



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: 2004131856/09, 26.03.2003

(24) Дата начала отсчета срока действия патента:
26.03.2003

(30) Конвенционный приоритет:
02.04.2002 US 10/115,268

(43) Дата публикации заявки: 10.05.2005

(45) Опубликовано: 27.10.2008 Бюл. № 30

(56) Список документов, цитированных в отчете о
поиске: US 6018762 A1, 25.01.2000. RU 2157596
C2, 10.10.2000. WO 1997/24678 A1, 10.07.1997.
US 5647002 A, 08.07.1997. US 5948059 A,
07.09.1999. US 5592664 A, 07.01.1997.

(85) Дата перевода заявки РСТ на национальную фазу:
02.11.2004

(86) Заявка РСТ:
IB 03/01097 (26.03.2003)

(87) Публикация РСТ:
WO 03/083684 (09.10.2003)

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):

ПИИСПАНЕН Юсси (FI),
САХИНОЯ Микко (FI)

(73) Патентообладатель(и):

НОКИА КОРПОРЕЙШН (FI)

(54) СПОСОБ И УСТРОЙСТВО ДЛЯ синхронизации того, как данные сохраняются в
различных хранилищах данных

(57) Реферат:

Изобретение относится к передаче данных, в
которых синхронизируется структура данных в
хранилищах данных, содержащих элементы
данных из одного или более приложений, таких как
календари или электронная почта настольных
компьютеров. Технический результат - повышение
точности синхронизации. Для этого в способе,
включающем в себя SyncML или иной сходный
язык разметки, посредством чего два устройства
синхронизируют то, как организуются данные в
соответствующих хранилищах данных,
поддерживаемых этими устройствами, которые
работают согласно способу, а также

соответствующей компьютерной программе,
посредством которой любое из упомянутых
устройств работает согласно этому способу.
Согласно способу два устройства синхронизируют
свои соответствующие хранилища данных в
отношении папок для содержания блоков данных и,
возможно, других папок при помощи обмена
сообщениями, которые включают в себя элементы
идентификации данных, которые ссылаются на
папки, чтобы было возможно синхронизировать
хранилища данных в отношении папок, причем
элементы идентификации данных предусмотрены в
сообщении внешними для любых ссылок на блоки
данных. 4 н. и 20 з.п. ф-лы, 4 ил.





FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: 2004131856/09, 26.03.2003

(24) Effective date for property rights: 26.03.2003

(30) Priority:
02.04.2002 US 10/115,268

(43) Application published: 10.05.2005

(45) Date of publication: 27.10.2008 Bull. 30

(85) Commencement of national phase: 02.11.2004

(86) PCT application:
IB 03/01097 (26.03.2003)

(87) PCT publication:
WO 03/083684 (09.10.2003)

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

(72) Inventor(s):

PIISPANEN Jussi (FI),
SAKHINOJa Mikko (FI)

(73) Proprietor(s):

NOKIA KORPOREJShN (FI)

(54) METHOD AND DEVICE FOR DATA STORAGE SYNCHRONISATION IN DIFFERENT DATA STORAGES

(57) Abstract:

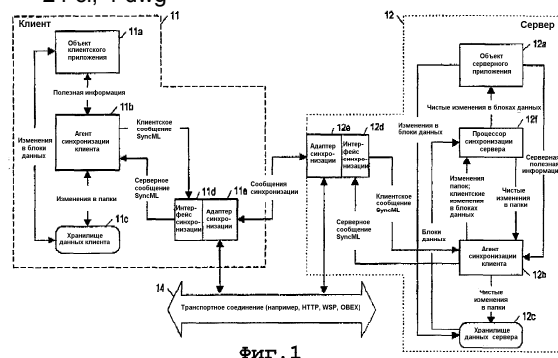
FIELD: physics, computation equipment.

SUBSTANCE: method involves SyncML or any similar tag language allowing two devices to synchronise data arrangement in respective data storages supported by these devices, functioning according to the method and respective computer program, which allows any of the mentioned devices to function according to the method. According to the method, two devices synchronise their respective data storages in respect of data block directories and possibly other directories via exchange of messages containing data identification elements referring to directories to enable data storage synchronisation in respect

of directories. Data identification elements in messages are external to any data block references.

EFFECT: enhanced synchronisation accuracy.

24 cl, 4 dwg



Область, к которой относится изобретение

Настоящее изобретение относится к тому, каким образом синхронизируются перспективные хранилища данных, содержащие элементы данных из одного или более приложений, таких как календари или электронная почта настольных компьютеров.

5 Конкретнее, настоящее изобретение касается того, как синхронизируется структура данных в таких хранилищах данных.

Уровень техники

В настоящее время деловому человеку можно использовать приложение электронной почты и приложение календаря (для отслеживания встреч) в мобильном телефоне или
10 миниатюрном (ручном) компьютере (или каком-либо ином мобильном терминале) и иметь помощника, использующего то же самое приложение в настольном компьютере, чтобы отслеживать электронную почту, посылаемую и принимаемую этим помощником от имени делового человека, и отслеживать встречи для этого делового человека. В такой компоновке часто случается, что группы данных (называемые также *блоками данных*) для
15 приложения электронной почты и группы данных для календаря содержатся в хранилище данных в мобильном телефоне и в другом хранилище данных в настольном компьютере. Таким образом, когда делаются изменения в любом из этих хранилищ данных, каждое из которых включает в себя группы данных как для приложения электронной почты, так и для приложения календаря, эти хранилища данных должны синхронизироваться. При
20 синхронизации соответствующих хранилищ данных (т.е. синхронизации как групп данных, так и структуры данных двух этих хранилищ данных), используемых двумя приложениями, исполняемыми на различном оборудовании, содержания этих двух хранилищ данных устанавливаются так, чтобы они соответствовали друг другу, на основании протокола, по которому сообщаются изменения с момента последней синхронизации, разрешаются
25 конфликты между изменениями, сделанными на обеих частях этого оборудования (согласно некоторой политике), и вносятся изменения или в элементы данных, или в их организацию, либо и в то и в другое в одном или обоих хранилищах данных.

Для синхронизации таких хранилищ данных разрабатываются средства на основании того, что называется SyncML (synchronization markup language) (язык разметки
30 синхронизации), разрабатываемый по так называемой Инициативе SyncML (информацию о SyncML, в том числе стандарты и описания для SyncML, и об Инициативе SyncML, специально включающей в себя Протокол представления SyncML и Протокол синхронизации SyncML, см. [http:// www.syncml.org/](http://www.syncml.org/)). SyncML представляет собой открытый промышленный стандарт для общего языка для универсальной синхронизации
35 удаленных данных (т.е. групп данных, хранящихся в различном оборудовании и, следовательно, в различных хранилищах данных) и личной информации по множеству сетей, платформ и устройств. С помощью SyncML группы данных - но все же не структура данных - могут быть синхронизированы на различных устройствах, соединенных через одну или более взаимосвязанных сетей, включающих, к примеру, универсальную систему
40 мобильной связи (УСМС) (UMTS), сеть радиодоступа (СРД) (UTRAN) и Интернет, где связь может быть беспроводной в целом или частично, либо проводной. Таким образом, устройства могут осуществлять связь через, например, фиксированные сети (в том числе беспроводные сети), инфракрасные, кабельные или Bluetooth.

Помимо использования для синхронизации хранилищ данных, SyncML (т.е. язык) можно
45 использовать для управления устройствами и более точно, для переноса управляющих действий между клиентом и управляющим сервером для клиента. Описание Протокола управления устройствами SyncML см. <http://www.syncml.org>. Протокол управления устройствами SyncML позволяет управляющим командам выполняться в управляемых объектах, и он использует формат пакетов, аналогичный Протоколу синхронизации SyncML
50 и Протоколу представления SyncML. Управляемый объект может отражать набор параметров конфигурации для устройства. Действия, которые можно предпринимать по отношению к этому объекту, могут включать в себя считывание и установку ключей и значений параметров. Другим управляемым объектом может быть среда выполнения

реального времени для программных приложений в устройстве. Действия, которые можно предпринимать в отношении этого типа объекта, включают в себя установку, обновление или деинсталляцию программных элементов. Действия представляются командами Протокола управления устройствами SyncML, которые описываются в

5 Протоколе представления SyncML, Пользование управлением устройствами. Используемые команды и структура сообщений идентичны тем, что содержатся в Протоколе синхронизации SyncML. (Таким образом, так называемое определение типа документа для Протокола управления представляет собой определение типа документа из Протокола синхронизации SyncML).

10 До настоящего времени развитие синхронизации мобильной передачи данных в наибольшей степени определялось ограничениями. Синхронизация хранилищ данных основана на наборе различных проприетарных протоколов, и функционирующих каждый с очень ограниченным числом устройств, систем и типов данных. Эти не взаимодействующие друг с другом технологии усложняют задачи пользователей, производителей, поставщиков

15 услуг и разработчиков. Далее, расширение числа различных проприетарных протоколов синхронизации хранилищ данных является преградой для широкого использования мобильных устройств, ограничивает доступ к данным и доставку данных и ограничивает мобильность пользователей.

По мере роста популярности мобильных вычислительных и связанных устройств

20 пользователи хотят обращаться к обновляемой информации и приложениям, где бы они ни находились и какие бы устройства ни использовали, подталкивая к необходимости открытого стандарта, такого как стандарт SyncML.

SyncML использует синтаксис так называемого расширяемого языка разметки (XML), как установлено в документе Extensible Markup Language (XML) 1.0, продукт учреждения XML

25 в консорциуме World Wide Web Consortium (W3C), имеющего вебсайт по адресу <http://www.w3.org>. За информацией о XML следует обращаться по адресу <http://www.w3.org/XML>.

Как используется в настоящем описании, термин *группа данных* или *блок данных* указывает блок данных, которые организованы в то, что в настоящем описании

30 называется *папка*, что в свою очередь составляет то, что в настоящем описании называется структура директории, также называемая в настоящем описании структура данных. На фиг.4 структура директории показана включающей в себя различные папки, каждая из которых включает в себя один или более блоков данных.

Выражение структура директории или структура данных и выражение папка следует

35 понимать широко. Во-первых, папку следует понимать как любое вместилище блоков данных. Таким образом, к примеру, то, что называется папка в разных операционных системах (таких как Windows, доступная от Microsoft Corporation), нужно понимать под термином папка, как он используется в настоящем описании, но это и запись в таблице реляционной базы данных, поскольку запись в такой таблице включает в себя поля,

40 которые являются блоками данных. Каждое такое поле можно рассматривать как папку, поскольку поле содержит знаки, числа или иные элементы, которые можно рассматривать как блоки данных, а тем самым каждое поле является папкой.

Выражение структура директории или структура данных указывает размещение папок в хранилище данных, как, например, древовидная структура папок в директории,

45 поддерживаемой операционной системой, и в этом случае блоки данных будут папками. Но следует также понимать, что выражение структура директории или структура данных охватывает структуры директории, поддерживаемые внутренне приложениями, т.е. не поддерживаемые операционной системой, но вместо этого поддерживаемые приложением, работающим под управлением операционной системы. Примером является структура

50 директории, поддерживаемая большинством приложений электронной почты, которые обычно всегда включают в себя по меньшей мере две папки: папку принятой электронной почты (поступило) и папку переданной электронной почты (отправлено). Кроме того, следует понимать также, что выражение структура директории или структура данных

охватывает внутренние структуры директорий, которые используются совместно различными приложениями (так что структура директории не является внутренней только для одного приложения, а для нескольких, обычно хорошо взаимосвязанных приложений, любое из которых может вносить в нее изменения). Далее, следует понимать, что

5 выражение структура директории или структура данных охватывает как структуры системной директории, так и структуры внутренней директории. В случае внутренней директории, поддерживаемой приложением электронной почты, блок данных будет электронной почтой, видом внутреннего файла. В структуре директории любого типа папка может включать в себя, в общем случае, блоки данных, которые принадлежат различным
10 приложениям, и приложение знает, принадлежит ли блок данных ему или другому приложению, либо на основании некоторого атрибута, поддерживаемого в связи с этим блоком данных, либо на основании приложения, хранящего закрытую для доступа таблицу всех блоков данных, которые принадлежат ему, где бы они ни были расположены. Также, поскольку, как уже пояснено, папка может указывать, к примеру, запись в таблице базы
15 данных или даже поле в такой записи, следует также понимать, что выражение структура директории или структура данных охватывает структуру базы данных (когда папка является записью) или структуру записи (когда папка является полем). Важно понимать, что выражение папка используется, таким образом, расширительно, чтобы указывать любое вместилище блоков данных, и выражение структура директории или структура
20 данных имеет соответствующее расширительное значение.

Пользователь приложения может вносить изменения как в блоки данных, которые принадлежат приложению (такие как добавление новых блоков данных или замена блока данных обновленной версией, потому что пользователь воспользовался приложением для
25 изменения содержания блока данных, и т.д.), так и в то, как сохраняются группы данных (такие как добавление новой папки и перемещение некоторых блоков данных из уже существующей папки в новую папку), т.е. как в блоки данных, так и в папки (т.е. вся структура директории). Однако, согласно существующему SyncML, если пользователь меняет папки на устройстве, SyncML не может использоваться для синхронизации хранилища данных в этом устройстве с соответствующим хранилищем данных,
30 поддерживаемым в другом устройстве; прототип SyncML позволяет лишь синхронизацию в отношении изменений блоков данных.

Сообщение SyncML представляет собой вложенную структуру, и одно или более сообщений SyncML могут связываться с тем, что называется пакет SyncML. Сообщение SyncML является отдельным документом XML, состоящим из одного или более типов
35 элементов. Этот документ состоит из заголовка, определяемого типом элемента ЗаголСинхр (SyncHdr), и тела, определяемого типом элемента ТелоСинхр (SyncBody). Заголовок SyncML конкретизирует информацию маршрутизации и контроля версий о сообщении SyncML. Тело SyncML представляет собой вместилище для одной или более команд SyncML. Команды SyncML конкретизируются отдельными типами элементов.
40 Команды SyncML действуют как вместилища для других типов элементов, которые описывают подробности команды SyncML, в том числе любые данные или мета-информацию.

SyncML определяет *команды запроса* и *команды ответа*. Команды запроса включают в себя, например, команды: добавить (команда, которая позволяет отправителю просить, чтобы один или более блоков данных были добавлены к данным, доступным для
45 получателя); уведомить (позволяющая отправителю известить получателя о состоянии); копировать (позволяющая отправителю просить, чтобы копировались один или более блоков данных, доступных для получателя); удалить (позволяющая отправителю просить, чтобы были удалены или архивированы один или более блоков данных, доступных для
50 получателя); получить (позволяющая отправителю просить один или более блоков данных от получателя) и искать (позволяющая отправителю просить, чтобы выполнялся введенный запрос относительно одного или более блоков данных, доступных для получателя). Единственными командами ответа в настоящее время являются: статус (указывающая

статус завершения операции или тот факт, что произошла ошибка при обработке предыдущего запроса) и результат (используемая для возвращения результатов данных на команду SyncML либо Получить, либо Искать).

SyncML использует идентификаторы, чтобы идентифицировать блоки данных или папки.

- 5 Идентификаторы включаются в то, что называется типами Исходных и Целевых элементов, и могут быть комбинациями, унифицированными идентификаторами ресурсов (УИР) (URIs), унифицированными названиями ресурсов (УНР) (URNs) и текстовыми названиями. (Чтобы указать международный идентификатор мобильного оборудования (МИМО) (IMEI), SyncML использует тип УНР МИМО. Это УНР МИМО определяет достоверный 15-разрядный
- 10 МИМО. Вдобавок, SyncML использует тип УНР SyncML, чтобы идентифицировать в SyncML конкретные пространства названий и уникальные названия. Иные типы УНР также могут использоваться в типе элемента МестнУИР (LocUPI)).

Как уже пояснялось, протокол представления SyncML (т.е. сообщение SyncML)

- представляет собой разметку документа, состоящую из типов элементов XML. Эти типы
- 15 элементов определяются в терминах своего назначения или использования, родительских элементов, любых ограничений на содержание или использование и контентной модели. Эти типы элементов включают в себя так называемые элементы общего пользования, элементы вместилища сообщений, элементы описания данных, элементы управления протоколом и элементы команд протокола.

- 20 Элементы общего пользования представляют собой типы элементов, используемые другими типами элементов SyncML, и включают в себя, например, «архивировать» для указания того, что данные, конкретизированные в команде «удалить», должны архивироваться получателем команды «удалить», а не просто удаляться. Тем самым команда «удалить» может использовать элемент общего пользования «архивировать», а
- 25 потому называется *родительским элементом* для типа элемента общего пользования «архивировать» в этом контексте. Другим типом команды общего пользования является тип элемента Кмд (Cmd), который используется для конкретизации команды SyncML, на которую ссылается тип элемента Статус (а потому тип элемента Статус является родительским элементом в этом контексте). Другим является тип элемента ИДКмд (CmdID),
- 30 который используется для конкретизации идентификатора уникальной для сообщения команды SyncML и может иметь разные родительские элементы, в том числе: Добавить, Уведомить, Элементарный, Копировать, Удалить, Исполнять, Получить, Отображать, Поместить, Заменить, Результаты, Искать, Последовательность, Статус и Синхронизировать.

- 35 В отношении изобретения особо отметим типы общих элементов МестнНазвание, МестнУИР, Исходный и Целевой. МестнНазвание используется для конкретизации названия дисплея для целевого или исходного адреса, а потому может иметь в качестве родительских элементов Целевой или Исходный. МестнУИР конкретизирует целевой или исходный конкретный адрес и также может иметь в качестве родительских элементов
- 40 Целевой или Исходный. Тип общего элемента Исходный используется для конкретизации исходной информации маршрутизации или отображения; его родительские элементы включают в себя: Элемент, Отображение, ЭлемОтображения, Искать, Синхронизировать и ЗаголСинхр. Целевой используется для конкретизации целевой информации маршрутизации или отображения, и его родительские элементы включают в себя: Элемент,
- 45 Отображение, ЭлемОтображения, Искать, Синхронизировать и ЗаголСинхр.

- Типы элементов вместилища сообщений обеспечивают поддержку основного вместилища для сообщений SyncML. Три такими типами элементов являются: SyncML, для конкретизации вместилища для сообщения SyncML, не имеющий родителей, поскольку он является тем, что называется корневым или документным элементом; ЗаголСинхр, для
- 50 конкретизации вместилища для информации пересмотра или информации маршрутизации, в качестве родительского элемента имеющий элемент SyncML; и ТелоСинхр, для конкретизации вместилища для тела или содержания в сообщении SyncML, также в качестве родительского элемента имеющий элемент SyncML.

Элементы описания данных используются в качестве элементов вместилища для данных, обмениваемых в сообщении SyncML; элементы описания данных включают в себя следующие типы элементов: Данные, для конкретизации дискретных данных SyncML, используемый (родительскими элементами) типами элементов Уведомить, Полномоч., Элемент, Статус и Искать; Элемент, для конкретизации вместилища для групп данных, используемый (родительскими элементами) типами элементов Добавить, Уведомить, Копировать, Исполнять, Получать, Поместить, Заменить, Результаты и Статус; и Мета, для конкретизации мета-информации о типе родительских элементов, используемый (родительскими элементами) типами элементов Добавить, Элементарный, Запросить, Копировать, Полномоч., Удалить, Получить, Элемент, Отображать, Поместить, Заменить, Результаты, Искать, Последовательность и Синхронизировать.

Элементы руководства протоколом включают в себя в настоящее время только тип элемента Статус, для конкретизации кода статуса требования на указанную команду SyncML, используемый (родительским элементом) ТелоСинхр.

Наконец, имеются элементы команд протокола. Они включают в себя уже упомянутые элементы команд, т.е., к примеру, Добавить, для конкретизации того, что данные подлежат добавлению к набору данных, используемый (родительскими элементами) Элементарный, Последовательность, Синхронизировать, ТелоСинхр, Удалить, Заменить и т.д.

Все об этих типах элементов установлено в стандарте, протоколе представления SyncML, доступном в Интернете по адресу:

http://www.syncml.org/docs/syncml_represent_v11_20020215.pdf.

Сообщение об изменении в структуре директории не является проблематичным, если одно и то же приложение заботится о манипулировании данными и о манипулировании связью согласно протоколу синхронизации. Но в случае более общей задачи, в котором на каждом устройстве несколько приложений совместно используют хранилище данных, организованный как структура директории, сохраняя свои соответствующие блоки данных в одной или более папок в этой структуре директории, неясно, как лучше построить протокол синхронизации для этого. Согласно открытому стандарту - протоколу синхронизации SyncML, продукту Инициативы SyncML, - каждое из двух удаленных устройств, имеющих соответствующие хранилища данных, подлежащие синхронизации (в любом отношении), включает в себя агент синхронизации (sync) (один на устройство, а потому обслуживающий, возможно, много различных приложений) и один или более прикладных объектов, а нижеследующий протокол синхронизации (т.е. обеспечение связи согласно этому протоколу синхронизации, установленному в стандарте) несет ответственность не за эти приложения, а за агентов синхронизации. (Только агент синхронизации, а не приложения, понимает SyncML.) Но только приложения знают, заменен ли блок данных после самой свежей уже выполненной синхронизации. Имеется несколько опций, относящихся к расширению протокола синхронизации SyncML, которые требуют включить в каждое из приложений знание SyncML наподобие того, что кодируется в агенте синхронизации, и некоторые опции, которые требуют, чтобы агент синхронизации был способен интерпретировать блоки данных таким образом, как это делают различные приложения.

Поэтому необходимо, чтобы SyncML (или любой аналогичный язык синхронизации) ссылался на такие аспекты организации данных, как папки в структуре директории, так, чтобы минимизировать запросы агента синхронизации и приложения и тем самым избежать необходимости тиражировать информацию кода или папки.

Раскрытие изобретения

Соответственно, в первом аспекте изобретения предлагается способ для использования первым агентом синхронизации, работающим совместно с первым хранилищем данных, и вторым агентом синхронизации, работающим совместно со вторым хранилищем данных, посредством чего первое хранилище данных синхронизируется со вторым хранилищем данных, причем эти хранилища данных используются для хранения данных как блоков

данных в папках первым и вторым соответствующими приложениями, отличными от первого и второго агентов синхронизации, при этом папки в сочетании определяют структуру данных, содержащий: этап, в котором первый и второй агенты синхронизации устанавливают транспортное соединение для обеспечения связи между первым и вторым агентами синхронизации; и этап, в котором второй агент синхронизации передает по транспортному соединению сообщение первому агенту синхронизации, причем это сообщение выражается с помощью языка разметки и имеет по меньшей мере один элемент идентификации данных (выражение *элемент идентификации данных* используется здесь, чтобы указать либо на то, что в SyncML называется Целевым элементом, либо на то, что в SyncML называется Исходным элементом, но следует также понимать, что оно охватывает подобные же элементы в любом языке разметки, похожем на SyncML, так что выражение *элемент идентификации данных* противопоставляется выражению *элемент данных*, которое относится к элементу в SyncML или подобном языке, который содержит блок данных); при этом информация об изменении в структуре данных первого или второго хранилищ данных передается в упомянутом сообщении; и дополнительно при этом упомянутая информация об изменении в структуре данных первого или второго хранилища данных помещается в по меньшей мере один элемент идентификации данных.

В соответствии с первым аспектом изобретения по меньшей мере один элемент идентификации данных может быть внешним для элемента данных, т.е. не содержащимся в этом элементе данных (т.е. особый тип элемента идентификации данных в языке разметки SyncML или подобный же элемент сходного языка разметки). Также в соответствии с первым аспектом изобретения информация об изменении в структуре данных первого или второго хранилища данных может включать в себя идентификацию папки. Также по меньшей мере один элемент идентификации данных может быть либо целевым элементом, либо исходным элементом. К тому же, элемент идентификации данных может содержаться в командном элементе протокола, и этот командный элемент протокола в комбинации с элементом идентификации данных может индентифицировать информацию об изменении в структуре данных первого или второго хранилища данных. И еще, элемент идентификации данных может содержать так называемый (в языке разметки SyncML) элемент МестнУИР (или подобный же элемент в сходном языке разметки), назначенный вторым агентом синхронизации, и информация об изменении структуры данных первого или второго хранилища данных может быть представлена в элементе МестнУИР.

Помимо этого, в соответствии с первым аспектом изобретения первое устройство может выполнять роль ведущего для первого агента синхронизации, а также для первого хранилища данных и первого приложения, а второе устройство может выполнять роль ведущего для второго агента синхронизации, а также второго хранилища данных и второго приложения. Далее, первое устройство может функционировать в качестве клиента в протоколе клиент/сервер, а второе устройство может функционировать в качестве сервера в протоколе клиент/сервер, и сообщение может поставляться клиентом серверу. Далее также первое устройство может функционировать в качестве сервера в протоколе клиент/сервер, а второе устройство может функционировать в качестве клиента в протоколе клиент/сервер, и сообщение может поставляться сервером клиенту, а шаг передачи сообщения может быть откликом на клиентское сообщение и может (также) включать в себя разрешение любых конфликтов, вызванных клиентским сообщением в отношении хранилища данных, для которого роль ведущего выполняет сервер.

Также в соответствии с первым аспектом изобретения первое приложение может быть откликом на любой элемент данных, включенный в сообщение, а первый агент синхронизации может быть откликом на элемент идентификации данных и может вносить изменения в структуру данных первого хранилища данных на основании информации, переданной элементом идентификации данных, перед тем, как первое приложение внесет какие-либо изменения в блоки данных хранилища данных на основании информации, переданной любыми элементами данных, включенными в это сообщение. Также второй

агент синхронизации может представлять в элементе идентификации данных информацию об изменении в структуре данных первого или второго хранилища данных, а второе приложение может представлять в элементе данных информацию об изменении в блоках данных в первом или втором хранилище данных. Помимо этого, первый агент синхронизации в ответ на это сообщение может извлекать информацию об изменении в структуре данных из элемента идентификации данных и может представлять эту информацию как процессору синхронизации, так и прикладному объекту.

Во втором аспекте изобретения предлагается устройство, работающее согласно первому аспекту изобретения в отношении работы первого устройства, а также в отношении работы второго устройства. В соответствии со вторым аспектом изобретения это устройство может быть терминалом беспроводной связи. Также это устройство может быть, вместо того, терминалом проводной связи. Еще это устройство может функционировать как сервер в модели клиент-сервер и может включать в себя процессор синхронизации для разрешения конфликтов, вызванных сообщением.

В третьем аспекте изобретения предлагается система, содержащая первое устройство как во втором аспекте изобретения, при этом первое устройство функционирует как сервер в модели клиент-сервер и включает в себя процессор синхронизации для разрешения конфликтов, вызванных сообщением; а также содержащая второе устройство во втором аспекте изобретения, при этом второе устройство функционирует как клиент в модели клиент-сервер.

В четвертом аспекте изобретения предлагается компьютерная программа для индикации операционных команд для устройства (11 12) связи, чтобы это устройство (11 12) связи работало согласно первому аспекту изобретения в отношении первого устройства, а также в отношении второго устройства.

В пятом аспекте изобретения предлагается компьютерная программа для индикации операционных команд для агента синхронизации, чтобы этот агент синхронизации работал согласно первому аспекту изобретения в отношении первого агента синхронизации, а также в отношении второго агента синхронизации.

Таким образом, изобретение расширяет SyncML, чтобы обеспечить путь для синхронизации хранилищ данных к изменениям в папках, причем тех хранилищ данных, которые могут использоваться приложениями либо в связи с пользовательскими данными, либо в связи с руководством устройствами; теперь как изменения в папках, так и изменения в блоках данных в одном хранилище данных могут отражаться в другом хранилище данных.

Изобретение имеет преимущество в том, что не требует, чтобы каждое приложение в устройстве кодировалось для понимания SyncML, а только для взаимодействия с агентом синхронизации для устройства и - в сервере - для процессора синхронизации. Вдобавок изобретение избегает необходимости включать в агент синхронизации в каждом устройстве какое-либо знание о том, как интерпретировать блоки данных для различных приложений, совместно использующих хранилище данных, синхронизируемое с соответствующим хранилищем данных в другом устройстве.

Краткое описание чертежей

Вышеуказанные и другие цели, признаки и преимущества изобретения станут понятны из рассмотрения последующего подробного описания, представленного вместе с сопровождающими чертежами, на которых:

Фиг. 1 является блок-схемой/схемой сигналов клиента (такого как мобильный телефон) и сервера (такого как сетевой сервер или ПК), обменивающихся сообщениями SyncML для синхронизации их соответствующих хранилищ данных согласно изобретению;

Фиг. 2 представляет собой схему сообщения SyncML согласно изобретению;

Фиг.3 является блок-схемой алгоритма, иллюстрирующей одно размещение для взаимодействия между компонентами клиента и сервера во время синхронизации с помощью сообщений, построенных согласно изобретению; и

Фиг.4 представляет собой схему структуры директории (согласно прототипу).

Наилучший режим осуществления изобретения

На фиг.1 клиент 11 показан синхронизирующим клиентское хранилище 11с данных с серверным хранилищем 12с данных в сервере 12. (Эти хранилища данных могут использоваться либо в руководстве устройствами, либо для хранения пользовательских данных, таких, к примеру, как электронная почта). Хранилище данных в общем случае может включать в себя одну (или даже больше) структуру системной директории и возможно также одну или более структур внутренней директории (т.е. структура внутренней директории является структурой, поддерживаемой одним или более приложениями в противоположность операционной системе) с любой структурой директории, способной быть представленной, как проиллюстрировано на фиг.4, чтобы структура директории представляла собой древовидную структуру папок, а каждая папка способна включать в себя один или более блоков данных (таких как файлы, т.е. блоки данных, поддерживаемые в структуре системной директории, или блоки либо данные, поддерживаемые приложениями во внутренней директории, такими как электронная почта, поддерживаемая приложением электронной почты) и одну или более других папок древовидной структуры. Синхронизация происходит через обмен сообщениями SyncML, хотя могут также использоваться сообщения согласно любому, подобным же образом структурированному или аналогичному языку. Некоторые из обмениваемых сообщений SyncML включают в себя команды для синхронизации хранилищ 11с 12с данных в отношении блоков данных, а некоторые - для синхронизации хранилищ данных в отношении организации данных, т.е. чтобы оба хранилища данных имели эквивалентную структуру в терминах папок, по меньшей мере постольку, поскольку приложения касаются того, в чью пользу выполняется синхронизация. Клиент 11 представляет собой любое устройство, которое играет роль клиента в модели клиент-сервер, и согласно протоколу SyncML клиент (обычно носимый компьютер, или портативный телефон, или иной беспроводный терминал) посылает все изменения в своем хранилище 11с данных на сервер 12с - устройству (обычно, к примеру, настольный компьютер), играющему роль сервера в модели клиент-сервер. (Обычно предполагается, что из двух устройств, синхронизирующих свои соответствующие хранилища данных, устройство с наибольшей вычислительной способностью будет играть роль сервера).

Клиент 11 и сервер 12 включают в себя один или более соответствующих прикладных объектов 11а 12а, и они также включают в себя (родительские, т.е. не конкретизированные для конкретного приложения) соответствующие агенты 11b 12b синхронизации. Прикладной объект 12а сервера отличается от прикладного объекта 11а клиента в том, что он не только взаимодействует с серверным агентом синхронизации, но также взаимодействует с серверным процессором 12f синхронизации, который не имеет аналога в клиенте 11. Аналогично, серверный агент 12с синхронизации отличается от клиентского агента 11с синхронизации в том, что он также взаимодействует с серверным процессором 12f синхронизации. Прикладной объект 11а-12а клиента-сервера представляет собой, к примеру, приложение календаря, используемое для отслеживания встреч для торговца. Если встреча назначена на клиентском устройстве 11, то информация о встрече сохраняется в клиентском хранилище 11с данных в качестве нового файла. Позже, либо по запросу клиента 11, либо по подсказке сервера 12, хранилище 12с данных синхронизируется с клиентским хранилищем 11с данных в отношении новых данных (т.е. нового файла). Кроме этого, если изменение сделано в структуре директории в клиентском хранилище 11с данных, то опять клиентское хранилище 11с данных и серверное хранилище 12с данных должны синхронизироваться в этом случае в отношении структуры директории, т.е. по отношению к тому, что для хранения блоков данных для приложения используются папки. Также любое изменение, сделанное в серверном хранилище 12с данных, в том числе изменения либо в папках, либо в блоках данных для приложений 11а 12а, потребуют синхронизации клиентского хранилища 11с данных и серверного хранилища 12с данных путем обмена сообщениями SyncML.

Также на фиг. 1 сообщения SyncML могут передаваться согласно разным транспортным

соединениям 14, в том числе, например, гипертекстовый транспортный протокол (HTTP), беспроводной сеансовый протокол (WSP) и протокол обмена объектами (OBEX).

(Транспортные соединения могут обеспечиваться с помощью любого вида соединения физического уровня, в том числе либо беспроводное соединение, либо проводное

соединение). Как показано, когда сервер 12 передает сообщение SyncML клиенту 11 (и наоборот), серверный агент 12b использует интерфейс 12d синхронизации, чтобы предоставить сообщение на серверный адаптер 12е синхронизации, который, в свою очередь, предоставляет это сообщение SyncML клиенту 11 по транспортному соединению 14. На принимающем конце клиентский адаптер 11е синхронизации принимает сообщение SyncML и пропускает его на клиентский интерфейс 11d синхронизации, который, в свою очередь, предоставляет его клиентскому агенту 11b синхронизации.

Теперь, чтобы обеспечить синхронизацию хранилищ 11с 12с данных по отношению к изменениям в папках, равно как и в блоках данных (чтобы одни и те же блоки данных находились в одних и тех же или эквивалентных папках каждого устройства и содержимое каждого соответствующего блока данных было тем же самым), изобретение имеет SyncML, использующий те же самые команды для изменения папок, которые используются для изменения блоков данных; тем самым, изменения в папки делаются путем отправки сообщений, передающих операционные элементы (командные элементы протокола), такие как Синхронизировать, Добавить, Заменить и Удалить, в которых на затронутые папки делаются ссылки внутри того, что называется Элементы идентификации данных, означающие либо Целевые, либо Исходные элементы, внешние к Элементам данных. Чтобы внести изменение в блок данных, SyncML согласно прототипу вызывает сообщение, ссылаясь на блок данных внутри Элемента данных, вложенного в операционный элемент. Как раз использование ссылки на затронутые папки, внешней к Элементу данных (т.е. элемент описания данных), и дает изобретению преимущество над остальными возможными компоновками для ссылки на затронутые папки (а именно, отсутствие необходимости дублировать код, позволяющий SyncML делать синтаксический разбор каждого приложения, и отсутствие необходимости включать в каждый агент синхронизации код для интерпретации блоков данных каждого отличающегося приложения).

На фиг. 2, таким образом, согласно изобретению, папки (а тем самым и организация данных в терминах директорий), затронутые в ходе синхронизации хранилищ 11с 12с данных, идентифицируются или индицируются в элементах 26а 27а МестнУИР Исходного элемента 26 и Целевого элемента 27 в командном элементе 25 протокола (таком как Заменить), внешним для элемента 28 29 данных, т.е. не содержащимся в этом элементе данных (т.е. особый тип элемента описания данных). Как Целевой, так и Исходный элементы называются здесь *Элементами идентификации данных*, чтобы противопоставить их элементам данных 28 29, которые содержатся в блоках данных. Элементы идентификации данных согласно изобретению обычно внедрены во вместилище элементов не-данных (такое как элемент группы), которое, в свою очередь, внедрено в командный элемент 24 протокола. Командный элемент 24 протокола передается как часть элемента 23 вместилища ТелоСинхр вместе с элементом 22 вместилища ЗаголСинхр, которые оба включены в элемент вместилища SyncML, т.е. сообщение 21 SyncML.

Когда изменения сделаны как в блоки данных, так и в папки, эти папки должны синхронизироваться первыми, и сообщения формулируются так, чтобы сначала делались изменения папок, а затем изменения блоков данных. Каждое изменение (либо в блоки данных, либо в папки) совершается через операционный элемент. Изменения в папки следует расположить так, чтобы они поступили перед какими-либо изменениями в блоках данных.

Таким образом, согласно изобретению, при посылке изменения папок в сообщении, например, при добавлении новой папки клиенту, элемент данных (т.е. конкретный тип элемента описания данных) операционного элемента «добавить» остается пустым, либо такой элемент («добавить») не указывается в сообщении, и информация папки (идентификатор папки и полный тракт) передается внутри элемента МестнУИР элемента

идентификации данных (Исходного элемента или Целевого элемента), внешнего для любого элемента данных.

Нижеследующий фрагмент сообщения SyncML показывает, как в команде Заменить нужно адресовать папку, имеющую идентификатор 7 и расположенную внутри папки, имеющей идентификатор 1.

```
<Replace>
...
  <Item>
    <Source>
      <LocURI>./1/7/</LocURI>
      <LocName>My own messages</LocName>
    </Source>
  ...
</Replace>
```

При адресации блока данных в папке (в противоположность самой папке) идентификатор папки и тракт папки по отношению к корню структуры директории (в случае древовидной структуры) переносится внутри МестнУИР Целевого и Исходного типов общего элемента в операционном элементе, т.е. командном элементе протокола (таком как Заменить или Удалить). Приведенный ниже пример показывает, как адресовать группу, имеющую локально уникальный идентификатор (ЛУИ) (LUID) 123 в папке, имеющей идентификатор 7, когда папка 7 располагается внутри папки, имеющей идентификатор 1.

```
<Replace>
...
  <Item>
    <Source>
      <LocURI>./1/7/123</LocURI>
      <LocName>Hi there!</LocName>
    </Source>
  ...
</Replace>
```

Ниже дан пример сообщения SyncML согласно изобретению, включающего операционные элементы (команды), используемые для создания информации папки на принимающем конце (который может быть либо клиентом, либо сервером). В этом примере используется открытый текст XML; в беспроводной среде открытый текст обычно не используется, и вместо него используется беспроводный двоичный XML (WBXML).

```
<SyncML>
<SyncHdr>
  <VerDTD>1.0</VerDTD>
  <VerProto>SyncML/1.0</VerProto>
  <SessionID>1</SessionID>
  <MsgID>2</MsgID>
  <Target><LocURI>OBEX://123456/messaging-server</LocURI></Target>
  <Source><LocURI>IMEI:493005100592800</LocURI></Source>
</SyncHdr>
<SyncBody>
...
  <Sync>
    <CmdID>3</CmdID>
    <Target>
      <LocURI>./Messaging/MMS</LocURI>
    </Target>
    <Source>
      <LocURI>./MMS</LocURI>
```

```

</Source>
<Replace>
  <!--Sending the folder structure -->
  <CmdID>4</CmdID>
  <Item>
    <Source>
      <LocURI>./100/</LocURI>
      <LocName>My Folder</LocName>
    </Source>
  </Item>
</Replace>

```

...

</SyncML>

Вышеприведенный фрагмент сообщения переименовывает папку, имеющую МестнУИР./100/, чтобы она имела название «Моя папка».

СИНХРОНИЗАЦИЯ

Согласно изобретению (двусторонняя) синхронизация может проводиться, как описано, например, в описании SyncML Synchronization Protocol (Протокол синхронизации SyncML), версия 1.1, дополненном следующим образом.

Если имеются модификации в структуре клиентской папки после предыдущей синхронизации, то нужно соблюдать следующие требования к операционным элементам (например, Заменить, Добавить и Удалить) внутри элемента SyncML.

Отправка информации папки должна осуществляться перед отправкой каких-либо объектов сообщения, т.е. команды Заменить должны конкретизироваться в упаковке модификаций перед конкретизацией команд, определяющих какие-либо модификации для объектов сообщения.

Элемент данных не должен конкретизироваться для элемента Группы.

Если операция не является удалением, следует использовать Маркировочный элемент внутри Мета элемента в элементе Группы, чтобы нести информацию о папке (например, тип папки).

Если операция не является удалением, элемент МестнНазвание внутри Исходного в элементе Группы должен конкретизировать отображаемое название папки.

Элемент МестнУИР внутри Исходного или Целевого элемента должен конкретизировать идентификатор папки и весь тракт этой папки, т.е. идентификаторы всех возможных подпапок также должны быть конкретизированы в элементе МестнУИР. Идентификаторы папок должны ограничиваться знаком косой черты (слэш) («/»).

Если имеются модификации в объектах сообщения в клиенте после предыдущей синхронизации, то нужно соблюдать следующие требования к операционным элементам внутри элемента SyncML.

Если операция не является удалением, элемент МестнНазвание внутри Исходного в элементе Группы должен конкретизировать отображаемое название сообщения.

Если операция не является удалением, следует использовать Маркировочный элемент внутри Мета элемента в элементе Группы, чтобы нести информацию о состоянии сообщения (например, считывать/не считывать информацию).

Элемент МестнУИР внутри Исходного или Целевого элемента должен конкретизировать идентификатор группы, в том числе весь тракт, т.е. идентификаторы всех возможных папок и подпапок также должны быть конкретизированы в элементе МестнУИР. Идентификатор группы (т.е. ЛУИ) всегда должен быть представлен как последнее значение элемента МестнУИР. Идентификаторы папок должны ограничиваться знаком косой черты (слэш) («/»).

Ниже следует пример клиента, посылающего модификации к серверу.

```
<SyncML>
  <SyncHdr>
    <VerDTD>1.0</VerDTD>
    <VerProto>SyncML/1.0</VerProto>
    <SessionID>1</SessionID>
    <MsgID>2</MsgID>
5    <Target><LocURI>OBEX://123456/messaging-server</LocURI></Target>
    <Source><LocURI>IMEI:493005100592800</LocURI></Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>1</MsgRef>
```

10

15

20

25

30

35

40

45

50

```

    <CmdRef>0</CmdRef>
    <Cmd>SyncHdr</Cmd>
    <TargetRef>IMEI:493005100592800</TargetRef>
    <SourceRef>OBEX://123456/messaging-server</SourceRef>
    <Data>212</Data> <!--Statuscode for OK, authenticated for session-->
</Status>
5 <Status>
    <CmdID>2</CmdID>
    <MsgRef>1</MsgRef>
    <CmdRef>5</CmdRef>
    <Cmd>Alert</Cmd>
    <TargetRef>./MMS</TargetRef>
    <SourceRef>./Messaging/MMS</SourceRef>
10 <Data>200</Data> <!--Statuscode for Success-->
    <Item>
        <Data>
            <Anchor xmlns='syncml:metinf'><Next>200005022T093223Z </Next></Anchor>
        </Data>
    </Item>
</Status>
15 <Sync>
    <CmdID>3</CmdID>
    <Target>
        <LocURI>./Messaging/MMS</LocURI>
    </Target>
    <Source>
        <LocURI>./MMS</LocURI>
20 </Source>
    <Meta>
        <Mem xmlns='syncml:metinf'>
            <FreeMem>8100</FreeMem>
            <!--Free memory (bytes) in MMS database on a device -->
            <FreeId>81</FreeId>
            <!--Number of free records in MMS database-->
25 </Mem>
        </Meta>
        <Replace>
            <!--Sending the folder structure -->
            <CmdID>4</CmdID>
            <Item>
                <Source>
30 <LocURI>./100</LocURI>
                <LocName>My Folder</LocName>
            </Source>
            </Item>
        </Replace>
        <Replace>
            <!--Sending the folder structure -->
35 <CmdID>5</CmdID>
            <Item>
                <Source>
                    <LocURI>./102</LocURI>
                    <LocName>My messages</LocName>
                </Source>
40 </Item>
            </Replace>
            <Replace>
                <!--Sending the actual message -->
                <CmdID>6</CmdID>
                <Meta>
                    <Type xmlns='syncml:metinf'>application/wap.mms-message</Type>
45 </Meta>
                <Item>
                    <Source>
                        <LocURI>./100/123</LocURI>
                        <LocName>Hi there!</LocName>
                    </Source>
                    <Data><!--The MMS data would be placed here.--></Data>
50

```

```

        <Meta>
          <Mark xmlns='syncml:metinf'>unread</Mark>
          <!-- Item's status is 'unread' -->
        </Meta>
      </Item>
    </Replace>
  </Sync>
  <Final/>
</SyncBody>
</SyncML>

```

Если имеются модификации в структуре серверной папки после предыдущей синхронизации, то нужно соблюдать следующие требования к операционным элементам внутри элемента SyncML.

Отправка информации папки должна осуществляться перед отправкой каких-либо объектов сообщения, т.е. команды Заменить должны конкретизироваться в упаковке модификаций перед конкретизацией каких-либо команд, определяющих модификации для объектов сообщения.

Элемент данных не должен конкретизироваться для элемента Группы.

Если операция не является удалением, следует использовать Маркировочный элемент внутри Мета элемента в элементе Группы, чтобы нести информацию о папке (например, тип папки).

Если операция не является удалением, элемент МестнНазвание внутри Исходного в элементе Группы должен конкретизировать отображаемое название папки.

Если операция является добавлением, то МестнНазвание Исходного элемента должно конкретизировать отображаемое название папки.

Элемент МестнУИР внутри Целевого в элементе Группы должен конкретизировать идентификатор папки клиентской стороны и весь тракт этой папки, т.е. идентификаторы всех возможных подпапок также должны быть конкретизированы в элементе МестнУИР. Идентификаторы папок должны ограничиваться знаком косой черты (слэш) («/»).

Если операция является добавлением, то МестнУИР Исходного элемента должен конкретизировать идентификатор папки серверной стороны и тракт. МестнУИР Целевого элемента должен конкретизировать тракт папки у клиента. Идентификатор папки не должен конкретизироваться в Целевом МестнУИР.

Если имеются какие-либо модификации в объектах сообщения в сервере после предыдущей синхронизации, то нужно соблюдать следующие требования к операционным элементам внутри элемента SyncML.

В случае, если операция является добавлением, то элемент МестнУИР внутри Исходного в элементе Группы должен конкретизировать отображаемое название сообщения. Элемент МестнУИР внутри Исходного в элементе Группы должен конкретизировать идентификатор группы, включая и весь тракт, т.е. идентификаторы всех возможных папок и подпапок также должны быть конкретизированы в элементе МестнУИР. Идентификатор группы (т.е. ЛУИ) всегда должен быть представлен как последнее значение в элементе МестнУИР. Идентификаторы папок должны ограничиваться знаком косой черты (слэш) («/»). МестнУИР Целевого элемента должен конкретизировать тракт папки у клиента, но идентификатор группы не должен быть конкретизирован внутри Целевого МестнУИР.

Если операция является модификацией, то элемент МестнНазвание внутри Целевого в элементе Группы должен конкретизировать отображаемое название сообщения. Элемент МестнУИР внутри Целевого в элементе Группы должен конкретизировать идентификатор группы, включая и весь тракт, т.е. идентификаторы всех возможных папок и подпапок также должны быть конкретизированы в элементе МестнУИР. Идентификатор группы (т.е. ЛУИ) всегда должен быть представлен как последнее значение в элементе МестнУИР. Идентификаторы папок должны ограничиваться знаком косой черты (слэш) («/»).

Если операция является удалением, то элемент МестнУИР внутри Исходного в элементе Группы должен конкретизировать идентификатор группы на клиентской стороне, включая и весь тракт клиента, т.е. идентификаторы всех возможных папок и подпапок

также должны быть конкретизированы в элементе МестнУИР. Идентификаторы папок должны ограничиваться знаком косой черты (слэш) («/»).

Если операция не является удалением, следует использовать Маркировочный элемент внутри Мета элемента в элементе Группы, чтобы нести информацию о состоянии сообщения (например, считывать/не считывать информацию).

Ниже следует пример сервера, посылающего модификации к клиенту.

```
<SyncML>
  <SyncHdr>
    <VerDTD>1.0</VerDTD>
    <VerProto>SyncML/1.0</VerProto>
    <SessionID>1</SessionID>
    <MsgID>2</MsgID>
    <Target><LocURI>IMEI:493005100592800</LocURI></Target>
    <Source><LocURI>OBEX://123456/messaging-server</LocURI></Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>2</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
      <TargetRef>OBEX://123456/messaging-server</TargetRef>
      <SourceRef>IMEI:493005100592800</SourceRef>
      <Data>200</Data>
    </Status>
    <Status><!--This is a status for the client modifications to the server.-->
    <CmdID>2</CmdID>
      <MsgRef>2</MsgRef><CmdRef>3</CmdRef><Cmd>Sync</Cmd>
      <TargetRef>./Messaging/MMS</TargetRef>
      <SourceRef>./MMS</SourceRef>
      <Data>200</Data> <!--Statuscode for Success-->
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>2</MsgRef><CmdRef>4</CmdRef><Cmd>Replace</Cmd>
      <SourceRef>./100/</SourceRef>
      <Data>200</Data> <!--Statuscode for Success-->
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>2</MsgRef><CmdRef>5</CmdRef><Cmd>Replace</Cmd>
      <SourceRef>./102/</SourceRef>
      <Data>200</Data> <!--Statuscode for Success-->
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>2</MsgRef><CmdRef>6</CmdRef><Cmd>Replace</Cmd>
      <SourceRef>./100/123</SourceRef>
      <Data>200</Data> <!--Statuscode for Success-->
```

```

</Status>
<Sync>
  <CmdID>4</CmdID>
  <Target><LocURI>./MMS</LocURI></Target>
  <Source><LocURI>./Messaging/MMS</LocURI></Source>
  <Add>
5    <!--Sending the folder structure -->
    <CmdID>5</CmdID>
    <Item>
      <Target>
        <LocURI>./102/</LocURI>
      </Target>
      <Source>
10     <LocURI>./4044/223/</LocURI>
        <LocName>My Own messages</LocName>
      </Source>
    </Item>
  </Add>
  <Replace>
    <CmdID>6</CmdID>
15    <Meta>
      <Type xmlns='syncml:metinf'>application/wap.mms-message</Type>
    </Meta>
    <Item>
      <Target>
        <LocURI>./102/1023/</LocURI>
        <LocName>A holiday picture.</LocName>
20    </Target>
      <Data><!--The MMS data would be placed here.--></Data>
      <Meta>
        <Mark xmlns='syncml:metinf'>send</Mark>
      </Meta>
    </Item>
  </Replace>
25  <Add>
    <CmdID>7</CmdID>
    <Meta>
      <Type xmlns='syncml:metinf'>application/wap.mms-message</Type>
    </Meta>
    <Item>
      <Target>
30    <LocURI>./102/</LocURI>
      </Target>
      <Source>
        <LocURI>./4044/1980556/</LocURI>
        <LocName>Hello John!</LocName>
      </Source>
      <Data><!--The MMS data would be placed here.--></Data>
35    <Meta>
        <Mark xmlns='syncml:metinf'>read</Mark>
      </Meta>
    </Item>
  </Add>
</Sync>
<Final/>
40 </SyncBody>
</SyncML>

```

В вышеприведенной команде, имеющей идентификатор команды 5, сервер использует глобально уникальный идентификатор (ГУИ) (GUID)./4044/223/ для папки с названием «Мои собственные сообщения». Как установлено в Протоколе синхронизации SyncML

(стандарте), версия 1.1, в разделе 5.3, агент синхронизации клиента назначит ЛУИ для папки, а затем вернет серверу отображение, показывающее, как ЛУИ (клиента) относится к ГУИ (сервера). (См. пример, данный в разделе 5.3.1.)

НЕКОТОРЫЕ МАНИПУЛЯЦИИ С ПАПКАМИ

Как уже пояснялось, папка представляет собой просто иной объект в сеансе синхронизации. Возьмем в качестве примера случай, когда пользователь создал новую папку в клиентском устройстве (например, переносном компьютере или мобильном телефоне), так что клиент должен послать команду Заменить, содержащую информацию о новой папке в следующем сеансе синхронизации. Это будет сделано в следующем

фрагменте сообщения SyncML согласно изобретению.

```

<Sync>
  <CmdID>4</CmdID>
  <Target><LocURI>./MMS</LocURI></Target>
  <Source><LocURI>./Messaging/MMS</LocURI></Source>
5  <Replace>
    <CmdID>5</CmdID>
    <Item>
      <Source>
        <LocURI>./4/7</LocURI>
        <LocName>My messages</LocName>
10      </Source>
    </Item>
  </Replace>
  ...
</Sync>

```

Когда получатель принимает операцию (т.е. Заменить или Добавить), которая не содержит элемент Данных, он должен предположить, что эта операция является операцией манипулирования папкой.

Создание папок

Создание папок осуществляется конкретизацией команды Заменить или Добавить. Нижеследующее является примером того, как создать новую папку, имеющую идентификатор 7, в папке Поступило клиента (с папкой Поступило, имеющей идентификатор 1011, на клиентской стороне).

```

<Add>
  <CmdID>4</CmdID>
  <Item>
    <Target>
      <LocURI>./1011</LocURI>
    </Target>
25  <Source>
      <LocURI>./1/7</LocURI>
      <LocName>New Folder</LocName>
    </Source>
  </Item>
</Add>

```

Клиент должен вернуть на сервер информацию отображения, указывающую для сервера, как соотносить идентификатор серверной папки с идентификатором клиентской папки, а сервер должен идентифицировать папки, использующие идентификаторы клиентских папок. Тем самым, Целевой элемент внутри Группы операционного элемента, посланного сервером, всегда должен содержать тракт папки на клиентской стороне.

В случае, когда новая папка создана на сервере и к этой папке добавлено сообщение, сервер должен послать свои операции с папкой (т.е. команду Добавить, содержащую информацию папки) в первом сообщении упаковки модификаций сервера, и в этом сообщении не должен быть конкретизирован конечный флаг. Клиент должен быть способен вернуть отображающую идентификатор папки информацию на сервер в последующем сообщении, чтобы сервер мог конкретизировать идентификатор папки клиента внутри Целевого в операции Добавить, содержащей действительное сообщение.

Переименование папок

Поскольку отображаемое имя папки переносится внутри элемента МестнНазвание, нет специальной операции для переименования папок. Поэтому, когда отображаемое название папки изменено, должна быть послана команда Заменить, а элемент МестнНазвание должен содержать новое отображаемое название папки.

Перемещение папок

Поскольку весь путь к папке переносится внутри элемента МестнУИР, нет специальной операции для перемещения папок. Поэтому, когда папка перемещена в другую папку, должна быть послана команда Заменить, и новый путь к папке должен быть включен в элемент МестнУИР. Если новая папка создана в сервере, а существующая папка перемещена в новую папку, сервер должен сначала добавить новую папку клиенту, а после получения отображающей информации для идентификатора папки сервер должен

переместить существующую папку, т.е. он должен затем послать команду Заменить, содержащую новый путь к папке на клиентской стороне.

ВЗАИМОДЕЙСТВИЕ МЕЖДУ ЭЛЕМЕНТАМИ ПРИЛОЖЕНИЙ, ЭЛЕМЕНТАМИ СИНХРОНИЗАЦИИ И ПРОЦЕССОРОМ СИНХРОНИЗАЦИИ

5 Одна возможная схема, посредством которой клиентское и серверное приложения 11a 12a и элементы 11b 12b и серверный процессор 12 синхронизации взаимодействуют, чтобы манипулировать изменениями в организации данных (структуре директории), указана на фиг. 1, а также на фиг. 3, но такие схемы не являются предметом изобретения. Важно понимать, что изобретение обеспечивает, чтобы элементы 11b 12b синхронизации клиента
10 и сервера не нуждались в синтаксическом анализе сообщения SyncML до уровня элемента данных, чтобы извлечь информацию для действий с изменениями в организации данных, и что клиентское и серверное приложения 11a 12a не нуждаются в понимании чего бы то ни было о структуре директории, чтобы обращаться к своим соответствующим хранилищам 11c 12c данных, которые должны быть способны к добавлению, замене или удалению
15 данных.

Тем не менее, возвращаясь к фиг. 1 и 3, в одном размещении для синхронизации двух хранилищ 11c 12c данных на первом шаге 31a агент 12b синхронизации сервера устанавливает транспортное соединение с агентом 11b синхронизации клиента (обратное действие, предпринимаемое агентом синхронизации клиента, предполагается, но не
20 показано на фиг. 3). На следующем шаге 31b клиентское приложение 11a (предположим, только одно приложение, хотя может быть и больше одного, причем все совместно используют одно и то же хранилище данных) находит изменения в хранилище 11c данных клиента с момента последней синхронизации, а агент синхронизации клиента делает то же самое для папок в хранилище 11c данных. Вслед за этим, на шаге 31c агент
25 синхронизации помещает изменения папок в команды (командные элементы протокола) с помощью элементов идентификации данных, чтобы сослаться на затронутые папки, при этом, как поясняется выше, элементы идентификации данных являются внешними для любых элементов данных. Далее, на шаге 31d приложение помещает изменения блока данных в команды с помощью элемента описания данных (элемент данных, группы или
30 мета). Затем, на шаге 31e агент синхронизации посылает на сервер 12 сообщение, несущее команды с элементами описания данных и элементами идентификации данных (внешними для любых элементов данных).

Затем, на сервере 12, на следующем шаге 31f агент 12b синхронизации выделяет изменения папок из любых команд с элементами идентификации данных клиента,
35 внешними для любых элементов данных, и на основании того, какие команды содержат элементы идентификации данных клиента, указывает процессору 12f синхронизации, какие изменения были сделаны в структуре данных клиентского хранилища данных с момента последней синхронизации. На следующем шаге 31g процессор синхронизации разрешает конфликты папок и предоставляет чистые изменения папок агенту 12b синхронизации
40 сервера; агент 12b синхронизации сервера делает соответствующие изменения папок в хранилище 12c данных сервера; и затем агент синхронизации сервера предоставляет (серверному) процессору синхронизации остальное сообщение (передающее любые изменения блока данных клиентом). На следующем шаге 31h процессор 12f синхронизации разрешает любые конфликты блоков данных с хранилищем данных сервера, которые
45 появились бы, если изменения, сделанные в хранилище 11c данных клиента, были сделаны также в хранилище 12c данных сервера, и предоставляет чистые изменения блока данных серверному приложению 12a. На следующем шаге 31i серверное приложение делает указанные изменения блока данных в хранилище 12c данных сервера и помещает эти чистые изменения блока данных в хранилище 11c данных клиента в соответствующих
50 командах. Затем, на следующем шаге 31j агент синхронизации сервера собирает чистые изменения, которые должны быть сделаны в папках в хранилище 11c данных клиента, и добавляет соответствующие команды, включающие в себя элементы идентификации данных, указывающие затронутые папки, причем эти элементы идентификации данных

помещаются снаружи от любых элементов данных внутри сообщения. (Это сообщение затем отправляется клиенту).

Когда клиент 11 принимает серверное сообщение, на следующем шаге 31k агент 11b синхронизации клиента выделяет все изменения папок путем синтаксического разбора команд вплоть до - но все же не включая - элементы данных, включенные в команды, изменяет соответственно хранилище 11с данных клиента и предоставляет приложению 11a все команды для изменений блока данных. Наконец, на шаге 31m приложение 11a изменяет хранилище 11с данных клиента согласно командам, предоставленным агентом 11b синхронизации клиента.

ОБЪЕМ ИЗОБРЕТЕНИЯ

Следует понимать, что вышеописанные размещения являются лишь иллюстрациями применения принципов настоящего изобретения. Специалисты могут придумать множество модификаций и альтернативных размещений без отхода от объема настоящего изобретения, и приложенная формула изобретения направлена на то, чтобы покрыть такие модификации и размещения.

Формула изобретения

1. Способ для синхронизации хранилища данных, предназначенный для, по меньшей мере, частичной синхронизации первого хранилища данных, размещенного в первом устройстве для манипуляции с данными, и второго хранилища данных, размещенного во втором устройстве для манипуляции с данными, причем каждое хранилище данных используется для хранения данных в виде блоков данных в папках, при этом папки в сочетании определяют структуру данных, указанный способ содержит этапы, на которых: первое устройство для манипуляции с данными отправляет сообщение второму устройству для манипуляции с данными;

при этом информацию о, по меньшей мере, некоторых данных блока данных, размещенных в структуре данных в первом хранилище данных, передают в указанном сообщении и информацию о размещении также передают в указанном сообщении в элементе или поле указанного сообщения; и

при этом указанную информацию о, по меньшей мере, некоторых данных блока данных в первом хранилище данных помещают в элемент или поле, отличном от элемента или поля, в котором помещена информация о размещении.

2. Способ по п.1, в котором элемент или поле, в котором помещена информация о, по меньшей мере, некоторых данных блока данных в первом хранилище данных, или элемент или поле, в котором помещена информация о размещении, представляют собой поле указанного сообщения.

3. Способ по п.1, в котором информация о, по меньшей мере, некоторых данных блока данных в первом хранилище данных включена в элемент данных указанного сообщения.

4. Способ по п.3, в котором элемент данных представляет собой элемент данных командного элемента протокола.

5. Способ по п.1, в котором информация о размещении включена в не относящийся к данным элемент сообщения.

6. Способ по п.5, в котором не относящийся к данным элемент представляет собой не относящийся к данным элемент командного элемента протокола.

7. Способ по п.1, в котором информация о размещении включает в себя информацию о папке.

8. Способ по п.1, в котором элемент идентификации данных содержится в командном элементе протокола в сообщении, при этом командный элемент протокола в сочетании с элементом идентификации данных представляет информацию о размещении.

9. Способ по п.1, в котором элемент идентификации данных содержится в сообщении, при этом информация о размещении предоставляется в элементе идентификации данных.

10. Способ по п.1, в котором первое устройство для манипуляции с данными функционирует в качестве клиента протокола клиент-сервер и второе устройство для

манипуляции с данными функционирует в качестве сервера протокола клиент-сервер.

11. Способ по п.1, в котором первое устройство для манипуляции с данными функционирует в качестве сервера протокола клиент-сервер и второе устройство для манипуляции с данными функционирует в качестве клиента протокола клиент-сервер, при этом этап отправки первым устройством для манипуляции с данными сообщения является ответным на клиентское сообщение из второго устройства для манипуляции с данными и включает в себя разрешение любых конфликтов, возникающих от клиентского сообщения в отношении первого хранилища данных.

12. Способ по п.1, в котором данные в хранилищах данных используются для управления устройством приложениями, выполняемыми на указанных устройствах.

13. Способ по п.1, в котором данные в хранилищах данных используются в качестве пользовательских данных приложениями, выполняемыми на указанных устройствах.

14. Машиночитаемый носитель данных, несущий код компьютерной программы, включающий в себя инструкции для выполнения этапов способа по п.1.

15. Первое устройство для манипуляции с данными, выполненное с возможностью, по меньшей мере, частичной синхронизации первого хранилища данных, размещенного в первом устройстве для манипуляции с данными, со вторым хранилищем данных, размещенным во втором устройстве для манипуляции с данными, причем каждое хранилище данных используется для хранения данных в виде блоков данных в папках, при этом папки в сочетании определяют структуру данных, указанное первое устройство для манипуляции с данными содержит:

средство для отправки сообщения второму устройству для манипуляции с данными; при этом информацию о, по меньшей мере, некоторых данных блока данных, размещенных в структуре данных в первом хранилище данных, передают в указанном сообщении, и информацию о размещении также передают в указанном сообщении в элементе или поле указанного сообщения; и

при этом указанную информацию о, по меньшей мере, некоторых данных блока данных в первом хранилище данных помещают в элемент или поле, отличном от элемента или поля, в котором помещена информация о размещении.

16. Устройство по п.15, в котором устройство представляет собой либо терминал беспроводной связи, либо терминал проводной связи.

17. Устройство по п.15, в котором первое устройство функционирует в качестве клиента модели клиент-сервер.

18. Устройство по п.15, в котором устройство функционирует в качестве сервера в модели клиент-сервер и дополнительно включает в себя средство для приема запроса синхронизации от второго устройства и последующей отправки сообщения в ответ на запрос синхронизации.

19. Устройство по п.15, дополнительно включающее в себя средство для приема сообщения, при этом устройство функционирует в качестве сервера в модели клиент-сервер и включает в себя средство для разрешения конфликтов, вызванных сообщением.

20. Устройство по п.15, в котором данные в хранилищах данных используются для управления устройством приложениями, выполняемыми на указанных устройствах.

21. Устройство по п.15, в котором данные в хранилищах данных используются в качестве пользовательских данных приложениями, выполняемыми на указанных устройствах.

22. Система для манипуляции с данными, содержащая первое устройство для манипуляции с данными, в котором размещено первое хранилище данных, и второе устройство для манипуляции с данными, в котором размещено второе хранилище данных, причем каждое хранилище данных используется для хранения данных в виде блоков данных в папках, при этом папки в сочетании определяют структуру данных, указанная система для манипуляции с данными дополнительно содержит:

средство для отправки сообщения второму устройству для манипуляции с данными; при этом информацию о, по меньшей мере, некоторых данных блока данных,

размещенных в структуре данных в первом хранилище данных, передают в указанном сообщении и информацию о размещении также передают в указанном сообщении в элементе или поле указанного сообщения; и

5 при этом указанную информацию о, по меньшей мере, некоторых данных блока данных в первом хранилище данных помещают в элементе или поле, отличном от элемента или поля, в котором помещена информация о размещении.

23. Система по п.22, в которой первое устройство для манипуляции с данными функционирует в качестве сервера модели клиент-сервер и второе устройство для манипуляции с данными функционирует в качестве клиента модели клиент-сервер.

10 24. Система по п.23, в которой средство для отправки сообщения второму устройству отвечает второму устройству на запрос синхронизации, посланный вторым устройством.

15

20

25

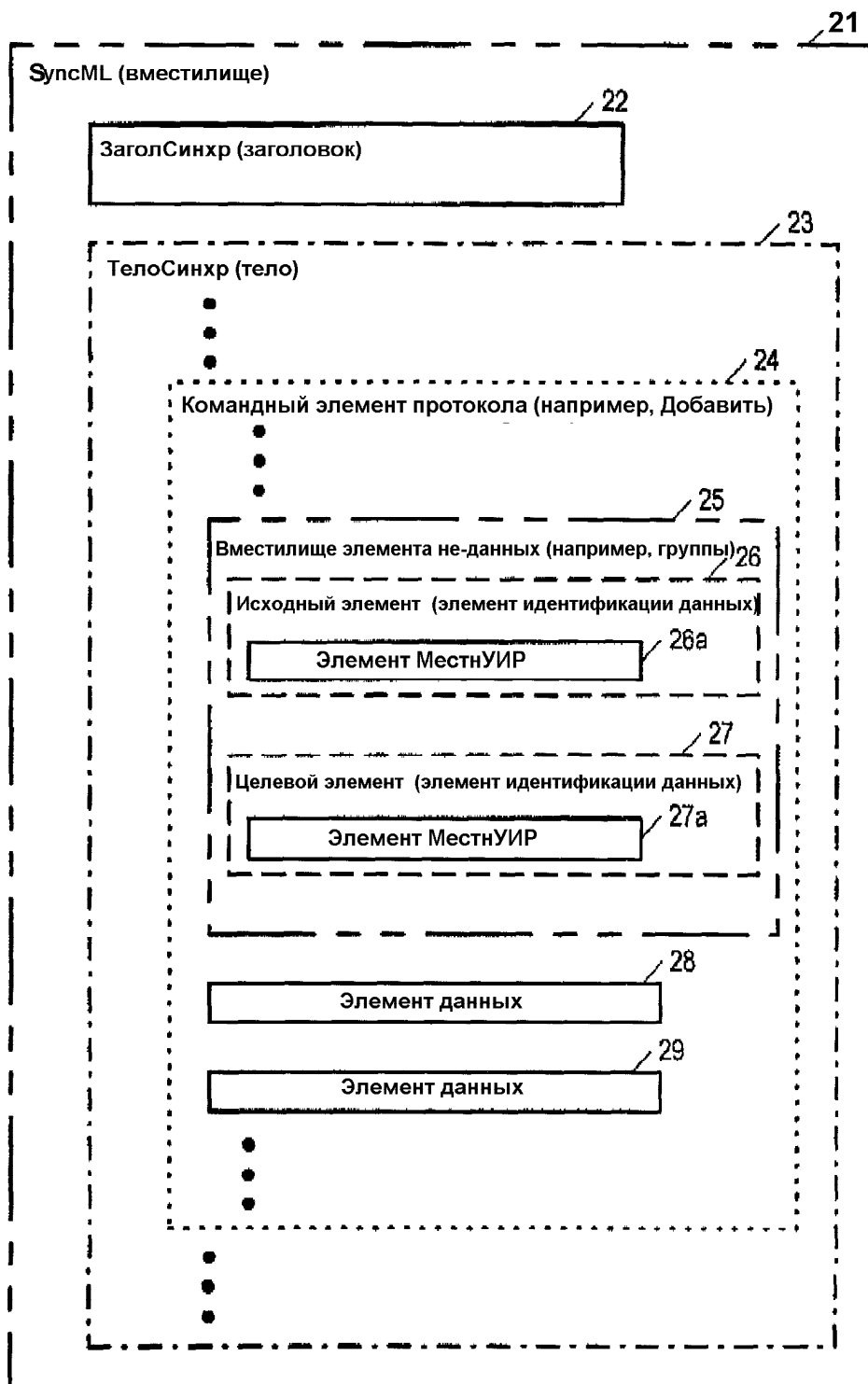
30

35

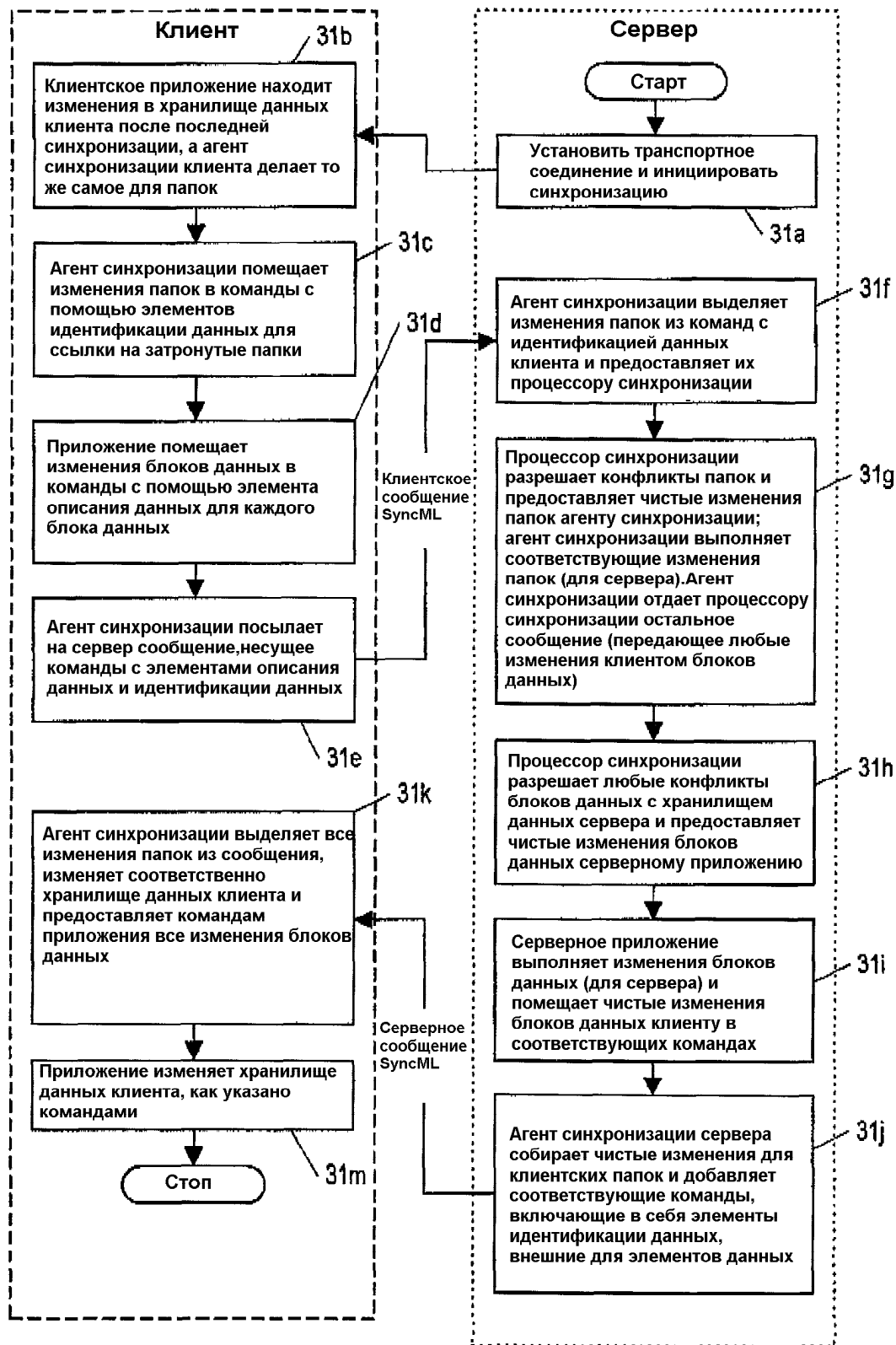
40

45

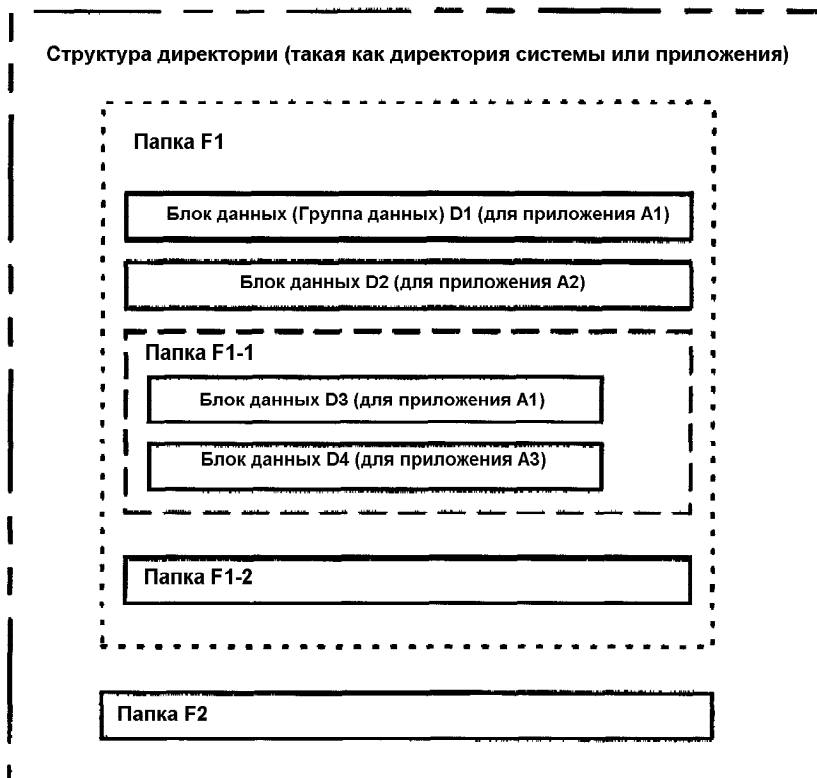
50



ФИГ. 2



ФИГ. 3



ФИГ. 4

(Предшествующий уровень техники)