



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: 2004105884/09, 27.02.2004

(24) Дата начала отсчета срока действия патента:
27.02.2004

(30) Конвенционный приоритет:
28.02.2003 US 10/376,738

(43) Дата публикации заявки: 10.08.2005

(45) Опубликовано: 20.03.2009 Бюл. № 8

(56) Список документов, цитированных в отчете о поиске: US 6449624 B1, 10.09.2002. Developing Software, Release 2002.05.00, Windows, Rational ClearCase, 2001, найден по адресу URL: <http://www.agsrhichome.bnl.gov/Controls/doc/ClearCaseEvn/5.0/ccase-nt/ccdev>, разделы 1.2, 1.3, 1.5, 1.7, 2.4, 2.5, 3.1-3.3, 3.5, 4.1-4.3, 5.1-5.3, 6.3. US 6256773 B1, 03.07.2001. RU 2108621 C1, 10.04.1998.

Адрес для переписки:
129090, Москва, ул. Б.Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры", пат.пов. Ю.Д.Кузнецову, рег.№ 595

(72) Автор(ы):

ЭДЕЛЬШТАЙН Ноа (US),
САЛИБА Хани (US),
ЛИУ Хай (US),
РЭЙСОН Стивен Джэймс (US)

(73) Патентообладатель(и):

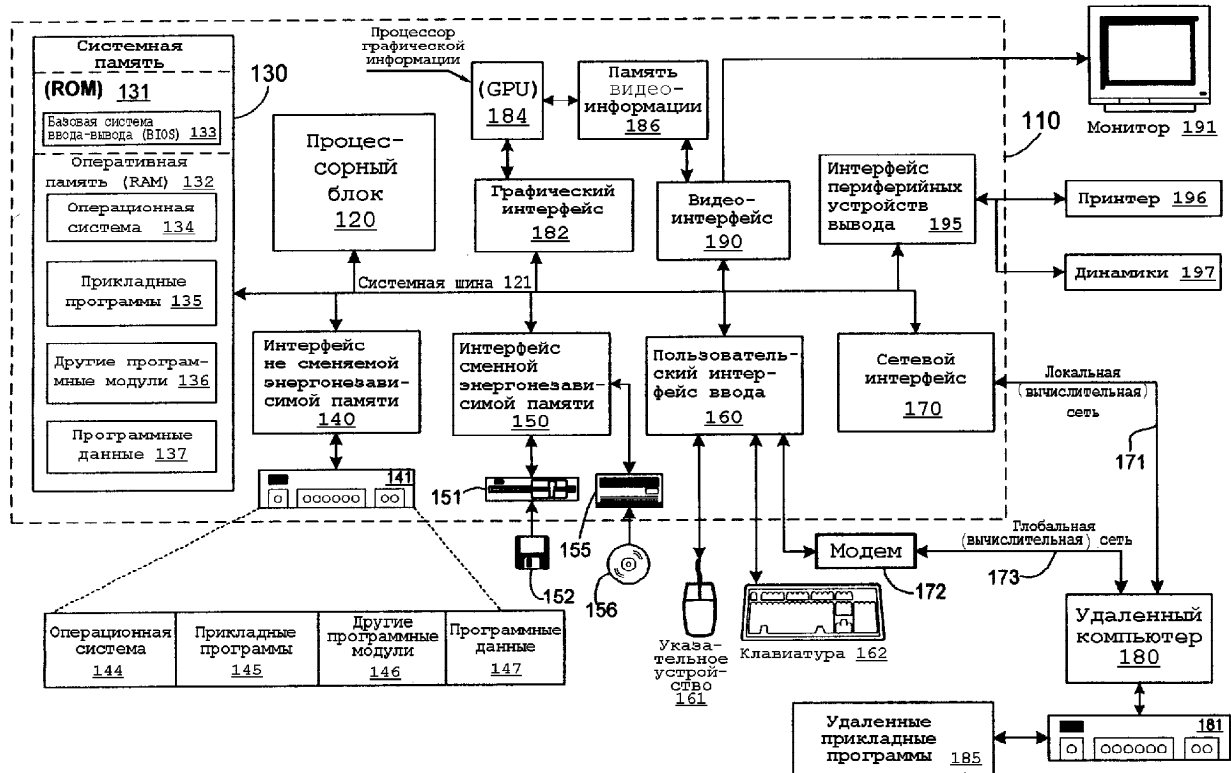
МАЙКРОСОФТ КОРПОРЕЙШН (US)

(54) СПОСОБ УПРАВЛЕНИЯ РЕПЛИКАЦИЕЙ ФАЙЛОВ В ПРИЛОЖЕНИЯХ

(57) Реферат:

Изобретение относится к системам управления версиями электронного файла. Техническим результатом является обеспечение возможности запрашивания процесса синхронизации файла и управления этим процессом. При осуществлении способа распознают первую и вторую версии электронного файла и инициируют событие синхронизации. Программный интерфейс приложения осуществляет доступ к первой и второй версиям, сравнивает информацию,

связанную с каждой из них, генерирует результат и сообщает этот результат первому приложению программного обеспечения. Второе приложение программного обеспечения генерирует пользовательский интерфейс в соответствии с признаком второго приложения программного обеспечения, согласно которому электронный файл связан со вторым приложением программного обеспечения, а пользовательский интерфейс отображается на отображающем устройстве. 2 н. и 18 з.п. ф-лы, 4 ил.



Фиг. 1



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: 2004105884/09, 27.02.2004

(24) Effective date for property rights: 27.02.2004

(30) Priority:
28.02.2003 US 10/376,738

(43) Application published: 10.08.2005

(45) Date of publication: 20.03.2009 Bull. 8

Mail address:

129090, Moskva, ul. B.Spasskaja, 25, str.3,
OOO "Juridicheskaja firma Gorodisskij i
Partnery", pat.pov. Ju.D.Kuznetsovu, reg.№ 595

(72) Inventor(s):

EhDEL'ShTAJN Noa (US),
SALIBA Khani (US),
LIU Khaj (US),
REhJSON Stiven Dzhehjms (US)

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)

(54) METHOD OF FILE REPLICATION MANAGEMENT IN APPLICATIONS

(57) Abstract:

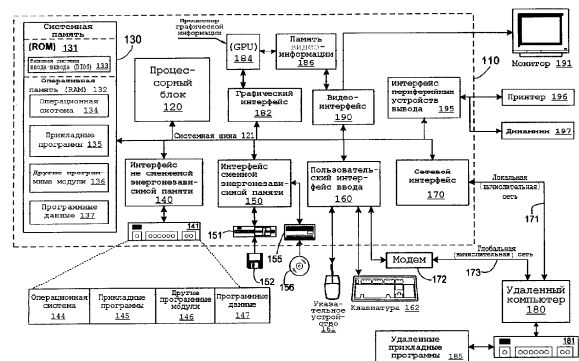
FIELD: information technologies.

SUBSTANCE: during realisation of the method, distinguish the first and second versions of an electronic file and initiate synchronisation event. The program interface of the application provides access to the first and second versions, compares the information connected with each of them, generates result and informs this result to the first application of the software. The second application of the software generates the user interface according to sign of the second software application according to which the electronic file is connected with the second software application, and the user interface is displayed on the displaying device.

EFFECT: possibility of requesting the process of a file synchronisation and process management.

20 cl, 4 dwg

Вычислительная среда 100



Фиг. 1

Настоящая заявка на патент связана с принадлежащей тому же правообладателю и одновременно рассматриваемой заявкой № AA/BBBCCC на патент США, поданной 27 февраля 2003 г. под названием "Method and System for Managing Multiple File States" («Способ и система для управления множественными состояниями файлов») и имеющей
5 номер MSFT-1586 в реестре патентного поверенного.

Область техники, к которой относится изобретение

Настоящее изобретение посвящено способам управления множественными версиями электронного файла. Более конкретно, настоящее изобретение относится к способам использования функциональных возможностей релевантных приложений для согласования
10 версий файла во время события синхронизации.

Предпосылки создания изобретения

В среде компьютеров, объединенных в сеть, пользователь компьютера обычно участвует в совместной работе по созданию некоторого электронного документа вместе с другим пользователем сети. Такая сеть может быть локальной (вычислительной) сетью
15 (LAN), глобальной (вычислительной) сетью (WAN), сетью Internet и т.п. В процессе такой совместной работы каждый пользователь может создавать и редактировать электронный файл, пользуясь приложением программного обеспечения, таким как прикладное программное обеспечение общего назначения или программное обеспечение, разработанное специально для того типа файла, который является предметом совместной
20 работы, и т.п.

Проведение совместной работы возможно путем поддержания главной версии электронного файла в месте, доступном для всех пользователей, которые будут совместно работать над файлом. В обычной среде компьютеров, объединенных в сеть, могут существовать серверный компьютер и один или более пользовательских компьютеров.
25 Сервер поддерживает каналы связи между пользовательскими компьютерами и доступ к каждому из них, вследствие чего может хранить главную версию файла, являющегося предметом совместной работы пользователей, и управлять этой версией. Участвуя в совместной работе над электронным файлом, пользователь может сначала осуществить доступ к главной версии файла, находящейся на сервере, а затем скопировать этот файл
30 в свой компьютер. Затем пользователь может работать над файлом, изменяя этот файл любым способом, который приложение обеспечивает этому пользователю. Например, если приложение представляет собой программу подготовки и редактирования текстов, программу подготовки и редактирования электронных таблиц, базу данных и/или нечто подобное, это приложение сможет обеспечить средство для решения таких задач, как редактирование, копирование, исключение и/или подобные им задачи. По завершении
35 любых изменений пользователь может сохранить изменения в версию файла, хранящуюся на пользовательском компьютере, если этот пользователь захочет зафиксировать эти изменения в файл. В других системах, таких как система, в которой нет серверного компьютера, главная версия может храниться в пользовательском компьютере.

Чтобы принять участие в совместной работе с другими пользователями через
40 посредство сервера, пользователь, обладающий измененной версией, хранящейся в пользовательском компьютере, должен синхронизировать версию файла, находящуюся в его компьютере, с главной версией файла, находящейся в сервере. Синхронизация множества версий электронного файла в ее простейшей форме гарантирует, что все
45 версии сразу же после синхронизации станут идентичными. Таким образом, пользователь, осуществляющий синхронизацию, гарантирует всем пользователям, принимающим участие в совместной работе, доступ ко вновь измененному файлу. Обычно синхронизация осуществляется синхронизирующей утилитой. Синхронизирующая утилита обычно отделена от приложения и во многих случаях является частью операционной системы
50 компьютера пользователя. Синхронизирующая утилита в типичном случае отделена от приложения, а во многих случаях является частью операционной системы пользовательского компьютера. Обычная синхронизирующая утилита осуществляет синхронизацию в предварительно определенные моменты времени или по запросам

операционной системы.

При обычной синхронизации синхронизирующая утилита будет сравнивать пользовательскую версию файла с главной версией файла всякий раз, когда появляется информация о копиях, которую этот объект «в состоянии осмыслить». Как правило, такая информация сводится к дате и/или времени последнего обновления каждой версии. Если дата и/или время оказываются идентичными, то синхронизирующая утилита определяет, что обе версии идентичны и поэтому синхронизированы, так что никакое дополнительное действие не требуется. Однако если имеется различие в дате и/или времени, синхронизирующая утилита предоставляет пользовательский интерфейс пользователю, связанному с компьютером, запускающим событие синхронизации. Пользовательский интерфейс обычно предоставляет пользователю краткое уведомление о наличии различия, дату и/или время этого последнего обновления каждой версии файла, а также запрос команды пользователя по поводу того, какую копию файла пользователь хочет поддерживать. Если пользователь выбирает поддержание пользовательской версии, то синхронизирующая утилита записывает пользовательскую версию поверх главной версии (затирает главную версию пользовательской версией), тем самым делая пользовательскую версию главной версией в сервере. Если пользователь желает поддержать главную версию, то никакое действие не предпринимается или главная версия записывается поверх пользовательской версии (пользовательская версия затирается главной версией). В некоторых интерфейсах пользователя пользователю предоставляется опция отказа от синхронизации, а не выбора версии, становящейся главной версией.

Обычная синхронизация, о которой шла речь выше, имеет серьезные недостатки. Поскольку обычная синхронизирующая утилита имеет ограниченные функциональные возможности применительно к электронному файлу, пользовательский интерфейс, создаваемый обычным объектом, предоставляет пользователю лишь ограниченное количество информации, приглашая пользователя выбрать образ действий. Поэтому такое ограниченное количество информации может сделать пользователя не информированным, вследствие чего он может принять неправильное решение о том, какой образ действий следует предпочесть. Например, первый пользователь может изменить ранее синхронизированную пользовательскую версию файла, после чего обычная синхронизирующая утилита в некоторый момент иницирует событие синхронизации. После уведомления посредством обычного интерфейса пользователя пользователь может заметить, что пользовательская версия обновлена позже, чем главная версия, и поэтому принять решение о записи поверх главной версии. Однако информация, представленная посредством обычного интерфейса пользователя, не уведомляет пользователя о том, осталась ли главная версия неизменной после того, как пользователь начал изменять пользовательскую версию. Это приводит к проблемной ситуации, в которой второму пользователю может потребоваться внесение изменений в другую пользовательскую версию, а затем синхронизировать эту пользовательскую версию перед синхронизацией, проводимой вследствие работы первого пользователя. Версия первого пользователя является более новой, т.е. обновленной позже, но второй пользователь подверг главную версию изменениям, которые не были внесены, когда первый пользователь первоначально копировал файл из сервера. В результате изменения, внесенные вторым пользователем в главную версию, оказываются утраченными сразу же после того, как первый пользователь завершает синхронизацию.

Вышеупомянутая ситуация вызывается присущей утилите, осуществляющей обычную синхронизацию, недостаточностью полного диапазона функциональных возможностей в приложении, релевантном (имеющем отношение) к файлу, вследствие чего эта утилита оказывается неспособной воспользоваться такими функциональными возможностями во время события синхронизации. Следовательно, такая утилита может предоставить пользователю лишь ограниченное количество информации, касающейся различных версий электронного файла и представляющей собой, например, такие рассмотренные выше сведения, как дата и/или время последнего обновления каждой версии файла.

Другой недостаток, вызываемый ограниченными функциональными возможностями утилит, осуществляющих обычную синхронизацию, является недостаток множественных действий, доступных пользователю во время синхронизации. Так, в примере, рассмотренном выше, пользовательский интерфейс предоставляет лишь три опции:

- 5 обновление (замена) главной копии пользовательской копией, обновление (замена) пользовательской копии главной копией или отказ.

Дополнительный недостаток утилиты, осуществляющей обычную синхронизацию, вытекает из недостатка интеграции такой утилиты с таким приложением. Например, как говорилось выше, утилита, осуществляющая обычную синхронизацию, является частью
10 операционной системы пользовательского или серверного компьютера. Следовательно, процесс синхронизации, как и любой имеющийся пользовательский интерфейс, обычно не учитывает предысторию работы пользователя с приложением. Это может привести к путанице среди пользователей, которые - хотя и могут иметь опыт работы с приложением - не имеют опыта работы с утилитой операционной системы. Кроме того - и в
15 результате недостатка интеграции, - синхронизация может инициироваться синхронизирующей утилитой в момент, не являющийся оптимальным для наступления такого события. Дополнительным недостатком обычных утилит синхронизации является невозможность использования такой утилиты в качестве компонента приложения программного обеспечения. Например, пользуясь обычными способами, разработчик
20 компьютерных программ не сможет создать первое приложение программного обеспечения, в котором используется синхронизирующей утилитой и которое способно использовать второе приложение, подходящее для электронного файла в плане согласования множественных версий этого электронного файла. Следовательно, существует потребность в эффективном средстве использования функциональных
25 возможностей релевантных приложений программного обеспечения для разрешения конфликтов между копиями файла во время синхронизации.

Краткое изложение сущности изобретения

Ввиду вышеизложенного в настоящем изобретении предложен программный интерфейс приложения, обеспечивающий приложению возможность запрашивать процесс
30 синхронизации файла и управлять этим процессом с учетом предыстории работы пользователя с приложением. В изобретении предложен способ управления первой и второй версиями электронного файла. При осуществлении этого способа распознают первую и вторую версии электронного файла и инициируют событие синхронизации с помощью первого приложения программного обеспечения, работающего на первом
35 компьютере. Программный интерфейс приложения осуществляет доступ к первой и второй версиям, сравнивает информацию, связанную с каждой из них, генерирует результат и сообщает этот результат первому приложению программного обеспечения. Второе приложение программного обеспечения генерирует пользовательский интерфейс в соответствии с особенностью второго приложения программного обеспечения, когда
40 электронный файл связан со вторым приложением программного обеспечения, а пользовательский интерфейс отображается на отображающем устройстве.

В изобретении также предусмотрена возможность генерирования опции синхронизации вторым приложением программного обеспечения, а также предоставления этой опции с помощью интерфейса пользователя и отображения интерфейса пользователя на
45 отображающем устройстве. Второе приложение программного обеспечения также может принимать вариант выбора опции и соответственно синхронизировать первую и вторую версии электронного файла. Изобретение также предусматривает, что первое приложение программного обеспечения и второе приложение программного обеспечения могут быть одним и тем же приложением программного обеспечения. Это приложение может
50 обеспечивать предоставление пользователю любого количества информации, касающейся первой и второй версий, а пользователь может вводить любое количество или любой тип вариантов выбора синхронизации. В одном конкретном варианте осуществления пользователь может создать третью версию, содержащую любую часть либо первой либо

второй версии или их обе или не содержащую ни одну часть ни первой, ни второй версии, ни их обе. Это приложение может инициировать синхронизацию в любой момент, например, по запросу пользователя. Дополнительные признаки и преимущества изобретения изложены в нижеследующем описании.

5 Краткое описание чертежей

Предшествующее краткое изложение сущности изобретения, а также нижеследующее подробное описание предпочтительных конкретных вариантов осуществления станут более понятными при изучении их со ссылками на прилагаемые чертежи. В целях иллюстрации изобретения на чертежах приведены возможные конкретные варианты осуществления изобретения, однако изобретение не сводится к описываемым ниже конкретным способам и аппаратуре. В связи с чертежами нужно отметить, что:

на фиг. 1 представлена блок-схема, иллюстрирующая возможную вычислительную среду, в которой можно реализовать аспекты изобретения;

на фиг. 2 представлена функциональная схема, иллюстрирующая пример взаимосвязи между первым компьютером, сетью и вторым компьютером;

на фиг. 3 представлен алгоритм, иллюстрирующий процесс синхронизации в соответствии с одним конкретным вариантом осуществления настоящего изобретения;

на фиг. 4 представлен алгоритм, иллюстрирующий способ инициирования процесса синхронизации в соответствии с одним конкретным вариантом осуществления настоящего изобретения.

Подробное описание иллюстративных конкретных вариантов осуществления изобретения

Обзор

В настоящем изобретении предложены способы интеграции программного интерфейса приложения с приложением программного обеспечения, что обеспечивает этому приложению возможность запрашивать процесс синхронизации файла и управлять этим процессом с учетом предыстории (опыта) работы пользователя с приложением. В изобретении предложен способ управления первой и второй версиями электронного файла. При осуществлении этого способа распознают первую и вторую версии электронного файла и инициируют событие синхронизации для осуществления синхронизации первой версии, находящейся на первом компьютере, со второй версией с помощью первого приложения программного обеспечения, работающего на первом компьютере. Программный интерфейс приложения осуществляет доступ к первой и второй версиям, сравнивает информацию, связанную с каждой из первой и второй версий, и генерирует результат. Результат сравнения сообщается первому приложению программного обеспечения. Обеспечивают генерирование интерфейса пользователя вторым приложением программного обеспечения в соответствии с признаком второго приложения программного обеспечения, по которому электронный файл связан со вторым приложением программного обеспечения, а пользовательский интерфейс отображается на отображающем устройстве.

В изобретении также предусмотрена возможность генерирования опции синхронизации вторым приложением программного обеспечения, а также обеспечивается предоставление этой опции синхронизации с помощью интерфейса пользователя и отображение на отображающем устройстве. Второе приложение программного обеспечения также принимает вариант выбора опции и синхронизирует первую и вторую версии электронного файла в соответствии с этим вариантом выбора опции. Изобретение также предусматривает, что первое приложение программного обеспечения и второе приложение программного обеспечения могут быть одним и тем же приложением программного обеспечения, вследствие чего все этапы могут выполняться с помощью единственного приложения.

Это приложение может обеспечивать предоставление пользователю любого количества информации, касающейся первой и второй версий, а пользователь получает возможность вводить любое количество или любой тип вариантов выбора синхронизации. В одном

конкретном варианте осуществления пользователь может создать третью версию, содержащую любую часть либо первой, либо второй версии или их обе или не содержащую ни одну часть ни первой, ни второй версии, ни их обе. Это приложение может инициировать синхронизацию в любой момент, например, по запросу пользователя.

5 Пример вычислительной среды

На фиг. 1 изображена подходящая вычислительная системная среда 100, в которой можно осуществить изобретение. Вычислительная системная среда 100 представляет собой лишь один пример подходящей вычислительной среды и не предполагает внесение
10 возможностей изобретения. Также не следует интерпретировать вычислительную среду 100 как имеющую какую-либо зависимость или подчиняющуюся какому-либо требованию в связи с любым из иллюстрируемых компонентов возможной операционной среды 100 или с любой их совокупностью.

Изобретение работоспособно и со многими другими вычислительными системными
15 средами или конфигурациями общего назначения или специального назначения. Примеры хорошо известных вычислительных систем, сред и/или конфигураций, которые могут оказаться подходящими для использования совместно с предлагаемым изобретением, включают в себя, но не в ограничительном смысле, персональные компьютеры, серверные компьютеры, карманные или портативные компьютерные устройства, многопроцессорные
20 системы, системы на основе микропроцессоров, компьютерные приставки к телевизорам, программируемые потребительские электронные приборы, сетевые персональные компьютеры (ПК), миникомпьютеры, универсальные компьютеры, распределенные вычислительные среды, которые включают в себя любые из вышеупомянутых систем или устройств и т.п.

Изобретение может быть описано в обобщенном контексте исполняемых компьютером команд, таких, как программные модули, исполняемые компьютером. В общем случае, программные модули включают в себя подпрограммы, программы, объекты, компоненты, структуры данных и т.п., которые решают конкретные задачи или реализуют конкретные абстрактные типы данных. Изобретение также может быть воплощено в распределенных
30 вычислительных средах, в которых задачи решаются удаленными процессорными устройствами, связанными посредством сети связи или другой среды передачи данных. В распределенной вычислительной среде программные модули и другие данные могут находиться на компьютерных носителях информации, включая запоминающие устройства как локальных, так и удаленных компьютеров.

Обращаясь к фиг. 1, отмечаем, что возможная система для осуществления изобретения включает в себя вычислительное устройство общего назначения в виде компьютера 110. Компоненты компьютера 110 могут включать в себя, но не в ограничительном смысле, процессорный блок 120, системную память 130 и системную шину 121, которая связывает различные системные компоненты, включая системную память, с процессорным блоком
40 120. Системная шина 121 может относиться к любому из нескольких типов структур шин, включая шину памяти или контроллер памяти, периферийную шину и локальную шину, и в ней может быть использовано множество шинных архитектур (топологий). В качестве примера, не носящего ограничительный характер, можно отметить, что такие архитектуры включают в себя шину стандартной промышленной архитектуры (шину ISA), шину
45 микроканальной архитектуры (шину MCA), шину расширенной стандартной промышленной архитектуры (шину EISA), локальную шину стандарта «Ассоциации по стандартизации в области видеоэлектроники» (шину VESA) и шину взаимного соединения периферийных компонентов (шину PCI) (также известную под названием «шина второго уровня»).

Компьютер 110 в типичном случае включает в себя совокупность считываемых
50 компьютером носителей. Считываемые компьютером носители могут быть любыми имеющимися в продаже носителями, доступ к которым можно получить с помощью компьютера 110, и включают в себя как энергозависимые, так и энергонезависимые носители, как сменные, так и несменяемые носители. В качестве примера, не носящего

ограничительный характер, следует отметить, что считываемые компьютером носители информации могут представлять собой компьютерные носители информации и средства компьютерной связи. Компьютерные носители информации включают в себя как энергозависимые, так и энергонезависимые, как сменные или несменяемые носители, реализованные с помощью любого способа или любой технологии запоминания информации, такой, как считываемые компьютером команды, структуры данных, программные модули или иные данные. Считываемые компьютером носители включают в себя, но не в ограничительном смысле, оперативную память (RAM), называемую также памятью с произвольным доступом, постоянную память (ROM), электрически стираемую программируемую постоянную память (EEPROM), флэш-память или память, созданную по иной технологии, постоянную память на компакт-дисках (CD-ROM), память на универсальных цифровых дисках (DVD) или иных оптических дисках, запоминающие устройства на магнитных лентах, магнитных дисках или иных магнитных носителях или любой другой носитель, который можно использовать для запоминания желаемой информации и доступ к которому возможен с помощью компьютера 110. Средства связи обычно воплощают считываемые компьютером команды, структуры данных, программные модули или иные данные в виде модулированного сигнала, такого как несущая волна, или реализуют иной транспортный механизм, и включают в себя любые средства доставки информации. Термин «модулированный информационный сигнал» означает сигнал, одна или более характеристик которого задаются или изменяются с обеспечением возможности кодирования информации в этом сигнале. В качестве примера, не носящего ограничительный характер, следует отметить, что средства связи включают в себя проводные средства такие, как проводная сеть или соединение прямого доступа, и беспроводные средства, такие как акустические, радиочастотные, инфракрасные и иные беспроводные средства. К числу считываемых компьютером носителей относятся также комбинации любых из тех, которые упоминались выше.

Системная память 130 включает в себя компьютерные носители информации в форме энергозависимой и/или энергонезависимой памяти, такие как ROM 131 или RAM 132. В ROM 131, как правило, хранится базовая система 133 ввода-вывода (BIOS), содержащая базовые подпрограммы, способствующие передаче информации между элементами внутри компьютера 110, например, во время запуска. RAM 132 в типичном случае содержит информационные и/или программные модули, к которым возможен немедленный доступ и/или с которыми в данный момент работает процессорный блок 120. В качестве примера, не носящего ограничительный характер, отметим, что на фиг. 1 изображены операционная система 134, прикладные программы 135, другие программные модули 136 и программные данные 137.

Компьютер 110 также может включать в себя другие - сменные и/или несменяемые, энергозависимые и/или энергонезависимые - компьютерные носители информации. Лишь в качестве примера, следует отметить, что на фиг. 1 изображен накопитель 142 на жестких дисках, который осуществляет считывание с несменяемых энергонезависимых магнитных носителей и запись на них, накопитель 151 на магнитных дисках, который осуществляет считывание со сменного энергонезависимого магнитного диска 152 и запись на него, и накопитель 155 на оптических дисках, который осуществляет считывание со сменного энергонезависимого оптического диска 156, такого как CD-ROM или другой оптический носитель, и запись на него. Другие сменные и/или несменяемые, энергозависимые и/или энергонезависимые компьютерные носители информации, которые можно использовать в возможной операционной среде, включают в себя, но не в ограничительном смысле, кассеты с магнитными лентами, платы флэш-памяти, универсальные цифровые диски, цифровую видеоленту, твердотельную оперативную память (RAM), твердотельную постоянную память (ROM) и т.п. Накопитель 141 на жестких дисках в типичном случае подсоединен к системной шине 121 посредством интерфейса несменяемой памяти, такого как интерфейс 140, а накопитель 151 на магнитных дисках и накопитель 155 на оптических дисках в типичном случае подсоединены к системной шине 121 посредством

интерфейса сменной памяти, такого как интерфейс 150.

Накопители и связанные с ними компьютерные носители информации, рассмотренные выше и изображенные на фиг. 1, обеспечивают запоминание считываемых компьютером команд, структур данных, программных модулей и других данных для компьютера 110.

5 Например, накопитель 141 на жестких дисках показан на фиг. 1 как хранящий операционную систему 144, прикладные программы 145, другие программные модули 146 и программные данные 147. Отметим, что эти компоненты могут быть либо такими же, как операционная система 134, прикладные программы 135, другие программные модули 136 и программные данные 137, либо другими. Операционная система 144, прикладные
10 программы 145, другие программные модули 146 и программные данные 147 обозначены здесь разными позициями, чтобы проиллюстрировать, что они, как минимум, представляют собой разные копии. Пользователь может вводить команды и информацию в компьютер 110 посредством устройств ввода, таких как клавиатура 162 и указательное устройство 161, в общем случае именуемое мышью, шаровым указателем (трекболом) или сенсорной
15 панелью. Другие устройства (не показаны) могут включать в себя микрофон, рычажный указатель (джойстик), игровую панель, спутниковую антенну-тарелку, сканер и т.п. Эти и другие устройства ввода часто подсоединяют к процессорному блоку 120 посредством пользовательского интерфейса 160 ввода, который связан с системной шиной, но могут быть подсоединены и посредством других структур интерфейсов и шин, например,
20 посредством параллельного порта, игрового порта или универсальной последовательной шины (USB). К системной шине 121 через такой интерфейс, как видеоинтерфейс 190, также подсоединен монитор 191 или отображающее устройство другого типа. Помимо монитора компьютеры также могут включать в себя другие периферийные устройства вывода, такие как динамики 197 и принтер 196, которые можно подсоединить с помощью интерфейса 195
25 периферийных устройств вывода.

Компьютер 110 может работать в сетевой среде с использованием логических соединений с одним или более удаленными компьютерами, такими как удаленный компьютер 180. Удаленный компьютер 180 может быть персональным компьютером, сервером, маршрутизатором, сетевым персональным ПК, равноправным (одноранговым)
30 устройством или иным обычным сетевым узлом, и в типичном случае включает в себя многие из элементов, описанных выше в связи с компьютером 110, или все эти элементы, хотя на фиг. 1 изображено лишь одно запоминающее и хранящее устройство 181. Изображенные логические соединения включают в себя локальную (вычислительную) сеть (LAN) 171 и глобальную (вычислительную) сеть (WAN) 173, но могут включать в себя и
35 другие сети. Такие сетевые среды обычно практикуются в жилых домах, офисах, компьютерные сетях предприятий, интрасетях и сети Internet.

При использовании в сетевой среде LAN компьютер 110 подсоединяют к LAN 171 посредством сетевого интерфейса или адаптера 170. При использовании в сетевой среде WAN компьютер 110 в типичном случае включает в себя модем 172 или другие средства
40 для установления связи через WAN 173, такую как Internet. Модем 172, который может быть внутренним или внешним, можно подсоединять к системной шине 121 через пользовательский интерфейс 160 ввода или с помощью другого подходящего механизма. В сетевой среде программные модули, показанные относящимися к компьютеру 110, или их части можно хранить в удаленном запоминающем и хранящем устройстве. В качестве
45 примера, не носящего ограничительный характер, удаленные прикладные программы 185 изображены на фиг. 1 как записанные в запоминающем устройстве 181. Следует понять, что проиллюстрированные сетевые соединения являются лишь возможными, и для установления канала связи между компьютерами можно использовать другие средства.

Примерные распределенные вычислительные структуры или архитектуры
50 В свете слияния персональных вычислений и Internet уже разработаны и разрабатываются в настоящее время различные распределенные вычислительные структуры. Вследствие этого пользователям, являющимся физическими и юридическими лицами, предоставляется интерфейс прямого доступа, обеспечивающий возможности

взаимодействия и работы с web-страницами для приложений и вычислительных устройств, который представляет собой справляющийся с нарастающим объемом вычислительных операций web-браузер или браузер, ориентированный на сеть.

Например, платформа MICROSOFT®'s.NET включает в себя серверы, средства

- 5 предоставления услуг по созданию блоков, такие как средства хранения данных на основе web-страниц и загружаемое программное обеспечение устройств. Вообще говоря, платформа MICROSOFT®'s.NET обеспечивает: (1) возможность совместной работы всей номенклатуры вычислительных устройств, а также автоматического обновления и синхронизации информации пользователя со всеми этими устройствами; (2) повышенную
- 10 работоспособность в интерактивном режиме для web-сайтов, гарантируемую расширенным использованием такого языка форматирования web-документов, как расширяемый язык разметки (XML), вместо языка гипертекстовой разметки (HTML); (3) разновидности оперативного обслуживания (услуги), которые реализуют настраиваемый доступ и предоставление продуктов и услуг пользователю из центрального исходного пункта для
- 15 управления различными приложениями, например, такими, как электронная почта, или программным обеспечением, например, таким, как Office.NET; (4) централизованное хранение данных, которое будет увеличивать эффективность доступа к информации и упрощать его, а также синхронизацию информации между пользователями и устройствами; (5) возможность интеграции различных средств связи, таких как электронная почта, факсы и телефоны; (6) для разработчиков - возможность создавать модули многоразового
- 20 использования, увеличивая тем самым производительность и уменьшая количество ошибок программирования; а также (7) многие другие особенности межплатформной интеграции.

- Хотя возможные конкретные варианты осуществления описаны здесь в связи с программным обеспечением, установленным на вычислительном устройстве, одну или
- 25 более частей изобретения также можно реализовать посредством операционной системы, программного интерфейса приложения (API) или «промежуточного» объекта между сопроцессором и запрашивающим объектом, так что реализация услуг, их поддержка и доступ к ним возможны через посредство всех языков и услуг, предоставляемых
- 30 платформой MICROSOFT®'s.NET, а также в других вычислительных структурах.

Примеры вариантов осуществления

- Когда сформировано более одной версии электронного файла, в соответствии с изобретением можно конфигурировать программный интерфейс приложения так, чтобы обеспечить управление процессом синхронизации множества версий посредством
- 35 приложения программного обеспечения, связанного с файлом. В одном конкретном варианте осуществления приложение может инициировать синхронизацию автоматически, в то время как в другом приложение может инициировать синхронизацию на основе данных, вводимых пользователем. В другом конкретном варианте осуществления приложение предоставляет пользователю опции синхронизации, обуславливаемые
- 40 функциональными возможностями приложения, и позволяет этому пользователю выбирать опции и решать задачи с использованием функциональных возможностей приложения. В еще одном конкретном варианте осуществления первое приложение инициирует синхронизацию, тогда как второе приложение предоставляет опции синхронизации пользователю.

- 45 Обращаясь теперь к фиг. 2, следует отметить, что первый компьютер 210 содержит первый процессор 220 в оперативной связи с запоминающим устройством 218. Первый компьютер 210 может быть настольным персональным или портативным компьютером общего назначения, рабочей станцией, специализированным вычислительным устройством и/или подобным средством. Аналогично, первый процессор 220 может быть компьютерным
- 50 процессором общего назначения, специализированным процессором, сетью процессоров и т.п. С первым компьютером 210 оперативно связано отображающее устройство 230. Отображающее устройство 230 может быть любым средством предоставления опций и т.п. пользователю, например, таким как монитор, динамики и т.п. Такая оперативная связь

отображающего устройства 230 с первым компьютером 210 может быть реализована любыми проводными или беспроводными соединительными средствами. Хотя отображающее устройство 230 показано на фиг. 2 как находящееся внутри первого компьютера 210, отображающее устройство 230 может находиться в любом месте, где оно
 5 окажется в оперативной связи с первым процессором 220, включая удаленное место и т.п.

Запоминающее устройство 218 может представлять собой накопитель на жестких дисках, CD-ROM, магнитный диск, DVD, RAM и/или аналогичное средство. Подобно отображающему устройству 230, запоминающее устройство 218 может находиться в любом месте, где оно окажется в оперативной связи с первым процессором 220, включая
 10 удаленное место и т.п. Оперативные связи в такой ситуации могут быть проводными, беспроводными или их комбинацией. Внутри запоминающего устройства 218 содержится первая версия 212 электронного файла. Такая первая версия 212 может относиться к любому типу электронного файла, например, может быть документом, созданным с помощью программы подготовки и редактирования текстов или программы подготовки и
 15 редактирования электронных таблиц, базой данных и/или чем-либо подобным.

Первое приложение 208 и второе приложение 214 могут храниться в запоминающем устройстве 218. Первое приложение 208 и второе приложение 214 могут представлять собой исполняемые компьютером команды любого типа, например, такие как команды программы подготовки и редактирования текстов, электронных таблиц, базы данных,
 20 программируемого сценария (например, диалога «человек-машина»), специализированного программного обеспечения и т.п. В рассматриваемых ниже конкретных вариантах осуществления, где присутствуют и первое приложение 208, и второе приложение 214, как описано ниже, только второе приложение 214 должно быть совместимым с электронным файлом. В альтернативном варианте, который также будет рассмотрен ниже, первое
 25 приложение 208 и второе приложение 214 могут представлять собой одно и то же приложение. Одно из первого приложения 208 и второго приложения 214 может или они оба могут находиться вне запоминающего устройства 218, поскольку только одно из них должно находиться в оперативной связи с первым компьютером 210.

Программный интерфейс 216 приложения представляет собой исполняемые
 30 компьютером команды любого типа, предназначенные для проверки и сравнения двух или более копий электронного файла с целью синхронизации, и может быть автономным программным обеспечением, которое специально разработано для синхронизации двух или более копий электронного документа, объектом программного обеспечения, который работает как компонент другого приложения, интегральным компонентом первого или
 35 второго приложения 208, 214 и/или чем-либо подобным. Программный интерфейс 216 приложения может также решать дополнительные задачи, которые могут быть связаны или не связаны с синхронизацией. Первая версия 212, первое приложение 208, второе приложение 214 и программный интерфейс 216 приложения могут храниться в одном и том же запоминающем устройстве или в или на любом количестве разных запоминающих
 40 устройств 218.

Второй компьютер 222 содержит второй процессор 224 и запоминающее устройство 228, которое содержит вторую версию 226 электронного файла, упоминавшегося выше в связи с первой версией 212. Аналогично сказанному выше запоминающее устройство 228 может находиться в любом месте, где оно окажется в оперативной связи со вторым компьютером
 45 222, включая удаленное место и т.п. В конкретном варианте осуществления запоминающее устройство 228 и запоминающее устройство 218 также могут представлять собой одно и то же запоминающее устройство. С первым и вторым компьютерами 210, 222 оперативно связана сеть 219. Сеть 219 может быть любой сетью связи, предназначенной для реализации таких оперативных связей, например, такой как LAN, WAN, сеть
 50 беспроводной связи (радиосетью) и т.п. Как можно заметить, оперативно связанным с сетью 219 может быть любое количество первых компьютеров 210 и вторых компьютеров 222.

Из фиг. 3 видно, что процесс синхронизации начинается на этапе 311. В одном

конкретном варианте осуществления первое приложение 208 инициирует процесс синхронизации, а второе приложение 214 осуществляет этот процесс синхронизации. В другом конкретном варианте осуществления первое приложение 208 и второе приложения 214 представляют собой единственное приложение, которое и инициирует, и осуществляет синхронизацию. Хотя в ходе нижеследующего описания фиг. 3 описывается процесс, предусматривающий вовлечение в него и первого приложения 208, и второго приложения 214, следует признать, что все этапы, выполняемые первым приложением 208 и вторым приложением 214, может выполнять одно-единственное приложение.

Инициирование процесса синхронизации первым приложением 208 может быть вызвано любым количеством ситуаций, таких как открытие, закрытие, сохранение и т.п. первой версии 212; инициирование после распознавания первой и второй версий электронного файла; ручное инициирование по запросу пользователя; инициирование через фиксированные временные интервалы; инициирование после внесения записей пользователя в систему и/или подобные события. Кроме того, первое приложение 208 может накладывать дополнительные условия или может использовать заменяющие условия, например, такие, как инициирование синхронизации только тогда, когда имеется определенное количество версий. Более подробное рассмотрение инициирования события синхронизации в одном конкретном варианте осуществления последует ниже со ссылками на фиг. 4. Чтобы инициировать процесс, первое приложение 208 заставляет программный интерфейс 216 приложения осуществлять доступ к первой версии 212 и второй версии 226. Операцию доступа к первой версии 212 и второй версии 226 можно осуществлять с помощью любых вычислительных средств, например, путем просмотра информации, связанной с каждой версией, нахождения каждой версии путем использования функциональных возможностей поиска, локализации каждой версии путем использования идентифицирующей информации, предоставляемой первым приложением 208, и т.п.

Первое приложение 208 может снабжать интерфейс 216 любым количеством идентифицирующей информации о второй версии 226, например, такой как название файла, его местоположение, адрес и/или сведения, подобные этим. Интерфейс 216 может быть приспособлен для активного поиска второй версии 226 или может иметь ограниченные функциональные возможности, которые позволяют лишь поиск в месте, указываемом первым приложением 208, первой версией 212, и т.п. Кроме того, конфигурация интерфейса 216 может обеспечивать решение любого количества дополнительных задач, которые могут оказаться даже не связанными с процессом синхронизации. На этапе 313 программный интерфейс 216 приложения ищет вторую версию 226 и проверяет состояние синхронизации. Программный интерфейс 216 приложения определяет это состояние, сравнивая информацию, например, такую как время и/или дата последнего обновления и т.п., для первой версии 212 относительно второй версии 226. Состояние синхронизации можно также определять по критериям, дополняющим или заменяющим вышеупомянутую информацию. Если эта информация совпадает с информацией и/или критериями, указываемыми первым приложением 208, то первая и вторая версии 212, 226 имеют синхронизированное состояние. Если эта информация не совпадает с указанной, то первая и вторая версии 212, 226 не имеют синхронизированное состояние, а если так, то несовпадение свидетельствует о возможности того, что в одну из таких версий 212, 226 внесены изменения. В альтернативном варианте, если программный интерфейс 216 приложения обнаруживает ошибку, неудовлетворенное условие, условие, охарактеризованное первым приложением 208 в качестве причины окончания синхронизации, и т.п., то интерфейс 216 переходит к этапу 330. На этапе 330 интерфейс 216 информирует первое приложение 208 об упомянутом условии, а первое приложение 208 осуществляет соответствующее действие, которое может предусматривать любой тип действия или бездействия.

На этапе 315 программный интерфейс 216 приложения может воплотить один из двух образов действия в зависимости от исхода определения синхронизации согласно этапу 313. Если программный интерфейс 216 приложения определил, что первая версия 212 и

вторая версия 226 синхронизированы, то интерфейс 216 переходит к этапу 325. На этапе 325 программный интерфейс приложения сообщает первому приложению 208, что первая версия 212 и вторая версия 226 синхронизированы. Такое сообщение указывает первому приложению 208, что в данный момент дальнейшие операции синхронизации не требуются, поскольку первая версия 212 представляет собой самую последнюю версию электронного файла. Это сообщение также может указывать, что пользователь может использовать первое приложение 208 или второе приложение 214 для изменения первой версии 212.

На этапе 323 процесс синхронизации заканчивается. На этом этапе первое приложение 208 может прекращать функционирование, например, если такой процесс синхронизации имеет место, когда первое приложение 208 закрывает первую версию 212 электронного документа. В этой ситуации, так как синхронизация произошла, поэтому первое приложение может больше не иметь функций, которые нужно выполнить, и может быть завершено пользователем вручную или автоматически. В другом конкретном варианте осуществления первое приложение 208 инициирует процесс синхронизации при открытии первой версии 212. В такой ситуации приложение не должно завершаться по окончании такого события синхронизации, а должно позволять пользователю начинать изменение первой версии 212 с помощью функциональных возможностей первого приложения 208, если таковые имеются.

Однако если на этапе 315 программный интерфейс 216 приложения определил, что первая версия 212 и вторая версия 226 не синхронизированы, то работа интерфейса 216 выполняется в соответствии с этапом 317. На этапе 317 программный интерфейс 216 приложения сообщает первому приложению 208 и/или второму приложению 214, что первая версия 212 и вторая версия 226 не синхронизированы. Такое сообщение заставляет второе приложение 214 начать синхронизацию первой версии 212 и второй версии 226 с помощью данных, вводимых пользователем, что рассмотрено ниже со ссылками на этапы 319 и 321.

На этапе 319 второе приложение 214 предоставляет пользовательский интерфейс пользователю путем отображения пользовательского интерфейса на отображающем устройстве 230. Такой пользовательский интерфейс может быть представлен в форме текстового и/или графического окна, звукового приглашения и/или чего-либо подобного, и любая такая конфигурация пользовательского интерфейса одинаково совместима с конкретным вариантом осуществления настоящего изобретения. Предоставляя пользователю такой пользовательский интерфейс, второе приложение 214 может предоставлять пользователю одну или более опций с использованием функциональных возможностей второго приложения 214 и т.п. Например, если второе приложение 214 является программой подготовки и редактирования текстов, то второе приложение может предоставлять пользователю опцию синхронизации в рамках пользовательской среды подготовки и редактирования текста второго приложения 214, например, такой, как окно редактирования с возможностью просмотра страниц такой первой версии 212, и/или аналогичную опцию. Кроме того, второе приложение 214 может подчеркивать различия между первой версией 212 и второй версией 226, используя цвет, звуки, графику и т.п. для предоставления пользователю лучших возможностей принятия решений о синхронизации. Второе приложение 214 также может предоставлять дополнительную информацию, например, такую как идентификационная информация каждого пользователя, изменявшего любую из версий 212, 226, и может разрешать пользователю изменять только версию 212, 226, если такой пользователь имеет соответствующее разрешение - в форме категории допуска и т.п. - на внесение этого изменения. Второе приложение может позволять пользователю, например, объединять версии 212, 216 с получением третьей копии, которая будет включать в себя изменения, вносимые в обе версии 212, 226. Дополнительная опция также может предоставляться в форме руководящего указания, вытекающего из функциональных возможностей второго приложения 214, например, такого как указание об образе действий и/или указание, подобное этому. Как можно заметить, опция может представлять собой что-либо, генерируемое вторым приложением 214 с

использованием функциональных возможностей второго приложения 214 в отношении первой версии 212 и второй версии 226.

На этапе 321 второе приложение 214 принимает и обрабатывает данные, вводимые пользователем. Такие данные, вводимые пользователем, могут содержать любое количество информации, касающейся такой синхронизации. Например, данные, вводимые пользователем, могут обуславливать выбор конкретного изменения либо первой версии 212, либо второй версии 226 электронного файла, которое пользователь хочет сохранить. В другом конкретном варианте осуществления данные, вводимые пользователем, могут обуславливать выбор либо первой версии 212, либо второй версии 226 электронного файла, которую пользователь хочет сохранить во всей ее полноте. В еще одном конкретном варианте осуществления данные, вводимые пользователем, могут содержать редакционные изменения первой и/или второй версии 212, 216 с использованием функциональных возможностей второго приложения 214, которое по его завершении приводит к синхронизации первой версии 212 и второй версии 226. Можно понять, что данные, вводимые пользователем, могут представлять собой любые вводимые данные, которые второе приложение 214 делает доступными для выбора или использования пользователем, и любые такие вводимые данные находятся в рамках конкретного варианта осуществления.

Также очевидно, что этапы 319 и 321 могут повторяться, поскольку такой пользовательский интерфейс может быть неоднократно отображен и/или обновлен повторно, в зависимости от данных, вводимых пользователем. Например, пользовательский интерфейс может быть переконфигурирован вторым приложением 214 на основании данных, вводимых пользователем. Такое переконфигурирование может происходить, если, например, пользователь выбирает одно из множества различий между первой версией 212 и второй версией 226, а второе приложение 214 повторно отображает пользовательский интерфейс для пользователя, чтобы отразить такой выбор. Этапы 319 и 321 могут повторяться любое количество раз или не повторяться вовсе, и любая ситуация из этих двух равнозначно находится в рамках конкретного варианта осуществления настоящего изобретения. Кроме того, второе приложение 214 может сообщать результаты синхронизации первому приложению 208 или интерфейсу 216. Результаты, сообщаемые вторым приложением 214, могут быть любой информацией, связанной с синхронизацией, такой как вариант (варианты) выбора опций, результат (результаты) синхронизации и т.п.

Второе приложение 214 может переходить к этапу 323 окончания синхронизации, когда происходит событие переключения. Такое событие - переключение - может произойти в любой момент, наступающий, например, тогда, когда пользователь завершил повторное согласование всех различий между первой версией 212 и второй версией 226, когда пользователь уведомляет второе приложение 214 об отказе от процесса синхронизации или о его окончании без завершения синхронизации, в некоторый предварительно определенный момент, и т.п. На этапе 323 синхронизация оканчивается. После этого, как упоминалось выше в связи с этапом 325, первое приложение 208 или второе приложение 214 может предпринять на этапе 323 любое действие, соответствующее контексту, в котором имеет место синхронизация.

На фиг. 4 более подробно показан этап 311 в соответствии с одним конкретным вариантом осуществления изобретения. Как говорилось выше в связи с фиг. 3, на этапе 311 первое приложение 208 инициирует процесс синхронизации. Фиг. 4 подробнее иллюстрирует такой процесс синхронизации после открытия первой версии 212 электронного файла. Как сказано выше, первое приложение 208 может инициировать такой процесс синхронизации в любой момент, например, такой, который наступает при открытии, сохранении или закрытии первой версии 212, при задании параметров или изменяемых интервалов, после получения данных, вводимых пользователем, и т.п.

На этапе 411 первое приложение 208 открывает первую версию 212 электронного файла. Такое открытие будет обычно происходить после ввода пользователем данных с целью открытия такой первой версии 212, но в рамках конкретного варианта

осуществления настоящего изобретения находятся любые средства для инициирования такого открытия первой версии 212. На этапе 413 первое приложение 208 определяет, имеет ли первая версия 212 связанную с ней вторую версию 226. Первое приложение 208 может перейти к этапу 413 после полного открытия первой версии 212 на этапе 411 или
5 может перейти к этапу 413 в любой момент в течение такого процесса открытия, включая, например, момент перед открытием первой версии 212 на этапе 411. Кроме того, для обеспечения осуществления такого определения первым приложением 208 можно применить и другой способ. Например, с этой первой версией 212 может быть связана такая информация, как электронный ярлык и т.п., разрешающая первому приложению 208
10 осуществлять такое определение. В отличие от этого для такого определения первое приложение 208 может осуществлять поиск любых или всех вычислительных устройств, оперативно подсоединенных к сети 218, например, таких как второй компьютер 222. Как можно понять, первое приложение 208 может использовать любой способ определения наличия второй версии 226, который при этом находится в рамках конкретного варианта осуществления изобретения.
15

Если на этапе 413 первое приложение 208 не обнаруживает вторую версию 226, то первое приложение 208 определяет, что вторая версия 212 является лишь копией такого электронного файла, и переходит к этапу 417. На этапе 417 первое приложение 208 обрабатывает первую версию 212 обычным образом. Такая обычная обработка может
20 представлять собой обработку любого типа, которой первое приложение 208 в типичном случае подвергает электронные файлы такого же формата, как у первой версии 212, возможно, в соответствии с какими-либо данными, вводимыми пользователем. Например, на фиг. 4 показано, что первое приложение 208 находилось в процессе открытия первой версии 212, когда первое приложение 208 инициировало такой процесс синхронизации в соответствии с этапом 311. Не обнаружив потребности в синхронизации - ввиду
25 отсутствия второй версии 226, - первое приложение 208 завершает открытие первой версии 212 и переходит к соответствующему этапу.

Если на этапе 413 первое приложение 208 обнаруживает вторую версию 226, то первое приложение 208 переходит к этапу 415. На этапе 415 первое приложение 208 вступает в
30 контакт с программным интерфейсом 216 приложения, чтобы начать синхронизацию первой версии 212 и второй версии 226. После завершения контакта с программным интерфейсом 216 приложения такой процесс синхронизации переходит к этапу 313, как сказано выше в связи с фиг. 3.

Таким образом, представлены способы и устройства для управления репликацией
35 файлов в приложениях. Хотя настоящее изобретение описано в связи с предпочтительными конкретными вариантами его осуществления, представленными на различных чертежах, следует понять, что можно использовать другие аналогичные конкретные варианты осуществления и вносить дополнения в описанный конкретный вариант осуществления для выполнения той же самой функции настоящего изобретения в рамках его притязаний. Например, специалист в данной области техники поймет, что
40 настоящее изобретение в том виде, как оно описано в данной заявке, применимо к любому вычислительному устройству или любой вычислительной среде, проводной или беспроводной, и применимо к любому количеству таких вычислительных устройств, соединенных сетью связи и взаимодействующих через эту сеть. Кроме того, следует
45 подчеркнуть, что предусматривается возможность применения на множестве компьютерных платформ, включая операционные системы карманных устройств и другие операционные системы прикладной ориентации, и это особенно важно в связи с продолжающимся быстрым увеличением количества устройств в беспроводных сетях. Помимо этого, настоящее изобретение может быть реализовано во или на множестве
50 процессорных микросхем или устройств, и точно также на множестве устройств возможно простое осуществление хранения (запоминания). Поэтому не следует считать настоящее изобретение сводящимся к какому-либо одному конкретному варианту осуществления, а нужно рассматривать изобретение в рамках существа и объема притязаний в соответствии

с прилагаемой формулой изобретения.

Формула изобретения

1. Реализуемый компьютером и выполняющийся на компьютерной системе, содержащей
 5 процессор и память, способ управления первой и второй версиями электронного файла, заключающийся в том, что
 инициируют с использованием первого приложения программного обеспечения, выполняющегося на упомянутом процессоре, событие синхронизации для синхронизации первой версии со второй версией;
 10 заставляют посредством первого приложения программного обеспечения программный интерфейс приложения компьютерной системы осуществлять доступ к упомянутым первой и второй версиям;
 сравнивают посредством программного интерфейса приложения информацию, связанную с каждой из упомянутых первой и второй версий, и генерируют результат;
 15 сообщают посредством программного интерфейса приложения результат сравнения первому приложению программного обеспечения;
 заставляют посредством программного интерфейса приложения в ответ на упомянутый результат второе приложение программного обеспечения, выполняющееся на упомянутом процессоре, генерировать пользовательский интерфейс в соответствии с признаком
 20 второго приложения программного обеспечения, при этом сгенерированный пользовательский интерфейс представляют пользователю во втором приложении программного обеспечения, причем электронный файл связан со вторым приложением программного обеспечения;
 открывают первую и вторую версии электронного файла посредством второго
 25 приложения программного обеспечения и представляют такие открытые первую и вторую версии пользователю;
 представляют пользователю посредством второго приложения программного обеспечения опцию синхронизации в отношении представленных версий с использованием сгенерированного пользовательского интерфейса второго приложения программного
 30 обеспечения, при этом опция синхронизации предоставляется вторым приложением программного обеспечения;
 принимают от пользователя посредством второго приложения программного обеспечения входные данные через пользовательский интерфейс;
 автоматически синхронизируют посредством второго приложения программного
 35 обеспечения первую и вторую версии согласно пользовательским входным данным, принятым через пользовательский интерфейс; и
 при этом автоматическая синхронизация первой и второй версий содержит слияние первой и второй версий в третью версию согласно входным данным, принятым от
 пользовательского интерфейса, в то же время оставаясь в пользовательском интерфейсе
 40 второго приложения программного обеспечения, и перезапись первой и второй версий третьей версией.
2. Способ по п.1, при котором первое и второе приложения программного обеспечения представляют собой одно и то же приложение.
3. Способ по п.1, при котором результат упомянутого этапа сравнения представляет
 45 собой состояние синхронизации первой и второй версий.
4. Способ по п.1, при котором первая версия хранится на первом компьютере, а вторая версия хранится на втором компьютере, причем упомянутые первый и второй компьютеры оперативно соединены сетью связи.
5. Способ по п.1, в котором третья версия электронного файла содержит часть первой
 50 версии электронного файла и часть второй версии электронного файла.
6. Способ по п.1, в котором упомянутый этап предоставления опции синхронизации дополнительно предусматривает отображение первой и второй копий в пользовательском интерфейсе.

7. Способ по п.1, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения обнаруживает информацию, связанную с первой версией электронного файла, и указывающую о наличии второй версии электронного файла.

5 8. Способ по п.1, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения открывает первую версию электронного файла.

9. Способ по п.1, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения сохраняет
10 первую версию электронного файла в запоминающем устройстве компьютерной системы.

10. Способ по п.1, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения распознает первую версию и вторую версию.

11. Считываемый компьютером носитель, имеющий исполняемые компьютером команды
15 для осуществления способа управления первой и второй версиями электронного файла, при этом способ заключается в том, что

инициируют с использованием первого приложения программного обеспечения событие синхронизации для синхронизации первой версии на первом компьютере со второй
версией;

20 заставляют посредством первого приложения программного обеспечения программный интерфейс приложения компьютерной системы осуществлять доступ к упомянутым первой и второй версиям;

сравнивают посредством программного интерфейса приложения информацию, связанную с каждой из упомянутых первой и второй версий, и генерируют результат;

25 сообщают посредством программного интерфейса приложения результат сравнения первому приложению программного обеспечения;

заставляют посредством программного интерфейса приложения в ответ на упомянутый результат второе приложение программного обеспечения генерировать пользовательский интерфейс в соответствии с признаком второго приложения программного обеспечения,

30 при этом сгенерированный пользовательский интерфейс представляют пользователю второго приложения программного обеспечения, причем электронный файл связан со вторым приложением программного обеспечения;

открывают первую и вторую версии электронного файла посредством второго приложения программного обеспечения и представляют такие открытые первую и вторую
35 версии пользователю;

представляют пользователю посредством второго приложения программного обеспечения опцию синхронизации в отношении представленных версий с использованием сгенерированного пользовательского интерфейса второго приложения программного обеспечения, при этом опция синхронизации предоставляется вторым приложением
40 программного обеспечения;

принимают от пользователя посредством второго приложения программного обеспечения входные данные через пользовательский интерфейс;

автоматически синхронизируют посредством второго приложения программного обеспечения первую и вторую версии согласно пользовательским входным данным,
45 принятым через пользовательский интерфейс; и

при этом автоматическая синхронизация первой и второй версий содержит слияние первой и второй версий в третью версию согласно входным данным, принятым от пользовательского интерфейса, в то же время оставаясь в пользовательском интерфейсе второго приложения программного обеспечения, и перезапись первой и второй версий
50 третьей версией.

12. Считываемый компьютером носитель по п.11, в котором первое и второе приложения программного обеспечения представляют собой одно и то же приложение.

13. Считываемый компьютером носитель по п.11, в котором результат упомянутого

этапа сравнения представляет собой состояние синхронизации первой и второй версий.

14. Считываемый компьютером носитель по п.11, в котором первая версия хранится на первом компьютере, вторая версия хранится на втором компьютере, а упомянутые первый и второй компьютеры оперативно соединены посредством сети связи.

5 15. Считываемый компьютером носитель по п.11, в котором упомянутая третья версия содержит часть второй версии электронного файла.

16. Считываемый компьютером носитель по п.11, в котором упомянутый этап предоставления опции синхронизации дополнительно предусматривает отображение первой и второй копий в пользовательском интерфейсе.

10 17. Считываемый компьютером носитель по п.11, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения обнаруживает информацию, связанную с первой версией электронного файла, указывающую о наличии второй версии электронного файла.

15 18. Считываемый компьютером носитель по п.11, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения открывает первую версию электронного файла.

20 19. Считываемый компьютером носитель по п.11, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения сохраняет первую версию электронного файла в запоминающее устройство первого компьютера.

20. Считываемый компьютером носитель по п.11, в котором упомянутый этап инициирования события синхронизации имеет место после того, как первое приложение программного обеспечения распознает первую версию и вторую версию.

25

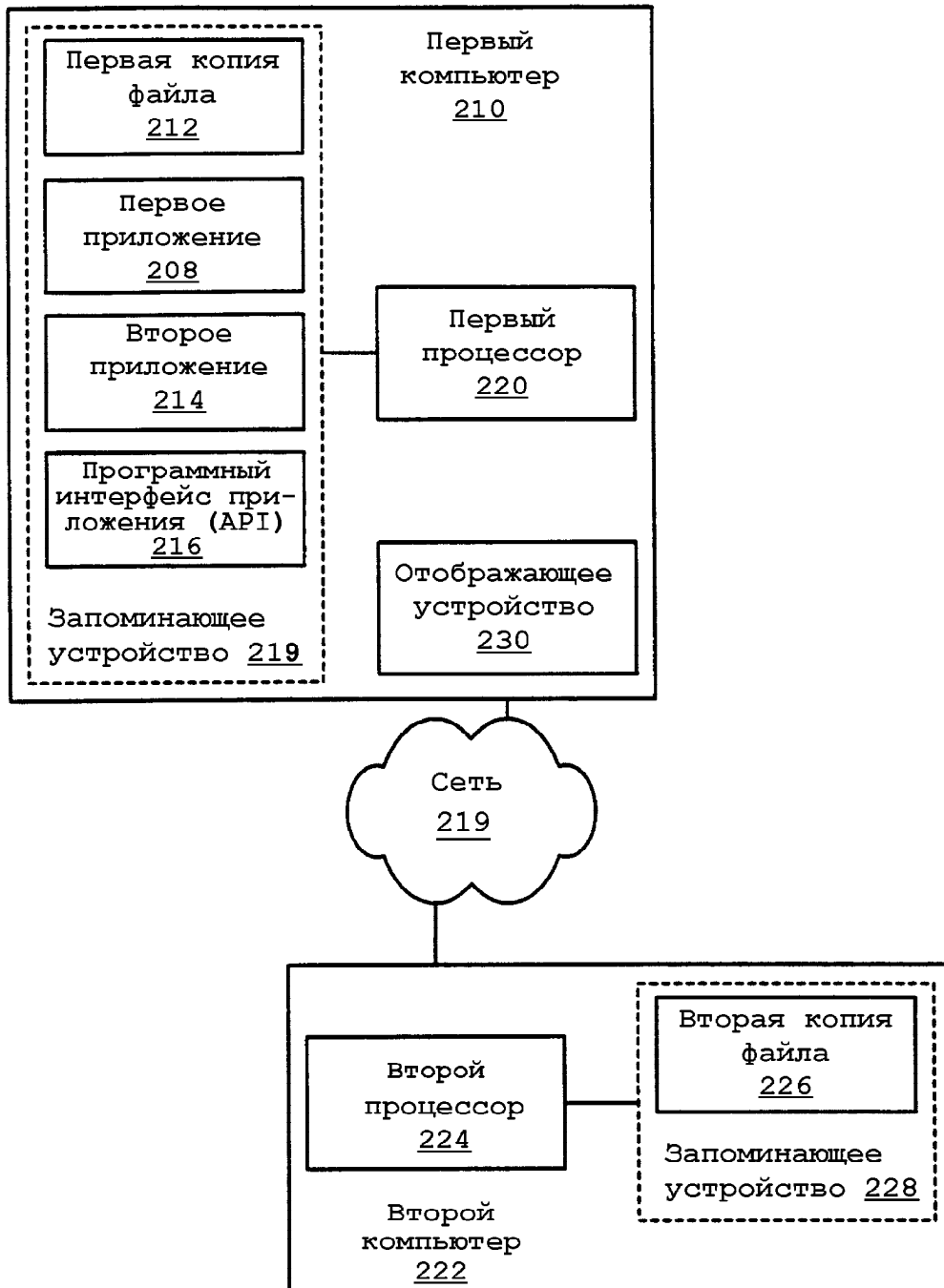
30

35

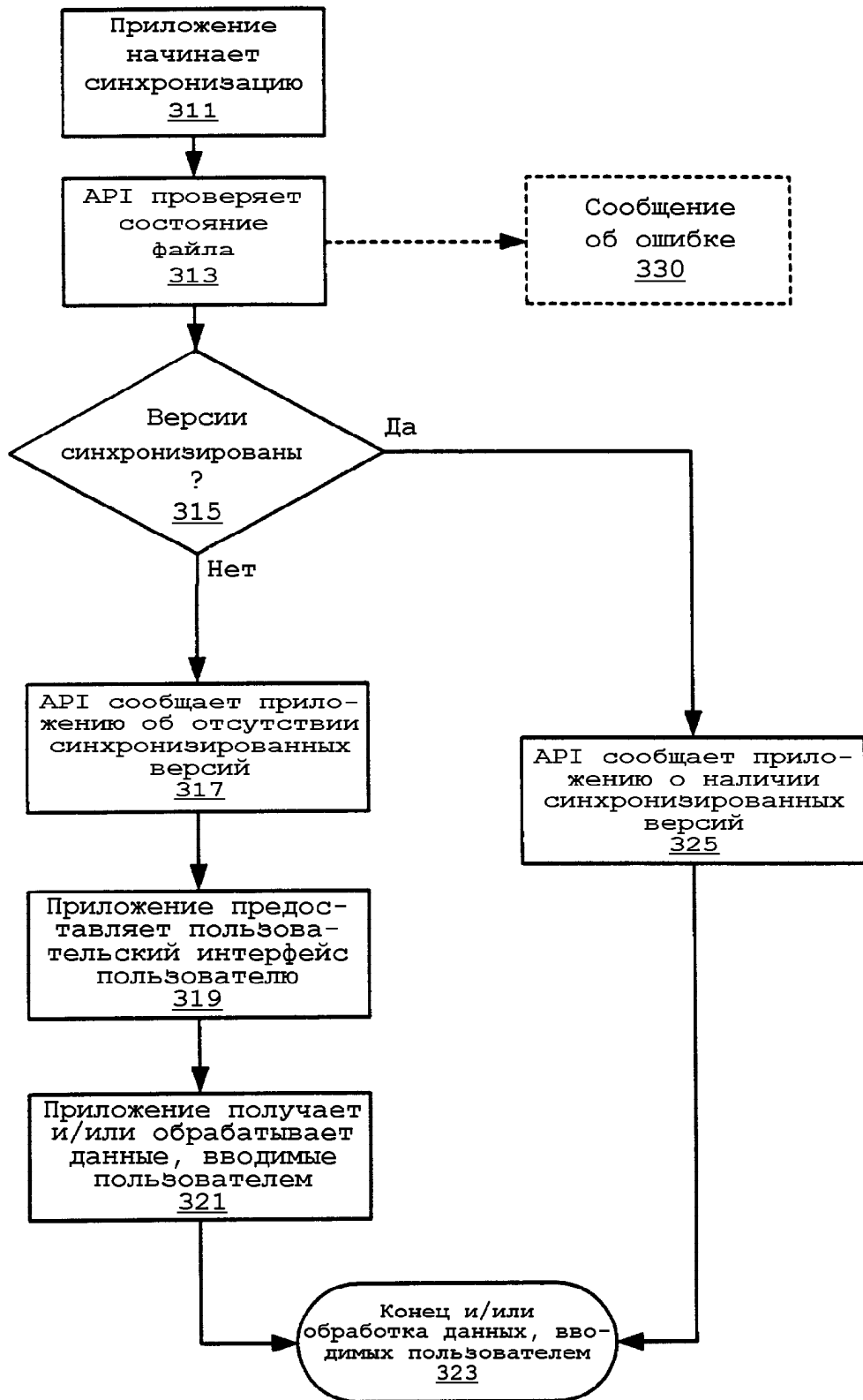
40

45

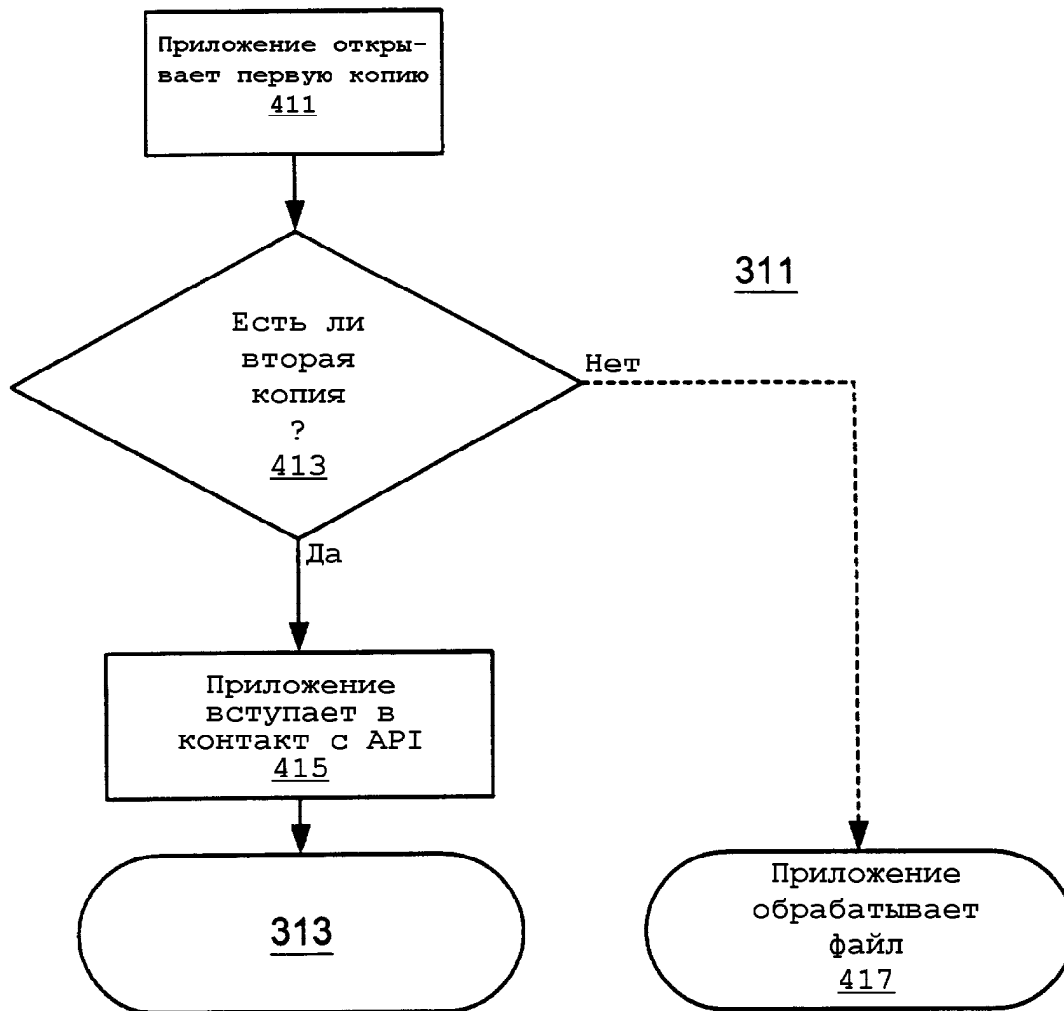
50



Фиг. 2



ФИГ. 3



Фиг. 4