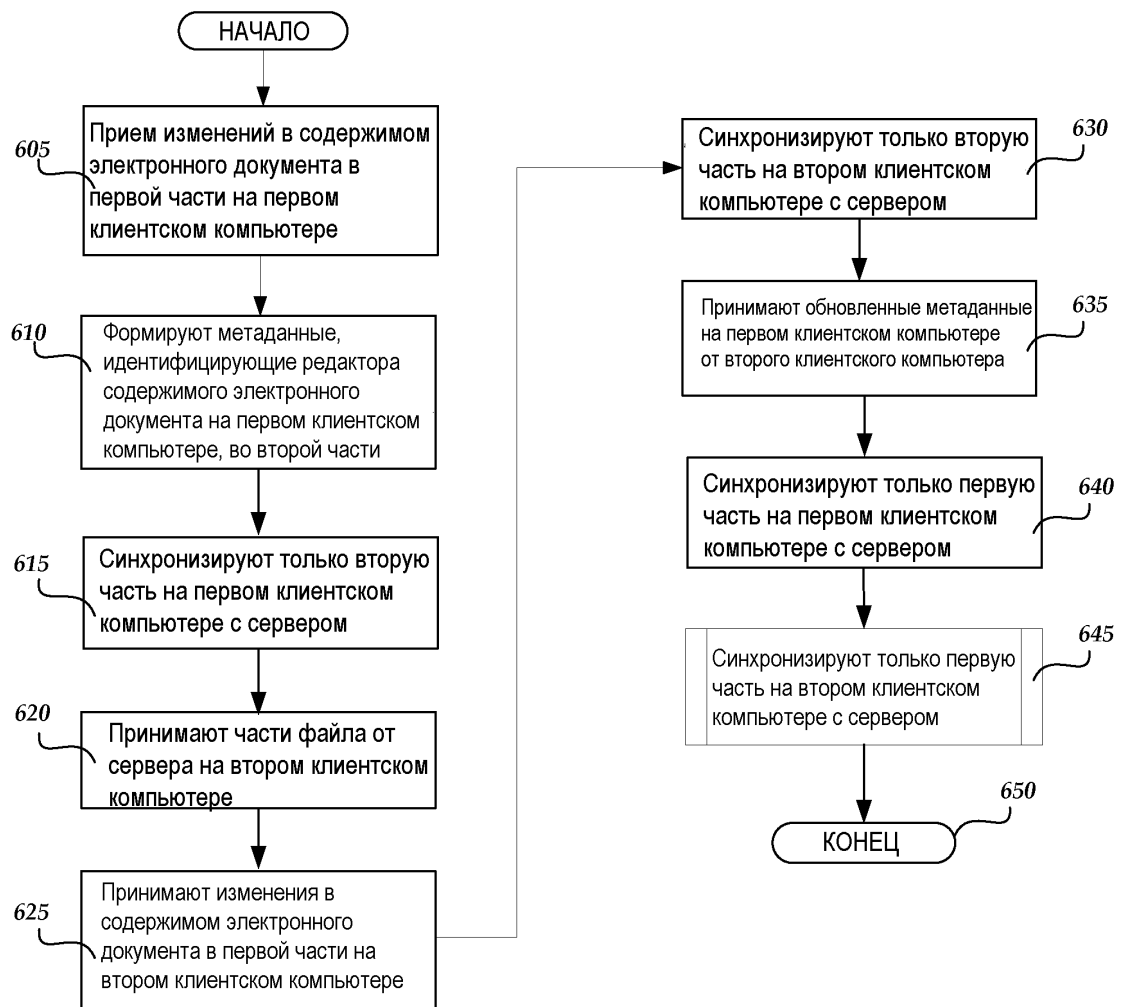


Фиг. 4

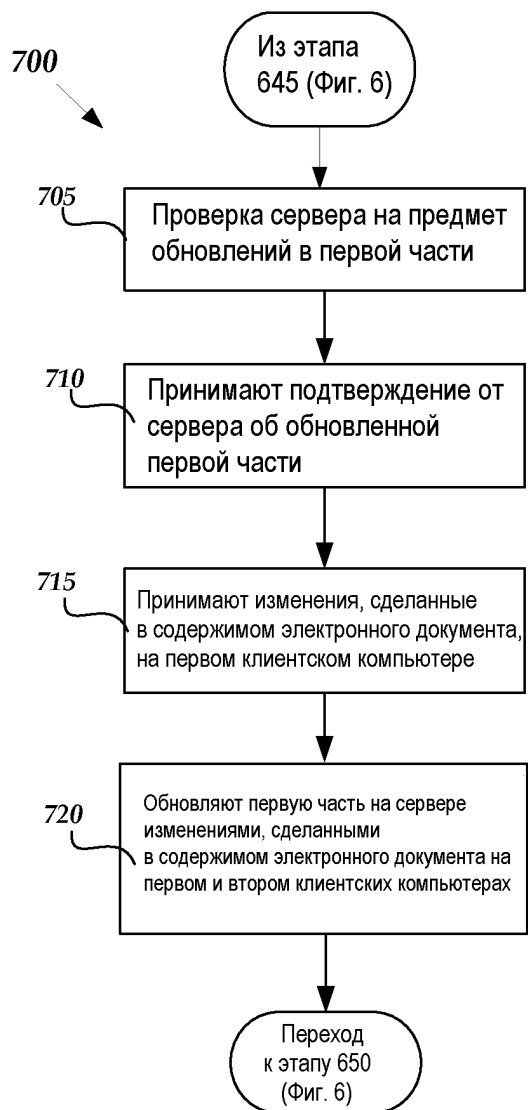


Фиг. 5

600



Фиг. 6



Фиг. 7