РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 378 686** (13) **C2**

(51) МПК
G06F 9/00 (2006.01)
H04L 29/00 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2005101833/09, 26.01.2005**

(24) Дата начала отсчета срока действия патента:
26.01.2005

(30) Конвенционный приоритет:
12.03.2004 US 10/799,440

(43) Дата публикации заявки: **10.07.2006**

(45) Опубликовано: **10.01.2010** Бюл. № 1

(56) Список документов, цитированных в отчете о поиске: **WO 02/46909 A1, 13.06.2002. RU 2149444 C1, 20.05.2000. US 2003088684 A1, 08.05.2003. US 6199204 B1, 06.03.2001.**

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):
АВЕРБУХ Аарон Г. (US),
МАРЛ Дэннис Крэйг (US),
ДЭГХЭН Дэвид Б. (US),
МЕНЗИС Дэрек П. (US),
ФИШЕР Джинетт Р. (US),
ШЕПАРД Марк (US),
ХАНГ Сеонг Коок (US)

(73) Патентообладатель(и):
МАЙКРОСОФТ КОРПОРЕЙШН (US)

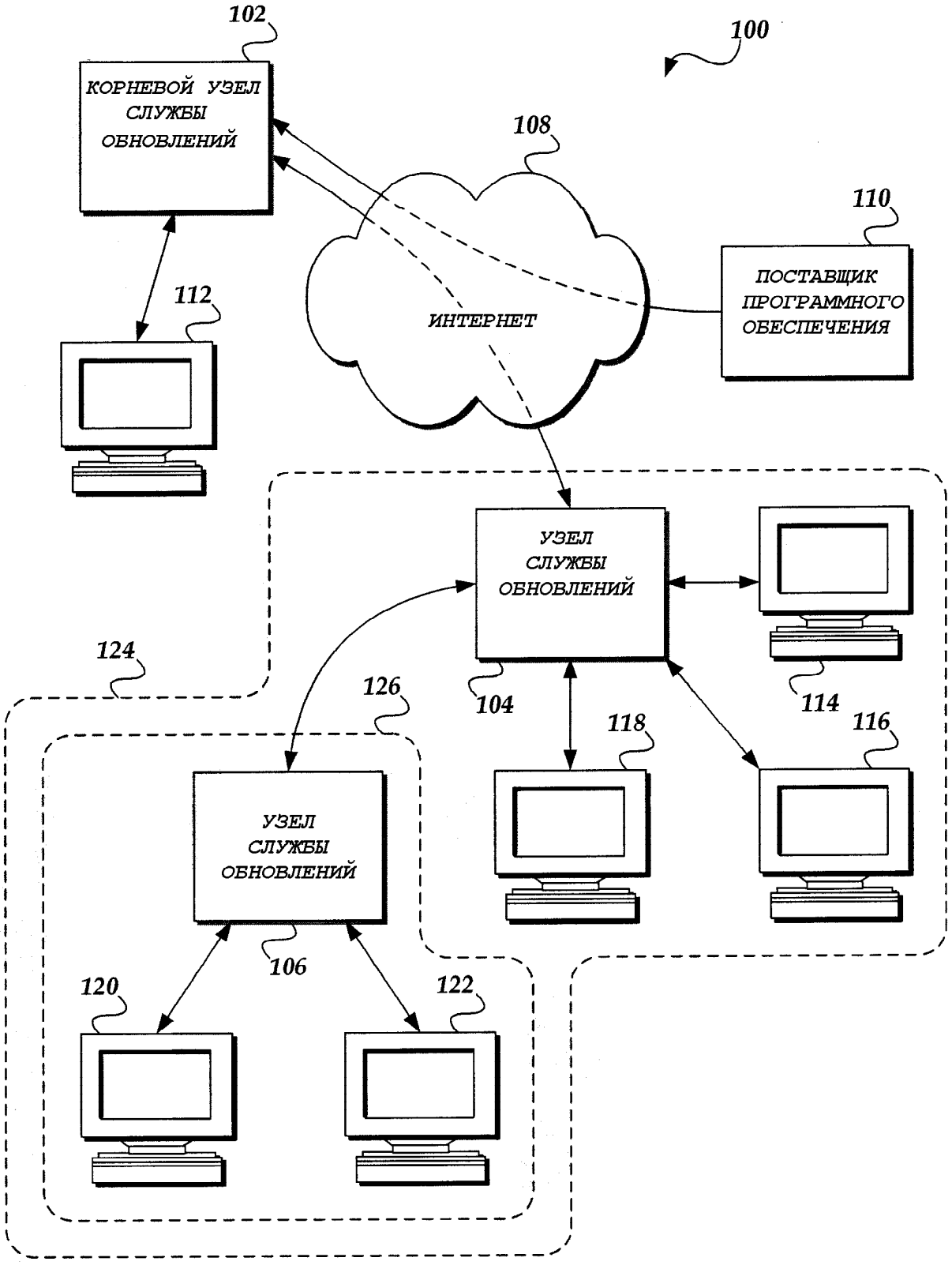
(54) ТЭГОВАЯ СХЕМА РАСПРОСТРАНЕНИЯ МЕТАДАННЫХ ОБНОВЛЕНИЯ В СИСТЕМЕ РАСПРОСТРАНЕНИЯ ОБНОВЛЕНИЙ

(57) Реферат:

Изобретение относится к средствам системы распространения обновлений. Техническим результатом является обеспечение контроля за распространением обновлений программного обеспечения. В способе передают метаданные обновления программного обеспечения, содержащие файл метаданных обновления программного обеспечения, который уникально идентифицирует обновления, тэги общих свойств, которые несут информацию общих свойств, относящуюся к обновлению, тэги локализованных свойств, которые несут информацию локализованных свойств, организованную в соответствии с языком, тэги соотношений, которые идентифицируют соотношения зависимости, которые текущее обновление программного обеспечения, согласно описанному в метаданных обновлению, имеет с другими обновлениями, тэги правил применимости, которые несут информацию для определения применимости обновления к компьютеру клиента, тэги файлов полезной нагрузки обновления и тэги данных, зависящие от обработчиков. 2 н. и 19 з.п. ф-лы, 10 ил.

RU 2378686 C2

RU 2378686 C2



ФИГ. 1

RU 2378686 C2

RU 2378686 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(19) **RU** (11) **2 378 686** (13) **C2**

(51) Int. Cl.
G06F 9/00 (2006.01)
H04L 29/00 (2006.01)

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2005101833/09, 26.01.2005**

(24) Effective date for property rights:
26.01.2005

(30) Priority:
12.03.2004 US 10/799,440

(43) Application published: **10.07.2006**

(45) Date of publication: **10.01.2010 Bull. 1**

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):
AVERBUKh Aaron G. (US),
MARL Dehnnis Krehjg (US),
DEhGKhEhN Dehvid B. (US),
MENZIS Dehrek P. (US),
FIShER Dzhinett R. (US),
ShEPARD Mark (US),
KhANG Seong Kook (US)

(73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

(54) TAG-BASED SCHEMA FOR DISTRIBUTING UPDATE METADATA IN UPDATE DISTRIBUTION SYSTEM

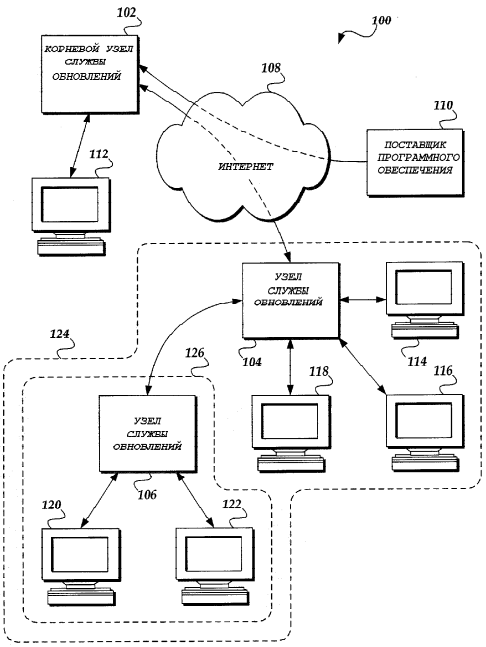
(57) Abstract:

FIELD: physics; computer engineering.

SUBSTANCE: invention relates to apparatus of an update distribution system. The method involves sending software update metadata, which contain a software update metadata file which uniquely identifies updates, general property tags which carry general property information relating to the update, localised property tags which carry localised property information organised according to language, relationship tags which identify dependency relationships the current software update, as described in the update metadata, has with other updates, applicability rules tags which carry information for determining applicability of the software update to a client computer, update payload file tags and data tags which depend on the handler.

EFFECT: control of distribution of software updates.

21 cl, 10 dwg



ФИГ. 1

RU 2378686 C2

RU 2378686 C2

ОПИСАНИЕ

Область техники, к которой относится изобретение

Настоящее изобретение относится к программному обеспечению и компьютерным сетям, и в частности настоящее изобретение относится к тэговой схеме распространения метаданных обновления в системе распространения обновлений.

Уровень техники

Почти все коммерчески доступные программные продукты проходят процесс непрерывной ревизии для исправления или обновления особенностей программного обеспечения. Каждая ревизия программного продукта часто требует добавления новых файлов, замены существующих файлов более новыми ревизиями, удаления устаревших файлов или различных комбинаций этих действий. Этот процесс замены более старых файлов, добавления новых файлов и удаления устаревших файлов программного продукта будем в дальнейшем называть «обновлением продукта», а совокупность данных, в том числе двоичных файлов, файлов данных, инструкций по обновлению, метаданных и т.п., используемых при обновлении продукта, будем в дальнейшем называть просто «обновление».

После того как поставщик программного обеспечения создал обновление программного продукта с целью устранения проблемы, либо повышения безопасности, либо добавления новых особенностей, ему потребуется сделать это обновление общедоступным для потребителей. Чаще всего, например, когда обновление направлено на исправление дефекта в продукте или решение критического вопроса безопасности, поставщику программного обеспечения нужно, чтобы это обновление как можно быстрее было установлено на компьютерах потребителей. Действительно, большинство поставщиков программного обеспечения имеет бизнес, стимулирующий распространение обновлений программного обеспечения на их компьютеры как можно быстрее и как можно безопаснее.

Компьютерная промышленность испытала резкий рост количества компьютеров, подключенных к сетям, в частности к Интернету. В силу этого резкого роста и в силу возможностей связи, доступных посредством подключения к Интернету, Интернет стал важным и универсальным каналом, которым пользуются поставщики программного обеспечения для распространения обновлений среди своих потребителей. Фактически Интернет стал главным каналом распространения для многих поставщиков программного обеспечения, которым они пользуются для предоставления обновлений программного обеспечения своим потребителям. Часто, интересам поставщиков программного обеспечения в наибольшей степени удовлетворяет распространение обновлений программного обеспечения через Интернет, поскольку электронное распространение обновлений по Интернету снижает совокупные затраты и позволяет потребителям получать обновления программного обеспечения по мере их появления. Эти обновления программного обеспечения все чаще поступают через Интернет автоматически, без какого-либо вмешательства пользователя.

Хотя Интернет в настоящее время широко используется в качестве канала для распространения обновлений программного обеспечения от поставщиков программного обеспечения, часто возникает ряд вопросов. Два таких вопроса включают в себя

(1) эффективность, относящуюся к инфраструктуре/ресурсам распространения обновлений, и

(2) административный контроль над распространением и установкой обновлений

программного обеспечения.

Что касается эффективности ресурсов распространения, сети, включая Интернет, располагают лишь конечным объемом ресурсов связи, которые обычно называют пропускной способностью. Конечная величина пропускной способности приводит к узким местам, особенно в отношении обновлений программного обеспечения для популярных программных продуктов, как-то: семейству операционных систем Windows® фирмы Microsoft Corporation и связанных с ними программных продуктов. Такие узкие места существуют даже, когда обновления программного обеспечения сделаны доступными во многих местах загрузки, распределенных по Интернету. Одна из причин возникновения таких узких мест состоит в модели неструктурированного доступа, обеспечиваемой Интернетом. Например, если первый пользователь на компьютере А запрашивает новейшую загрузку программного продукта, то загрузка проходит через независимого поставщика услуг (НПУ) первого пользователя. Кроме того, запрос обрабатывается как единичный, индивидуализированный доступ, и это значит, что запрос обрабатывается независимо от любого другого сетевого трафика и/или запроса и без связи с ним. Поэтому, если второй пользователь на компьютере В, который имеет того же НПУ, запрашивает ту же загрузку, что и первый пользователь, то запрос от второго пользователя также обрабатывается как единичный, индивидуализированный доступ. В этом примере одна и та же загрузка будет дважды передаваться по одной и той же инфраструктуре, поскольку каждый запрос обрабатывается по отдельности. Очевидно, что при существенном увеличении количества пользователей конечная пропускная способность канала связи приведет к появлению узкого места. В этом примере, который является весьма общим, было бы гораздо эффективнее, если бы загрузка кэшировалась в одном месте и каждый пользовательский запрос удовлетворялся из локального кэша.

Что касается контроля распространения, многие организации, в особенности крупные организации, имеют законные причины контролировать распространение обновлений на свои компьютеры. Например, к сожалению, некоторые обновления имеют или привносят дефекты, которые часто называют ошибками, которые нарушают особенности программного продукта. Эти нарушенные особенности могут быть несущественными, но, в большом количестве, они могут нарушать особенности, важные для предприятия. Поскольку предприятие не может позволить себе потерять свои важные для работы особенности, ответственное предприятие сначала оценивает и испытывает каждое обновление программного обеспечения в контролируемой среде в течение некоторого периода времени прежде, чем разрешить установку обновления на остальных своих компьютерах. Этот период оценки позволяет организации проверить, не окажет ли обновление негативное влияние на особенность, важную для работы. Только после того, как будет удовлетворительно определено, что обновление не повредит никакие важные для работы особенности, разрешается распространение обновления на остальные компьютеры организации. Очевидно, что большинство организаций должно контролировать установку обновлений программного обеспечения на своих компьютерах.

Другая причина того, что предприятию или организации часто приходится контролировать распространение обновлений программного обеспечения, состоит в необходимости обеспечения согласования между компьютерами организации. Для департаментов организационных услуг очень важно иметь стандартизованную, целевую платформу, на которой работают все компьютеры, будь то для текстового

редактора или операционной системы. В отсутствие стандарта обслуживание программного обеспечения и компьютеров может оказаться излишне сложным и трудным.

Еще одна причина важности локального контроля связана с целями тарификации. В больших организациях, часто оказывается неэффективным индивидуально устанавливать программное обеспечение на компьютер и индивидуально поддерживать лицензии на тот или иной программный продукт для каждого компьютера в организации. Вместо этого, единая лицензия на использование системы позволяет организации прогонять программный продукт на многочисленных компьютерах. Таким образом, организации может потребоваться отчитаться в количестве компьютеров, на которых действует продукт, подчиняющийся лицензии на использование системы, или может потребоваться ограничить количество компьютеров, на которых действует продукт, подчиняющийся лицензии на использование системы. Все эти причины часто приводят к необходимости в локальном контроле над распространением обновлений программного обеспечения.

В свете различных вышеуказанных вопросов, связанных с распространением программного обеспечения, требуется расширяемая архитектура распространения обновлений программного обеспечения для обеспечения контроля над распространением обновлений программного обеспечения, а также для повышения эффективности их распространения. Настоящее изобретение решает эти и другие вопросы, возникающие в уровне техники.

Раскрытие изобретения

Согласно аспектам настоящего изобретения, представлена тэгированная структура для передачи метаданных обновления от узла службы обновлений к дочернему узлу службы обновлений или компьютеру-клиенту. Тэгированная структура содержит: тэг идентификатора, содержащий идентификатор обновления программного обеспечения, который уникально идентифицирует обновление программного обеспечения; нуль или более тэгов общих свойств, которые несут информацию общих свойств, относящуюся к обновлению программного обеспечения; нуль или более тэгов локализованных свойств, которые несут информацию локализованных свойств, организованную в соответствии с языком; нуль или более тэгов соотношений, которые идентифицируют соотношения зависимости, которые текущее обновление программного обеспечения, согласно описанному в метаданных обновления, имеет с другими обновлениями программного обеспечения; нуль или более тэгов правил применимости, которые несут информацию для определения применимости обновления программного обеспечения к компьютеру-клиенту; нуль или более тэгов файлов, которые несут информацию, относящуюся к файлам полезной нагрузки обновления программного обеспечения; и тэги данных, зависящих от обработчика, которые несут информацию, предназначенную для программного обработчика для установки обновления программного обеспечения.

Согласно дополнительным аспектам настоящего изобретения, представлена тэгированная структура для передачи метаданных обновления от узла службы обновлений на дочерний узел службы обновлений или компьютер-клиент. Тэгированная структура содержит: тэг идентификатора, содержащий идентификатор обновления программного обеспечения, который уникально идентифицирует обновление программного обеспечения; нуль или более тэгов соотношений, которые идентифицируют соотношения зависимости, которые текущее обновление программного обеспечения, согласно описанному в метаданных обновления, имеет с

другими обновлениями программного обеспечения; и нуль или более тэгов файлов, которые несут информацию, относящуюся к файлам полезной нагрузки обновления программного обеспечения.

Краткое описание чертежей

Вышеизложенные аспекты и многие связанные с ними преимущества изобретения явствуют из нижеследующего подробного описания, приведенного в сочетании с прилагаемыми чертежами, в которых:

фиг.1 - наглядная схема иллюстративной системы распространения обновлений, сформированной согласно аспектам настоящего изобретения;

фиг.2 - блок-схема иллюстративных логических компонентов узла службы обновлений, сформированного согласно аспектам настоящего изобретения;

фиг.3 - блок-схема иллюстративных логических компонентов корневого узла службы обновлений, сформированного согласно аспектам настоящего изобретения;

фиг.4A и 4B - блок-схемы, иллюстрирующие обмен между родительским узлом службы обновлений и дочерним узлом службы обновлений при обеспечении обновления программного обеспечения от родительского узла службы обновлений к дочернему узлу службы обновлений согласно аспектам настоящего изобретения;

фиг.5 - логическая блок-схема иллюстративной процедуры, выполняемой на дочернем узле службы обновлений для периодического получения обновлений от своего родительского узла службы обновлений;

фиг.6 - логическая блок-схема иллюстративной подпроцедуры, пригодной для использования в иллюстративной процедуре, изображенной на фиг.5, для получения каталога обновлений от родительского узла службы обновлений;

фиг.7 - логическая блок-схема иллюстративной подпроцедуры, пригодной для использования в иллюстративной процедуре, изображенной на фиг.5, для получения обновления программного обеспечения от родительского узла службы обновлений;

фиг.8 - логическая блок-схема иллюстративной процедуры для обработки запроса обновления от дочернего узла службы обновлений; и

фиг.9 - блок-схема, иллюстрирующая фрагменты иллюстративной схемы метаданных обновления на основе XML, задающей содержимое файла метаданных обновления.

Осуществление изобретения

Согласно аспектам настоящего изобретения, представлена система распространения обновлений, организованная в иерархическом порядке, для распространения обновлений программного обеспечения. На фиг.1 показана наглядная схема иллюстративной системы 100 распространения обновлений, сформированной согласно аспектам настоящего изобретения. Согласно настоящему изобретению, на «вершине» системы распространения обновлений, например показанной системы 100 распространения обновлений, находится корневой узел 102 службы обновлений. Поставщики программного обеспечения, например поставщик 110 программного обеспечения, распространяют свои обновления программного обеспечения через систему 100 распространения обновлений, подавая обновления на корневой узел 102 службы обновлений. Согласно аспектам настоящего изобретения, поставщики программного обеспечения, например поставщик 110 программного обеспечения, могут подавать свои обновления программного обеспечения на корневой узел 102 службы обновлений через сеть, например Интернет 108.

Иерархическая система распространения обновлений, например, иллюстративная

система 100 распространения обновлений, вероятно, включает в себя, по меньшей мере, один другой узел службы обновлений, помимо корневого узла 102 службы обновлений. Согласно фиг.1, иллюстративная система 100 распространения обновлений содержит корневой узел 102 службы обновлений и два дополнительных узла службы обновлений: узел 104 службы обновлений и узел 106 службы обновлений. Согласно настоящему изобретению, каждая иерархическая система распространения обновлений организована в виде древовидной структуры под корневым узлом 102 службы обновлений. Другими словами, каждый узел службы обновлений в системе распространения обновлений имеет нуль или более дочерних узлов службы обновлений. Таким образом, хотя в иллюстративной системе 100 распространения обновлений показано, что каждый родительский узел службы обновлений, то есть корневой узел 102 службы обновлений и узел 104 службы обновлений, имеют по одному дочернему узлу, это показано только в иллюстративных целях и не следует рассматривать как ограничение настоящего изобретения. Поэтому, за исключением корневого узла 102 службы обновлений, каждый узел службы обновлений в системе распространения обновлений имеет один родительский узел службы обновлений. Соответственно, согласно фиг.1, узел 104 службы обновлений является дочерним узлом по отношению к корневому узлу 102 службы обновлений, и узел 106 службы обновлений является дочерним узлом по отношению к узлу 104 службы обновлений. Можно видеть, что каждый узел службы обновлений, за исключением корневого узла 102 службы обновлений, может быть как дочерним узлом службы обновлений, так и родительским узлом службы обновлений.

Как показано в иллюстративной системе 100 распространения обновлений, корневой узел 102 службы обновлений осуществляет связь с узлом 104 службы обновлений через Интернет 108. Однако, следует понимать, что это показано только в иллюстративных целях, и не следует рассматривать как ограничение настоящего изобретения. Каждому узлу службы обновлений в системе распространения обновлений нужно иметь возможность связи только со своим предком и/или потомком через некоторую сеть связи. Таким образом, хотя узел 104 службы обновлений осуществляет связь со своим предком, корневой узел 102 службы обновлений, через Интернет 108, может альтернативно осуществлять связь со своими дочерними узлами службы обновлений, например узлом 106 службы обновлений, через локальную сеть 124.

На фиг.1 также показано, что узел 106 службы обновлений находится в подсети 126 локальной сети 124. Например, локальная сеть 124 может соответствовать общей корпоративной сети 124 организации, и узел 104 службы обновлений представляет связь корпорации с системой 100 распространения обновлений, через свое соединение со своим родителем, корневым узлом 102 службы обновлений. Кроме того, подсеть 126 может соответствовать идентифицируемой группе компьютеров в корпоративной сети, например, группе испытания/оценки, отдельно расположенному офису или группе, важной для работы. Как будет подробно описано ниже, согласно аспектам настоящего изобретения, администратор на узле 104 службы обновлений способен контролировать распространение обновлений на узел 106 службы обновлений и, в итоге, на компьютеры-клиенты.

Следует понимать, что каждый узел службы обновлений, в том числе корневой узел 102 службы обновлений и узлы 104 и 106 службы обновлений способны распространять обновления программного обеспечения как на дочерние узлы службы обновлений, так и на компьютеры-клиенты. Согласно фиг.1, иллюстративная

система 100 распространения обновлений содержит компьютеры-клиенты 112-122. Каждый узел службы обновлений, в том числе корневой узел 102 службы обновлений, распространяет обновления программного обеспечения на дочерние узлы службы обновлений и компьютеры-клиенты согласно информации локальной конфигурации. Согласно одному варианту осуществления, администратор задает группы и связывает правила распространения обновления с этими группами.

В порядке примера, иллюстрирующего работу системы распространения обновлений, предположим, что локальная сеть 124 соответствует корпоративной сети коммерческой организации, администратор на узле 104 службы обновлений может задавать множественные группы распространения для корпоративной сети 124, в том числе группу оценки, соответствующую подсети 126, содержащей 106 службы обновлений и компьютеры-клиенты 120 и 122, для оценки пригодности обновления для всей корпоративной сети 124, а также общую корпоративную группу, содержащую узел 104 службы обновлений и компьютеры-клиенты 114-118.

Что касается группы оценки, администратор включает узел 106 службы обновлений в качестве члена и связывает правила для этой группы с этой группой, так что обновления немедленно распространяются на члены группы оценки по мере своего появления. Альтернативно, в отношении общей корпоративной группы, администратор добавляет компьютеры-клиенты 114-118 и связывает правило, так что обновления распространяются на члены общей корпоративной группы только по особому разрешению администратора. Предположим также, что администратор для дочернего узла 106 службы обновлений создает группу по умолчанию, состоящую из компьютеров-клиентов 120 и 122 в подсети 126 оценки, на которые может немедленно распространяться обновление программного обеспечения.

Согласно вышеприведенному примеру, поставщик 110 программного обеспечения подает обновление программного обеспечения на корневой узел 102 службы обновлений. Согласно правилам, установленным на корневом узле 102 службы обновлений, обновление, в конечном итоге, распространяется на корпоративный узел 104 службы обновлений. Получив обновление, согласно правилам, установленным администратором, корпоративный узел 104 службы обновлений распространяет обновление на члены группы оценки (заданные только как дочерний узел 106 службы обновлений), но удерживает обновление от общей корпоративной группы, ожидая особого разрешения на распространение обновления на эту группу.

Продолжая вышеприведенный пример, получив обновление, оценочный узел 106 службы обновлений обрабатывает обновление по отношению к каждой заданной группе. В этом примере, оценочный узел 106 службы обновлений имеет только одну группу. Однако, согласно вышеупомянутому, в фактической реализации может быть задано несколько групп, каждая из которых имеет уникальное множество соответствующее множество правил распространения. В этом примере, оценочный узел 106 службы обновлений немедленно делает обновление доступным для распространения на компьютеры-клиенты 120 и 122. Теперь можно обновлять компьютеры-клиенты 120 и 122 и начинать период/процесс оценки.

По-прежнему продолжая вышеприведенный пример, когда администратор корпоративного узла 104 службы обновлений достаточно удовлетворен тем, что обновление пригодно для распространения по всей корпоративной сети 124, администратор в явном виде разрешает распространение обновления на члены общей корпоративной группы. Корпоративный узел 104 службы обновлений соответственно делает обновление доступным для компьютеров-клиентов 114-118. Следует понимать,

что оценочный узел 106 службы обновлений также может входить в состав общей корпоративной группы. Однако, поскольку оценочный узел 106 службы обновлений уже обновлен, для распространения обновления на подсеть 126 оценки не требуется никаких дополнительных действий, связанных с обновлением.

Из вышеприведенного примера можно видеть, что настоящее изобретение дает значительные преимущества в отношении локального контроля распространения и эффективности загрузки. Помимо вышеописанных аспектов локального контроля распространения, реализуется также значительная экономия пропускной способности канала связи. Например, хотя иллюстративная корпоративная сеть 124, показанная на фиг.1, содержит пять компьютеров-клиентов, обновление от поставщика программного обеспечения было загружено с корневого узла 102 службы обновлений на корпоративный узел 104 службы обновлений только один раз. Очевидно, что, по мере увеличения количества компьютеров-клиентов, обслуживаемых узлом службы обновлений, использование пропускной способности канала связи между родительским узлом службы обновлений и клиентским узлом службы обновлений остается постоянным, что существенно уменьшает величину пропускной способности канала связи по сравнению с используемой в противном случае. Дополнительно, система распространения обновлений является расширяемой и масштабируемой. Система распространения обновлений имеет возможность расширения двумя путями: к родительскому узлу службы обновлений можно добавлять любое количество дочерних узлов службы обновлений, и дочерние узлы службы обновлений также могут быть родительскими узлами службы обновлений. Каждое поддерево системы распространения обновлений можно, таким образом, конфигурировать в соответствии с индивидуальными потребностями.

На фиг.2 показана блок-схема иллюстративных логических компонентов узла 200 службы обновлений, например, корпоративного узла 104 службы обновлений (фиг.1) или оценочного узла 106 службы обновлений (фиг.1), сформированного согласно аспектам настоящего изобретения. Согласно фиг.2, узел 200 службы обновлений содержит веб-службу 202 обновлений, модуль 204 обновлений клиентов, модуль 206 обновлений потомков и модуль 208 отчетов. Иллюстративный узел 200 службы обновлений также содержит модуль 210 аутентификации/авторизации, программный интерфейс приложений (API) 212 администрирования, хранилище 214 содержимого обновлений, пользовательский интерфейс 218 администрирования и хранилище 216 информации обновлений.

Веб-служба 202 обновлений обеспечивает обычный набор веб-услуг, посредством которых компьютеры-клиенты, дочерние узлы службы обновлений и родительский узел службы обновлений могут осуществлять связь с узлом службы обновлений. Например, согласно фиг.1, чтобы дочерний/оценочный узел 106 службы обновлений мог получать обновление программного обеспечения от родительского/корпоративного узла 104 службы обновлений, клиент осуществляет связь через веб-службу 202 обновлений предка. Аналогично, когда родительский узел службы обновлений, например, корневой узел 102 службы обновлений, имеет информацию, в том числе обновления, подлежащую передаче на его дочерний узел 104 службы обновлений, родительский узел службы обновлений осуществляет связь через веб-службу 202 обновлений потомка.

Модуль 204 обновлений клиентов манипулирует передачами между компьютером-клиентом и узлом 200 службы обновлений, связанными с обновлениями и информацией обновлений, хранящимися на узле службы обновлений. Передачи,

связанные с обновлениями, включают в себя, но не только, распространение обновлений в ответ на запросы клиентов и предоставление компьютеру-клиенту списка доступных программных продуктов и соответствующих обновлений. Модуль 204 обновлений клиентов также отвечает за определение, разрешено ли компьютеру-клиенту получать конкретное обновление согласно соответствующим правилам распространения, и отвечает компьютеру-клиенту посредством информации, связанной с обновлениями, доступ к которой разрешен компьютеру-клиенту.

Модуль 206 обновлений потомков манипулирует передачами, связанными с обновлениями, между родительским узлом службы обновлений и его дочерними узлами службы обновлений. Передачи, связанные с обновлениями, включают в себя, но не только, идентификацию списков программных продуктов и соответствующих обновлений, доступных дочернему узлу службы обновлений, а также ответы на запросы обновления от дочернего узла службы обновлений. Модуль 206 обновлений потомков отвечает за определение, разрешено ли дочернему узлу службы обновлений получать конкретное обновление согласно соответствующим правилам распространения, и отвечает дочернему узлу службы обновлений посредством информации, связанной с обновлениями, доступ к которой разрешен дочернему узлу службы обновлений.

Модуль 208 отчетов генерирует отчеты об обновлениях, например, какие группы получили или не получили конкретное обновление, какие компьютеры-клиенты загрузили/установили обновление или не сделали этого, и т.п. Эти отчеты могут использоваться внутренне, например, администратором, а также передаваться на родительский узел службы обновлений через интерфейс 202 службы обновления предка. Согласно вышеописанному, корпорациям часто требуется определять, какие компьютеры-клиенты установили то или иное обновление, например, в целях тарификации или в целях обслуживания. Информация/отчеты, сгенерированные модулем 208 отчетов, может служить основой этих отчетов.

Модуль 210 аутентификации/авторизации отвечает за аутентификацию, то есть определение идентичности того или иного компьютера-клиента или дочернего узла службы обновлений, и определение, разрешен ли компьютеру-клиенту или дочернему узлу службы обновлений доступ к обновлениям, имеющимся на узле 200 службы обновлений. Тем компьютерам-клиентам и дочерним узлам службы обновлений, которые аутентифицированы и авторизованы для доступа к обновлениям на узле службы обновлений, модуль 210 аутентификации/авторизации выдает жетон авторизации, который нужно использовать для получения обновлений. Выдача и использование жетона авторизации более подробно описана ниже со ссылкой на фиг. 4A и 4B.

API 212 администрирования представляет интерфейс приложения, посредством которого осуществляется управление узлом 200 службы обновлений и посредством которого обновления окончательно сохраняются и распространяются. Когда веб-служба 202 обновлений принимает различные запросы, связанные с обновлениями, от компьютеров-клиентов и дочерних узлов службы обновлений, эти запросы, в конце концов, разбиваются на вызовы в API 212 администрирования, либо напрямую, либо косвенно, через модуль 204 обновлений клиентов и модуль 206 обновлений потомков. С помощью пользовательского интерфейса 218 администрирования или какой-либо другой программы, установленной на узле 200 службы обновлений, надлежащим образом сконфигурированной для

использования API 212 администрирования, администратор, в итоге, контролирует все аспекты процесса обновления для этого узла службы обновлений, а также любых дочерних узлов службы обновлений и компьютеров-клиентов.

Посредством пользовательского интерфейса 218 администрирования, администраторы могут конфигурировать и обслуживать узел 200 службы обновлений через API 212 администрирования. Таким образом, посредством пользовательского интерфейса 218 администрирования, администратор создает, изменяет и удаляет группы, а также соответствующие правила для каждой группы. Кроме того, используя пользовательский интерфейс 218 администрирования, администратор устанавливает, какой группе принадлежит компьютер-клиент или дочерний узел службы обновлений. Посредством пользовательского интерфейса 218 администрирования, администратор также может в явном виде разрешать распространение обновлений на компьютеры-клиенты или дочерние узлы службы обновлений, конфигурировать узел 200 службы обновлений, чтобы он периодически запрашивал свой родительский узел службы обновлений на предмет новых обновлений, конфигурировать параметры составления отчета и просматривать внутренние отчеты и т.п. Согласно вышеуказанному, хотя пользовательский интерфейс 218 администрирования позволяет администратору осуществлять контроль над аспектами узла 200 службы обновлений, вместо пользовательского интерфейса 218 администрирования можно использовать другое приложение, установленное на узле 200 службы обновлений и приспособленное для работы с API 212 администрирования.

Как указано выше, согласно одному варианту осуществления настоящего изобретения, узел 200 службы обновлений содержит как хранилище 214 содержимого обновлений, так и хранилище 216 информации обновления. В хранилище 214 содержимого обновлений хранятся фактические файлы, представляющие обновления программного обеспечения, в частности, двоичные коды и файлы заплаток. Напротив, в хранилище 216 информации обновления хранятся информация и метаданные, соответствующие обновлениям, имеющимся на узле 200 службы обновлений, включая файлы обновлений, хранящиеся в хранилище 214 содержимого обновлений. Согласно одному варианту осуществления, хранилище 214 содержимого обновлений и хранилище 216 информации обновлений являются реляционными базами данных. Хотя показано, что иллюстративный узел 200 службы обновлений содержит два хранилища данных, настоящее изобретение не ограничивается этим. В альтернативном варианте осуществления, хранилище 214 содержимого обновлений и хранилище 216 информации обновлений могут быть объединены в единое хранилище информации.

Согласно аспектам настоящего изобретения, обновление программного обеспечения можно представить как «доступное» на узле 200 службы обновлений для компьютеров-клиентов и дочерних узлов службы обновлений даже, если обновление физически не хранится в хранилище 214 содержимого обновлений. В частности, вместо того, чтобы немедленно загружать и сохранять фактические файлы обновлений на узле 200 службы обновлений, на узле службы обновлений можно сохранять ссылку на файлы обновлений на родительском узле службы обновлений или где-либо еще. Таким образом, если компьютер-клиент запрашивает обновление, или дочерний узел службы обновлений запрашивает фактическое обновление, обновление переносится от родительского узла службы обновлений и сохраняется в хранилище 214 содержимого обновлений в порядке подготовки к доставке его на компьютер-клиент или дочерний

узел службы обновлений. Специалистам в данной области известно, что этот тип доступа к обновлению называется загрузкой «точно вовремя». Таким образом, «доступное» обновление не нужно распространять по различным каналам сети, пока оно не будет фактически запрошено. Согласно аспектам настоящего изобретения, администратор узла 200 службы обновлений может избирательно определять, получать ли обновления «точно вовремя».

Хотя вышеприведенное описание фиг.2 иллюстрирует различные компоненты иллюстративного модуля 200 службы обновлений, следует понимать, что могут существовать и другие компоненты модуля службы обновлений. Кроме того, вышеописанные компоненты следует рассматривать как логические компоненты, а не обязательно физические компоненты. В фактической реализации, вышеуказанные компоненты могут быть объединены друг с другом и/или с другими компонентами, отвечающими определениям реализации. Кроме того, следует понимать, что, хотя узел 200 службы обновлений можно рассматривать как компьютер-сервер в сети, в фактической реализации, узел службы обновлений может быть реализован на вычислительном устройстве любого типа. Например, каждый узел 200 службы обновлений может быть реализован и/или установлен в единичной автономной компьютерной системе или, альтернативно, в распределенной вычислительной системе, содержащей множественные вычислительные устройства.

На фиг.3 изображена блок-схема иллюстративных логических компонентов корневого узла 300 службы обновлений, например, корневого узла 102 службы обновлений, показанного на фиг.1, сформированного согласно аспектам настоящего изобретения. По аналогии с логическими компонентами узла 200 службы обновлений (фиг.2) корневой узел 300 службы обновлений содержит веб-службу 202 обновлений, модуль 206 обновлений потомков и модуль 210 аутентификации/авторизации. Кроме того, иллюстративный корневой узел 300 службы обновлений содержит (API) 212 администрирования, хранилище 214 содержимого обновлений, и хранилище 216 информации обновлений. В необязательном порядке, корневой узел 300 службы обновлений содержит модуль 204 обновлений клиентов, модуль 208 отчетов и пользовательский интерфейс 218 администрирования.

Модуль 204 обновлений клиентов является необязательным компонентом для корневого узла 300 службы обновлений в зависимости от того, предоставляет ли корневой узел службы обновлений обновления программного обеспечения компьютерам-клиентам напрямую. Например, согласно фиг.1, корневой узел 102 службы обновлений содержит необязательный модуль 204 обновлений клиентов, поскольку он является корневым узлом службы обновлений, который напрямую обслуживает компьютер-клиент 112. Если же корневой узел 300 службы обновлений непосредственно не обслуживает компьютеры-клиенты, то модуль 204 обновлений клиентов можно исключить.

Модуль 208 отчетов является необязательным компонентом для корневого узла 300 службы обновлений, поскольку корневой узел службы обновлений не имеет родительского узла службы обновлений, которому нужно предоставлять отчеты об обновлениях. Однако, в той степени, в какой отчеты об обновлениях требуются администратору корневого узла службы обновлений, модуль 208 отчетов может быть включен в необязательном порядке.

Помимо логических компонентов, входящих в состав узла 200 службы обновлений (фиг.2), корневой узел 300 службы обновлений содержит также интерфейс 302 поставщика программного обеспечения. Интерфейс 302 поставщика программного

обеспечения обеспечивает интерфейс связи, посредством которого поставщик 110 программного обеспечения (фиг.1) передает обновления программного обеспечения непосредственно на корневой узел 300 службы обновлений и опосредованно на иллюстративную систему 100 распространения обновлений.

Аналогично узлу 200 службы обновлений, показанному на фиг.2, вышеприведенное описание фиг.3 иллюстрирует различные компоненты иллюстративного корневого модуля 300 службы обновлений. Однако, следует заметить, что могут существовать и другие корневого модуля службы обновлений. Кроме того, вышеописанные компоненты следует рассматривать как логические компоненты, а не обязательно физические компоненты. В фактической реализации, вышеуказанные компоненты могут быть объединены друг с другом и/или с другими компонентами, отвечающими определениям реализации. Кроме того, следует понимать, что, хотя корневой узел 200 службы обновлений можно рассматривать как компьютер-сервер в сети, в фактической реализации, узел службы обновлений может быть реализован на вычислительном устройстве любого типа. Например, корневой узел 300 службы обновлений может быть реализован и/или установлен в единичной автономной компьютерной системе или, альтернативно, в распределенной вычислительной системе, содержащей множественные вычислительные устройства.

Чтобы можно было лучше понять, как обновление распространяется от корневого узла службы обновлений через систему 100 распространения обновлений, представлен пример обмена между родительским узлом службы обновлений и дочерним узлом службы обновлений. На фиг.4 показана блок-схема, иллюстрирующая обмен между родительским узлом 402 службы обновлений и дочерним узлом 404 службы обновлений при распространении обновления программного обеспечения от родительского узла службы обновлений к дочернему узлу службы обновлений согласно аспектам настоящего изобретения. Можно видеть, что иллюстративная схема 400 поделена пополам, причем левая половина соответствует действиям и событиям родительского узла 402 службы обновлений, а правая половина соответствует действиям и событиям дочернего узла 404 службы обновлений.

В целях рассмотрения фиг.4, следует заметить, что родительский узел 402 службы обновлений может быть или не быть корневым узлом службы обновлений в системе 100 распространения обновлений. Кроме того, в целях данного рассмотрения предположим, что родительский узел 402 службы обновлений сконфигурирован администратором так, что дочерний узел 404 службы обновлений может не принимать обновления программного обеспечения, пока не получит от администратора явное на то разрешение.

Как показано в иллюстративном обмене 400, начиная с события 406, родительский узел 402 службы обновлений принимает обновление программного обеспечения от поставщика 110 программного обеспечения либо напрямую, если родительский узел службы обновлений является корневым узлом 102 службы обновлений, либо косвенно, через систему 100 распространения обновлений. В какой-то момент после того, как родительский узел 402 службы обновлений получит обновление программного обеспечения от поставщика 110 программного обеспечения, дочерний узел 404 службы обновлений начинает процесс получения обновлений программного обеспечения с родительского узла службы обновлений.

Согласно одному варианту осуществления, дочерний узел 404 службы обновлений можно сконфигурировать так, чтобы он автоматически получал доступные обновления программного обеспечения от родительского узла 402 службы

обновлений на периодической основе. В частности, администратор, через пользовательский интерфейс 218 администрирования, может избирательно конфигурировать дочерний узел 404 службы обновлений, чтобы он автоматически получал последние обновления программного обеспечения, доступные на родительском узле 402 службы обновлений на периодической основе. Например, администратор может сконфигурировать дочерний узел 404 службы обновлений, чтобы он получал последние обновления программного обеспечения от своего родительского узла 402 службы обновлений ежедневно и/или ежечасно, а также задать время суток, когда должен совершаться процесс автоматического обновления. Можно также использовать другие периодические расписания и критерии. Аналогично, администратор может вручную инициировать процесс обновления через пользовательский интерфейс 218 администрирования.

Чтобы начать процесс обновления, в ходе события 408 дочерний узел 404 службы обновлений аутентифицирует и авторизует себя с помощью родительского узла 402 службы обновлений. Аутентификация и авторизация с помощью родительского узла 402 службы обновлений обеспечивает элемент контроля над распространением обновлений программного обеспечения, ограничивающий распространение обновлений авторизованными узлами службы обновлений. Методы аутентификации и авторизации общеизвестны в технике, и любые из них можно применять для аутентификации и авторизации дочернего узла 404 службы обновлений с помощью родительского узла 402 службы обновлений. Настоящее изобретение не ограничивается каким-либо одним методом.

После правильных аутентификации и авторизации с помощью родительского узла 402 службы обновлений, в ходе события 410 родительский узел 402 службы обновлений возвращает жетон авторизации дочернему узлу 404 службы обновлений. Согласно одному варианту осуществления, жетон авторизации является жетоном, чувствительным к времени, предоставляющим дочернему узлу 404 службы обновлений разрешение проводить дополнительные действия по обновлению с помощью родительского узла службы обновлений на ограниченный период времени. Таким образом, если дочерний узел 404 службы обновлений неправильно аутентифицирован и авторизован с помощью родительского узла службы обновлений, никакой жетон авторизации не возвращается, и дочерний узел службы обновлений не может осуществлять никакие другие действия по обновлению за исключением аутентификации и авторизации. Аналогично, по истечении времени жетона обновления, дочерний узел 404 службы обновлений не может осуществлять никакие другие действия по обновлению за исключением повторной аутентификации и повторной авторизации.

Получив жетон авторизации, в ходе события 412 дочерний узел 404 службы обновлений подает запрос родительскому узлу службы обновлений на предмет каталога обновлений продукта совместно с жетоном авторизации. Каталог обновлений продукта представляет список или таблицу содержимого программных продуктов, для которых родительский узел 402 службы обновлений распределяют обновления программного обеспечения.

Согласно аспектам настоящего изобретения, дочернему узлу 404 службы обновлений не требуется распространять все обновления программного обеспечения, имеющиеся на его родительском узле 402 службы обновлений. Например, со ссылкой на иллюстративную систему распространения обновлений, показанную на фиг.1, корпоративный узел 104 службы обновлений может иметь лицензии на пользование

системы только на часть программных продуктов, доступных на корневом узле 102 службы обновлений. Соответственно, корпоративному узлу 104 службы обновлений не требуется получать все обновления программного обеспечения, имеющиеся на корневом узле 102 службы обновлений, поскольку большинство из них никогда не будет использоваться. Соответственно, администратор на узле службы обновлений может избирательно устанавливать, какие обновления программных продуктов будут доступны на узле службы обновлений.

Согласно одному аспекту настоящего изобретения, каталог обновлений продуктов, полученный от родительского узла 402 службы обновлений, идентифицирует все программные продукты, для которых имеются обновления, независимо от того, настроен ли дочерний узел 404 службы обновлений на распространение обновлений для каждого продукта. Однако, согласно альтернативному аспекту настоящего изобретения, каталог обновлений продуктов, полученный от родительского узла 402 службы обновлений, идентифицирует только те программные продукты, для которых запрашивающий дочерний узел службы обновлений настроен на распространение обновлений. Например, ограничение, какие программные продукты перечислены в каталоге обновлений продуктов, можно определять в соответствии с группой или группами, которой(ым) принадлежит дочерний узел 404 службы обновлений.

В ходе события 414, родительский узел 402 службы обновлений возвращает каталог обновлений продуктов на дочерний узел 404 службы обновлений. В ходе события 416, дочерний узел 404 службы обновлений выбирает те продукты из каталога обновлений продуктов, обновление которых нужно в настоящее время. Заметим, что, хотя в каталоге обновлений продуктов могут быть перечислены только те программные продукты, которые распространяет дочерний узел 404 службы обновлений, причем дочерний узел службы обновлений может быть настроен на получение обновлений для разных программных продуктов в разное время или в соответствии с разными периодическими расписаниями.

В ходе события 418, дочерний узел 404 службы обновлений подает запрос синхронизации обновлений совместно с жетоном авторизации, идентифицируя выбранные продукты, для которых дочерний узел службы обновлений в данный момент ищет обновления. В запрос синхронизации входит информация, идентифицирующая самое последнее обновление, доступное для продукта на дочернем узле 404 службы обновлений. Информация, идентифицирующая последнее обновление продукта, будем называть «анкер обновления». Анкеры обновления для каждого программного продукта обычно хранятся в хранилище 216 информации обновлений (фиг.2). Согласно одному варианту осуществления, анкер обновления содержит номер ревизии и дату, связанную с номером ревизии.

В ответ на запрос синхронизации обновлений, в ходе события 420, родительский узел 402 службы обновлений определяет, какие, если таковые существуют, новые обновления доступны для дочернего узла 404 службы обновлений. Как отмечено выше, это определение базируется на конкретных правилах, связанных с конкретными обновлениями программного обеспечения и группой или группами, членом которой(ых) является дочерний узел 404 службы обновлений, а также анкер обновления. В этом примере, как было отмечено ранее, принятое ранее обновление программного обеспечения не было в явном виде авторизовано для дочернего узла 404 службы обновлений. Поэтому, обновление программного обеспечения, принятое в ходе события 406, не определено как «доступное» для дочернего узла 404 службы обновлений. Соответственно, в ходе события 422, список обновлений

возвращается дочернему узлу 404 службы обновлений без идентификации обновления программного обеспечения, принятого в ходе события 406. Согласно аспектам настоящего изобретения, в списке обновлений указаны все обновления, «доступные» на родительском узле 402 службы обновлений, согласно запросу синхронизации. Согласно одному варианту осуществления, список обновлений идентифицирует каждую информацию «доступного» обновления посредством уникального идентификатора, связанного с обновлением.

В ходе события 424, поскольку список обновлений пуст, то есть в данный момент на родительском узле 402 службы обновлений нет «доступных» обновлений, процесс обновления дочернего узла 404 службы обновлений просто задерживается или замирает на определенный период времени. Согласно данному примеру, в течение этого периода задержки, в ходе события 426, администратор на родительском узле 402 службы обновлений авторизует обновление программного обеспечения, принятое в ходе события 406, на распространение на дочерний узел 404 службы обновлений.

В ходе события 428 (фиг.4B), дочерний узел 404 службы обновлений вновь начинает процесс автоматического обновления, аутентифицируя и авторизуя себя с помощью родительского узла 402 службы обновлений. В ответ, в ходе события 430, родительский узел 402 службы обновлений возвращает жетон авторизации дочернему узлу 404 службы обновлений.

В ходе события 432, дочерний узел 404 службы обновлений подает запрос, совместно с жетоном авторизации, на родительский узел 402 службы обновлений. В ходе события 434, родительский узел 402 службы обновлений возвращает каталог обновлений продуктов дочернему узлу 404 службы обновлений. В ходе события 436, дочерний узел 404 службы обновлений выбирает продукты из каталога обновлений, для которых нужны обновления. В ходе события 438, дочерний узел 404 службы обновлений подает запрос синхронизации обновлений, идентифицирующий эти выбранные продукты с помощью жетона авторизации.

Поскольку дочернему узлу 404 службы обновлений уже разрешено получить обновление программного обеспечения, ранее принятое в ходе события 406, в ходе события 440, родительский узел 402 службы обновлений определяет, что обновление программного обеспечения «доступно» для дочернего узла 404 службы обновлений, и включает соответствующую информацию обновления в список обновлений. Затем, в ходе события 442, родительский узел 402 службы обновлений возвращает список обновлений, теперь идентифицирующий обновление программного обеспечения, принятое в ходе события 406, дочернему узлу 404 службы обновлений.

Благодаря списку обновлений, идентифицирующему «доступное» обновление на родительском узле 402 службы обновлений, дочерний узел 404 службы обновлений имеет информацию, необходимую для получения обновления программного обеспечения. Согласно вариантам осуществления настоящего изобретения, дочерний узел 404 службы обновлений получает обновление программного обеспечения от родительского узла 402 службы обновлений двумя частями: получает метаданные обновления и получает содержимое или файлы обновления, так называемую полезную нагрузку обновления. Как будет более подробно описано ниже, метаданные обновления описывают подходящие аспекты обновления программного обеспечения, в том числе, но не исключительно, идентификатор обновления, который уникально идентифицирует обновление, информацию номера ревизии, связанную с обновлением программного обеспечения, следует ли считать обновление программного обеспечения приоритетной, зависящей от языка информацией, соотношениями с

другими обновлениями программного обеспечения, местом полезной нагрузки обновления для целей загрузки, процедурами обработчика установки и т.д.

Некоторые из причин, по которым часто бывает выгодно загружать все обновление программного обеспечения двумя частями, то есть как метаданные и полезная нагрузка обновления, состоят в том, что полезная нагрузка обновления обычно значительно больше метаданных, и полезная нагрузка обновления не всегда нужна немедленно, то есть нужна для установки на компьютер-клиент, если вообще нужна. Таким образом, согласно одному варианту осуществления настоящего изобретения, полезная нагрузка обновления загружается отдельно от метаданных обновления и только при необходимости. Специалисты в данной области называют этот метод загрузки ленивой загрузкой или, альтернативно, загрузкой «точно вовремя». Согласно аспектам настоящего изобретения, администратор может сконфигурировать узел службы обновлений так, чтобы он получал полезную нагрузку обновления «точно вовремя» или же немедленно после получения метаданных обновления. Кроме того, согласно альтернативному варианту осуществления, метаданные обновления и полезная нагрузка обновления могут загружаться одновременно.

Согласно фиг.4В, для обновления, идентифицированного в списке обновлений, в ходе события 444, дочерний узел 404 службы обновлений запрашивает метаданные обновления для «доступного» обновления программного обеспечения согласно его уникальному идентификатору в списке обновлений. Как и для большинства других коммуникационных обменов с родительским узлом 402 службы обновлений, запрос обновления подается с жетоном авторизации. Заметим, что, хотя в показанном примере все метаданные обновлений загружаются в одном сеансе доступа, согласно альтернативным аспектам настоящего изобретения (не показаны), метаданные обновления можно загружать в течение более одного сеанса доступа. Например, в первом сеансе доступа, сначала загружают только элементы метаданных обновления, которые необходимы для определения, является ли обновление программного обеспечения применимым и/или желательным, например, правила применимости и зависимости от других обновлений программного обеспечения. Затем, после того, как определено, что обновление применимо и/или желательно, можно получить остальные метаданные. В ответ, в ходе события 446, родительский узел 402 службы обновлений возвращает метаданные обновления для обновления программного обеспечения дочернему узлу 404 службы обновлений, который, в свою очередь, сохраняет метаданные обновления в хранилище 216 информации обновления.

В необязательном порядке, в ходе события 448, дочерний узел 404 службы обновлений подает запрос на загрузку полезной нагрузки обновления с родительского узла 402 службы обновлений. В ответ, в ходе события 450, родительский узел 402 службы обновлений возвращает полезную нагрузку обновления дочернему узлу 404 службы обновлений, который, в свою очередь, сохраняет ее в хранилище 214 содержимого обновлений. Согласно альтернативному варианту осуществления, дочерний узел 404 службы обновлений загружает полезную нагрузку обновления из места хранения, которое может не являться родительским узлом 402 службы обновлений, из места, указанного метаданных обновления.

Поскольку деятельность по обновлению оказалась теперь на дочернем узле 404 службы обновлений, в ходе события 452 дочерний узел службы обновлений генерирует и подает отчет об обновлении на родительский узел 402 службы обновлений, в котором описаны действия по обновлению, которые только что были предприняты.

После этого, дочерний узел 404 службы обновлений снова задерживается до следующего раза, когда запланировано провести процесс обновления (не показано).

Специалистам в данной области известно, что вышеописанные события приведены только в иллюстративных целях и отражают один частный иллюстративный набор событий и обстоятельств. Очевидно, что могут происходить и другие события, соответствующие конкретным деталям и обстоятельствам, которые приведут к некоторым изменениям вышеописанных событий. Кроме того, следует понимать, что, хотя дочерний узел 404 службы обновлений получает, по меньшей мере, «доступные» обновления программного обеспечения от родительского узла 402 службы обновлений, дочерний узел службы обновлений может одновременно обрабатывать запросы обновления от своих дочерних узлов службы обновлений. Соответственно, вышеописанную последовательность событий нужно рассматривать только в иллюстративном порядке, а не в порядке ограничения настоящего изобретения.

На фиг.5 показана логическая блок-схема иллюстративной процедуры 500, выполняемой на дочернем узле службы обновлений, например, на корпоративном узле 104 службы обновлений, показанном на фиг.1, для периодического получения обновлений от своего родительского узла службы обновлений. Начиная с блока 502, дочерний узел службы обновлений получает синхронизированный список обновлений из «доступных» обновлений от родительского узла службы обновлений. Получение синхронизированного списка обновлений из «доступных» обновлений от родительского узла службы обновлений описано ниже со ссылкой на фиг.6.

На фиг.6 показана логическая блок-схема иллюстративной подпроцедуры 600, пригодной для использования в иллюстративной процедуре 500, изображенной на фиг.5, для получения синхронизированного списка обновлений из «доступных» обновлений от родительского узла службы обновлений. Начиная с блока 602, как было описано выше со ссылкой на фиг. 4A и 4B, дочерний узел службы обновлений аутентифицирует и авторизует себя с помощью родительского узла службы обновлений и, в случае правильной аутентификации и авторизации, принимает жетон авторизации. На блоке 604, с помощью жетона авторизации, дочерний узел службы обновлений устанавливает параметры связи с родительским узлом службы обновлений. Установление параметров связи позволяет родительскому и дочернему узлам службы обновлений правильно устанавливать общую основу, которую понимает как предок, так и потомок. Параметры связи включают в себя, но не только: протоколы или версии передачи обновлений; группировки продуктов и т.д.

Установив параметры связи с родительским узлом службы обновлений, дочерний узел службы обновлений, на блоке 606, получает каталог обновлений продуктов, описывающий программные продукты, для которых родительский узел службы обновлений обеспечивает/распространяет обновления. На блоке 608, дочерний узел службы обновлений выбирает те обновления программных продуктов, для которых в данный момент нужны обновления. На блоке 610 дочерний узел службы обновлений подает запрос синхронизации обновлений на родительский узел службы обновлений, содержащий жетон авторизации и «анкер», связанный с выбранными программными продуктами, идентифицирующий текущую ревизию и обновления уже на дочернем узле службы обновлений.

В ответ на запрос синхронизации обновлений, на блоке 612, дочерний узел службы обновлений получает список обновлений от родительского узла службы обновлений, синхронизированного согласно обновлениям программного обеспечения, «доступным» на родительском узле службы обновлений, в зависимости от того, что в

данный момент хранится на дочернем узле службы обновлений. Как было отмечено выше, список обновлений идентифицирует, посредством уникального идентификатора, те обновления программного обеспечения на родительском узле службы обновлений, которые «доступны» дочернему узлу службы обновлений. На этом, иллюстративная подпроцедура 600 заканчивается.

Возвращаясь к фиг.5, после получения синхронизированного списка обновлений от родительского узла службы обновлений, на блоке 504 принятия решения, производится определение, «доступны» ли какие-либо обновления программного обеспечения для загрузки с родительского узла службы обновлений. Это определение производится в зависимости от того, перечислены ли какие-либо идентификаторы обновлений в синхронизированном списке обновлений. В отсутствие обновлений программного обеспечения, «доступных» в настоящее время для загрузки, иллюстративная процедура 500 переходит к блоку 510 задержки, где иллюстративная процедура задерживается/спит, пока не начнется следующий период обновления. Альтернативно, при наличии обновлений, «доступных» для загрузки с родительского узла службы обновлений, на блоке 506, дочерний узел службы обновлений получает обновления от родительского узла службы обновлений. Получение «доступных» обновлений от родительского узла службы обновлений описано ниже со ссылкой на фиг.7.

На фиг.7 показана логическая блок-схема иллюстративной подпроцедуры 700, пригодной для использования в иллюстративной процедуре 500, изображенной на фиг.5, для получения «доступных» обновлений программного обеспечения от родительского узла службы обновлений. Начиная с блока 702, выбирают первый идентификатор обновления в списке обновлений. На блоке 704, дочерний узел службы обновлений получает метаданные обновления, соответствующие выбранному идентификатору обновления, от родительского узла службы обновлений и сохраняет их в хранилище 216 информации обновлений.

Согласно одному варианту осуществления, на блоке 706, дочерний узел службы обновлений получает полезную нагрузку обновления, соответствующую выбранному идентификатору обновления от родительского узла службы обновлений и сохраняет полезную нагрузку обновления в хранилище 212 содержимого обновлений. В необязательном порядке, содержимое обновления не требуется немедленно загружать в дочерний узел службы обновлений. Как отмечено выше, дочерний узел службы обновлений можно избирательно конфигурировать, чтобы загружать обновления от родительского узла службы обновлений «точно вовремя». Согласно этой необязательной обработке, как показано на фиг.7, вместо того, чтобы переходить от блока 704 к блоку 706, иллюстративная процедура 700 в необязательном порядке переходит от блока 704 к блоку 708.

На блоке 708 принятия решения, после получения метаданных обновления для выбранного идентификатора обновления и, в необязательном порядке, полезной загрузки обновления, производится определение, имеются ли какие-либо дополнительные идентификаторы обновления в списке обновлений. При наличии дополнительных идентификаторов обновлений, на блоке 710 выбирают следующий идентификатор обновления в списке обновления, и подпроцедура 700 возвращается к блоку 704 для дополнительной обработки. Процедура 700 продолжается до тех пор, пока на блоке 708 принятия решения не будет определено, что в списке обновлений не осталось идентификаторов обновлений, вследствие чего иллюстративная подпроцедура 700 заканчивается.

Вновь возвращаясь к фиг.5, получив «доступные» обновления от родительского узла службы обновлений, на блоке 508 дочерний узел службы обновлений сообщает о действиях по обновлению родительскому узлу службы обновлений. Затем, на блоке 510 задержки, иллюстративная процедура 500 задерживается/засыпает на определенный период времени до следующего периода обновления, после чего переходит к блоку 502, чтобы повторить вышеописанные операции обновления.

Согласно фиг.5, на блоке 504 принятия решения, даже в отсутствие «доступных» обновлений на родительском узле службы обновлений, дочерний узел службы обновлений можно, в необязательном порядке, сконфигурировать так, чтобы он сообщал о своих действиях по обновлению родительскому узлу службы обновлений. Согласно этой альтернативной конфигурации, в отсутствие доступных обновлений, иллюстративная процедура 500 может перейти к блоку 508 для сообщения о действиях по обновлению.

На фиг.8 показана логическая блок-схема иллюстративной процедуры 800, реализованной на родительском узле службы обновлений для генерации синхронизированного списка обновлений, идентифицирующего «доступные» обновления в ответ на запрос синхронизации обновлений от дочернего узла службы обновлений. Начиная с блока 802, родительский узел службы обновлений принимает запрос синхронизации обновлений от дочернего узла службы обновлений для списка обновлений, идентифицирующего «доступные» обновления. На блоке 804, выбирают первый программный продукт, идентифицированный в запросе синхронизации обновления.

На блоке 806 принятия решения, производится определение, имеются ли какие-либо доступные обновления для идентифицированного программного продукта. Это определение производится в соответствии с метаданными для программного продукта, хранящимися в хранилище 216 информации обновлений, в соответствии с анкером обновлений, обеспеченным дочерним узлом службы обновлений, и в соответствии с правилами распространения, связанными с группой, которой принадлежит дочерний узел службы обновлений. В соответствии с этим определением, при наличии «доступных» обновлений, на блоке 808, уникальные идентификаторы обновлений, связанные с «доступными» обновлениями, записываются в список обновлений. После записи уникальных идентификаторов обновлений для «доступных» обновлений в список обновлений, на блоке 810 принятия решений, производится определение, имеются ли какие-либо еще программные продукты, идентифицированные в запросе синхронизации обновлений. При наличии дополнительных обновлений программных продуктов в запросе синхронизации обновлений, на блоке 814, родительский узел службы обновлений выбирает следующий программный продукт, идентифицированный в запросе синхронизации обновлений, и возвращается к блоку 806 принятия решений для определения наличия «доступных» обновлений для выбранного программного продукта. Альтернативно, в отсутствие дополнительных программных продуктов, идентифицированных в запросе синхронизации обновлений, на блоке 814, список обновлений возвращается дочернему узлу службы обновлений. Затем, иллюстративная подпроцедура 800 заканчивается.

Согласно вышеописанному, метаданные обновления описывают подходящие аспекты обновления программного обеспечения. Фактически, многие из этих аспектов столь же важны для обновления программного обеспечения, как и сама полезная нагрузка обновления. Например, если метаданные обновления для первого обновления программного обеспечения указывают, что в качестве предварительного

условия должно быть установлено более раннее обновление программного обеспечения, и что в настоящее время более раннее обновление программного обеспечения не установлено согласно решениям администратора, не будет никакого смысла загружать полезную нагрузку обновления для первого обновления программного обеспечения. Таким образом, благодаря отказу от загрузки полезной нагрузки обновления для первого обновления программного обеспечения, можно значительно сэкономить пропускную способность канала связи. Конечно, специалистам в данной области понятно, что это лишь один пример того, как метаданные обновления обеспечивают важную и актуальную информацию, относящуюся к обновлению программного обеспечения.

Хотя метаданные обновления можно передавать в файле любого формата, согласно аспектам настоящего изобретения, метаданные обновления, соответствующие обновлению программного обеспечения, описаны и содержатся в тэгированном файле, например, файле расширяемого языка разметки (XML). Можно использовать любой подходящий формат тэгового файла, который допускает настраиваемые тэги, описывающие и содержащие данные. Тэгированные файлы, например, файлы на основе XML или гипертекстового языка разметки (HTML), для описания и вмещения информации общеизвестны в технике. Кроме того, как известно в технике, содержимое тэгового файла XML часто определяют согласно файлу определений, именуемому схемой. Эти метаданные обновления используются как дочерними узлами службы обновлений, так и компьютерами-клиентами.

Согласно фактическому варианту осуществления настоящего изобретения, метаданные обновления для обновления программного обеспечения содержатся в файле XML, согласующемся со схемой метаданных обновления. На фиг.9 показана блок-схема, иллюстрирующая фрагменты иллюстративной схемы 900 метаданных обновления на основе XML, задающей содержимое файла метаданных обновления. В частности, иллюстративная схема 900 метаданных задает Update 902, содержащее следующие элементы: элемент 904 UpdateIdentity; элементы 906 Properties; элементы 908 LocalizedPropertiesCollection; элементы 910 Relationships; элементы 912 ApplicabilityRules; элементы 914 Files; и элементы 916 HandlerSpecificData. Что касается элементов 906-916, подходящий файл метаданных обновления может включать в себя нуль или более этих элементов, согласно тэгу-описателю "minOccurs="0" 918. Напротив, файл метаданных обновления должен включать в себя один элемент 902 UpdateIdentity. Согласно показанной схеме 900 метаданных обновления, каждый элемент обновления в файле метаданных обновления, если таковой существует, должен появляться в вышеописанном порядке.

Согласно вышеописанному, каждому обновлению программного обеспечения присваивается уникальный идентификатор. Согласно аспектам настоящего изобретения, элемент 902 UpdateIdentity содержит этот уникальный идентификатор, а также другую информацию для идентификации обновления программного обеспечения, например, номер ревизии, также связанный с обновлением программного обеспечения.

Элемент 904 Properties обеспечивает информацию, относящуюся к обновлению программного обеспечения, в том числе, но не исключительно, язык, то есть английский, испанский, французский и т.д., в отношении которого локализовано обновление программного обеспечения; информация, подсказывающая влияние, которое обновление, скорее всего, окажет на вычислительную систему в ходе установки, например, минимальное, сильное, может потребовать или потребует

перезагрузки вычислительной системы и т.п.; тип программного обеспечения, подлежащего обновлению посредством обновления программного обеспечения, например, системный драйвер или прикладная программа; рейтинг важности, соответствующий важности обновления программного обеспечения для поставщика программного обеспечения; соответствующий бюллетень безопасности от поставщика программного обеспечения; и идентификацию обработчика обновлений для идентификации обработчика обновлений для обновления программного обеспечения. Как можно видеть из вышеперечисленных различных типов свойств, в файле метаданных может быть нуль или более элементов 904 Properties.

Элемент 906 LocalizedPropertiesCollection обеспечивает информацию, связанную с языком, относящуюся к обновлению программного обеспечения, например, но не исключительно: название обновления; описание обновления программного обеспечения, предназначенное для отображения пользователю компьютера; примечания по выпуску от поставщика программного обеспечения; лицензионные соглашения конечного пользователя и связанную с этим информацию; инструкции для пользователя по удалению обновления программного обеспечения; и т.д. Согласно аспектам настоящего изобретения, локализованные свойства группируются в соответствии с языком. Например, хотя элемент 906 LocalizedPropertiesCollection может содержать множественные фрагменты информации, английские версии вышеозначенной информации будут группироваться друг с другом и испанские версии будут группироваться аналогично. В файле метаданных обновления может быть нуль или более элементов 906 LocalizedPropertiesCollection, как и вышеописанных элементов 904 Properties.

Элемент 908 Relationships обеспечивает информацию, касающуюся соотношений между данным обновлением программного обеспечения и другими обновлениями программного обеспечения. Примеры этих соотношений включают в себя, но без ограничения, заранее необходимые обновления программного обеспечения, опережающие обновления программного обеспечения и связанные обновления программного обеспечения. Соотношение заранее необходимого обновления программного обеспечения указывает, что другое обновление программного обеспечения, идентифицируемое своим уникальным идентификатором, должно быть установлено в клиентской компьютерной системе до установки данного обновления программного обеспечения. Множественные соотношения предварительного условия могут быть связаны посредством булевых операторов, например логических операторов И и ИЛИ, в логические выражения, что позволяет, оценив логические выражения, определить пригодность обновления программного обеспечения для установки на компьютер-клиент.

Связанное обновление программного обеспечения идентифицирует совокупность обновлений программного обеспечения, которые нужно устанавливать совместно. Например, связанное приложение может указывать взаимозависимость между данным обновлением программного обеспечения и другими обновлениями программного обеспечения, так что все обновления, идентифицированные в связке, идентифицированные своим уникальным идентификатором обновления, должны быть установлены, если какие-либо из них установлены в компьютерной системе. Как и вышеупомянутые заранее необходимые обновления программного обеспечения, элементы связанного обновления программного обеспечения можно связывать посредством булевых операторов для формирования логических выражений, позволяющих оценивать пригодность связанного обновления программного

обеспечения для установки на компьютер-клиент. Напротив, опережающее обновление программного обеспечения идентифицирует другие обновления программного обеспечения, которые были установлены в опережение данному обновлению программного обеспечения. Как и вышеописанный элемент 904 Properties, в файле метаданных обновления может быть нуль или более элементов 908 Relationships.

Элемент 912 ApplicabilityRules обеспечивает правила или тесты для определения применимости и/или пригодности обновления программного обеспечения для установки в вычислительной системе. Хотя элемент 912 ApplicabilityRules в некоторых отношениях подобен элементу 908 Relationships, он проверяет условия в вычислительной системе, которые могут быть конкретно связаны или не связаны с другим обновлением программного обеспечения. Как и другие вышеозначенные элементы, в файле метаданных обновления может быть нуль или более элементов 912 ApplicabilityRules.

Элемент 914 Files идентифицирует файлы, то есть полезную нагрузку обновления, связанную с обновлением программного обеспечения, а также информацию, относящуюся к этим файлам. Эта дополнительная информация включает в себя, но без ограничения, имеется ли один или много файлов в полезной нагрузке обновления; место, откуда можно получить файлы; размер файлов; имя файла; дату создания файла и т.п. Специалистам в данной области известно, что единичное обновление программного обеспечения можно установить на компьютер-клиент разными способами. Например, одно и то же обновление программного обеспечения можно установить, изменив фрагменты существующих файлов с помощью заплатки, или, альтернативно, просто заменив существующие файлы более новыми версиями. Таким образом, согласно аспектам настоящего изобретения, полезная нагрузка обновления в элементе 914 Files может содержать или ссылаться на: заплатку для существующих файлов, файлы замены или и заплатку, и файлы замены. В файле метаданных обновления может быть нуль или более элементов 914 Files.

Обновления могут быть обеспечены в различных форматах, например, в виде дельта-заплаток, которые описывают области файла, подлежащие перезаписи информацией дельта-заплатки, выполнимых файлов, файлов замены и т.п. Для каждого из этих типов обновлений требуется особый обработчик обновлений для выполнения обновления программного обеспечения в вычислительной системе. Соответственно, элемент 916 HandlerSpecificData обеспечивает место в файле метаданных обновления для включения данных/информации, зависящих от обработчика. Например, эта информация может включать в себя, но без ограничения, директорию, в которой должен выполняться обработчик, аргументы командной строки для передачи обработчику, действия, которые нужно предпринимать в случае сбоя некоторых аспектов установки, действия, которые нужно предпринимать в случае успешной установки, и т.д. В файле метаданных обновления может быть нуль или более элементов 916 HandlerSpecificData.

Хотя были проиллюстрированы и описаны различные варианты осуществления и аспекты настоящего изобретения, в том числе предпочтительный вариант осуществления, очевидно, что можно предложить различные изменения, не выходящие за рамки сущности и объема изобретения. Например, хотя настоящее изобретение было описано применительно к доставке обновлений программного обеспечения через систему распространения обновлений по сети связи, метаданные обновлений, а также полезную нагрузку обновлений, можно доставлять с узла службы обновлений

на компьютер-клиент на компьютерно-считываемом носителе, например, компакт-диске или флоппи-диске.

Формула изобретения

1. Реализуемый на компьютере способ для передачи метаданных обновления, соответствующих обновлению программного обеспечения, на компьютер-клиент, при этом способ содержит этапы, на которых:

принимают от компьютера-клиента синхронизирующий запрос информации, относящейся обновлению программного обеспечения, соответствующего программному продукту,

в ответ на упомянутый синхронизирующий запрос определяют, доступно ли обновление программного обеспечения для упомянутого программного продукта,

генерируют тэгированную структуру данных, содержащую метаданные, соответствующие обновлению программного обеспечения доступного для установки на компьютер клиент, при этом упомянутая тэгированная структура данных содержит

тегированный элемент UpdateIdentity [идентификатор обновления],

содержащий метаданные для уникальной идентификации обновления программного обеспечения,

элемент Relationships [соотношения], содержащий метаданные, идентифицирующие соотношения между обновлением программного обеспечения и другими обновлениями программного обеспечения, причем элемент Relationships [соотношения] содержит совокупность подэлементов предварительного условия, идентифицирующих совокупность обновлений программного обеспечения, которые нужно устанавливать совместно, причем упомянутые обновления программного обеспечения упомянутой совокупности связаны друг с другом посредством булевых операторов в логическое выражение, так что оценка логического выражения определяет пригодность обновления программного обеспечения для установки на компьютер-клиент, и по меньшей мере один дополнительный элемент из нижеперечисленной группы элементов:

элемент Properties [свойства], содержащий метаданные, идентифицирующие общие свойства, относящиеся к обновлению программного обеспечения, включая информацию обработчика обновления, идентифицирующую обработчик обновления для установки упомянутого идентифицированного обновления программного обеспечения на компьютер-клиент,

элемент LocalizedPropertiesCollection [совокупность локализованных свойств], содержащий метаданные, идентифицирующие относящуюся к языку информацию, ориентированную на пользователя компьютера, относящегося к обновлению программного обеспечения,

элемент ApplicabilityRules [правила применимости], содержащий метаданные, идентифицирующие правила для определения применимости обновления программного обеспечения к компьютеру-клиенту,

элемент Files [файлы], содержащий метаданные, идентифицирующие информацию, относящуюся к полезной нагрузке обновления программного обеспечения, и

элемент HandlerSpecificData [данные, зависящие от обработчика], содержащий метаданные, идентифицирующие информацию для исполнения обработчика обновления, идентифицированного в упомянутом элементе свойства, для установки идентифицированного обновления программного обеспечения на компьютер-клиент; и

предоставление упомянутой тэгированной структуры данных в упомянутый

компьютер-клиент.

2. Способ по п.1, в котором тэгированная структура данных является структурой данных XML.

3. Способ по п.1, в котором элементы тэгированной структуры данных, если имеются, включены в метаданные обновления в вышеописанном порядке.

4. Способ по п.1, в котором элемент Updateldentity [идентификатор обновления] содержит подэлемент уникального идентификатора, который уникально идентифицирует обновление программного обеспечения, и подэлемент номера ревизии, идентифицирующий номер ревизии, связанный с обновлением программного обеспечения.

5. Способ по п.1, в котором элемент Relationships [соотношения] содержит подэлемент предварительного условия, который идентифицирует второе обновление программного обеспечения, которое должно быть установлено до установки обновления программного обеспечения.

6. Способ по п.5, в котором элемент Relationships [соотношения] содержит совокупность подэлементов предварительного условия, идентифицирующих совокупность обновлений программного обеспечения, связанных друг с другом посредством булевых операторов в логическое выражение, так что оценка логического выражения определяет пригодность обновления программного обеспечения для установки на компьютер-клиент.

7. Способ по п.1, в котором элемент Relationships [соотношения] содержит подэлемент опережения, который идентифицирует, по меньшей мере, одно другое обновление программного обеспечения, которое установлено в опережение обновления программного обеспечения.

8. Способ по п.1, в котором элемент Relationships [соотношения] содержит любое количество подэлементов предварительного условия, которые идентифицируют другие обновления программного обеспечения, которые должны быть установлены до установки обновления программного обеспечения, подэлементов связки, которые идентифицируют совокупность обновлений программного обеспечения, которые нужно устанавливать совместно, и подэлементов опережения, которые идентифицируют, по меньшей мере, одно другое обновление программного обеспечения, которое установлено в опережение обновления программного обеспечения.

9. Способ по п.1, в котором элемент Files [файлы] содержит информацию, идентифицирующую полезную нагрузку обновления программного обеспечения для установки заплаток на существующие файлы на компьютере-клиенте.

10. Способ по п.1, в котором элемент Files [файлы] содержит информацию, идентифицирующую полезную нагрузку обновления программного обеспечения для замены существующих файлов на компьютере-клиенте.

11. Способ по п.12, в котором элемент Files [файлы] содержит информацию, идентифицирующую полезную нагрузку обновления программного обеспечения как для установки заплаток на существующие файлы на компьютере-клиенте, так и для замены существующих файлов на компьютере-клиенте.

12. Реализуемый на компьютере способ для передачи метаданных обновления, соответствующих обновлению программного обеспечения, на компьютер-клиент, при этом способ содержит этапы, на которых:

принимают от компьютера-клиента синхронизирующий запрос информации, относящейся обновлению программного обеспечения, соответствующего

программному продукту,

в ответ на упомянутый синхронизирующий запрос определяют, доступно ли обновление программного обеспечения для упомянутого программного продукта,

генерируют тэгированную структуру данных, содержащую метаданные, соответствующие обновлению программного обеспечения, доступного для установки на компьютер клиент, причем тэгированные элементы упомянутой структуры данных являются текстовыми элементами, и упомянутая тэгированная структура данных содержит:

элемент Updateldentity [идентификатор обновления], используемый для уникальной идентификации обновления программного обеспечения,

элемент Relationships [соотношения] для сохранения соотношений между обновлением программного обеспечения и другими обновлениями программного обеспечения, причем элемент Relationships [соотношения] содержит совокупность подэлементов предварительного условия, идентифицирующих совокупность обновлений программного обеспечения, которые нужно устанавливать совместно, причем упомянутые обновления программного обеспечения упомянутой совокупности связаны друг с другом посредством булевых операторов в логическое выражение, так что оценка логического выражения определяет пригодность обновления программного обеспечения для установки на компьютер-клиент, и

по меньшей мере один дополнительный элемент из нижеперечисленной группы элементов:

элемент Properties [свойства], содержащий общие свойства, относящиеся к обновлению программного обеспечения, включая информацию обработчика обновления, идентифицирующую обработчик обновления для установки упомянутого идентифицированного обновления программного обеспечения на компьютер-клиент,

элемент Files [файлы], идентифицирующий полезную нагрузку упомянутого идентифицированного обновления программного обеспечения; и

элемент HandlerSpecificData [данные, зависящие от обработчика], содержащий метаданные, идентифицирующие информацию для исполнения обработчика обновления, идентифицированного в упомянутом элементе свойства, для установки идентифицированного обновления программного обеспечения на компьютер-клиент; и

предоставление упомянутой тэгированной структуры данных в упомянутый компьютер клиент.

13. Способ по п.12, в котором тэгированная структура данных является структурой данных XML.

14. Способ по п.12, в котором элемент Updateldentity [идентификатор обновления] содержит подэлемент уникального идентификатора, который уникально идентифицирует обновление программного обеспечения, и подэлемент номера ревизии, идентифицирующий номер ревизии, связанный с обновлением программного обеспечения.

15. Способ по п.14, в котором элемент Relationships [соотношения] содержит подэлемент предварительного условия, который идентифицирует второе обновление программного обеспечения, которое должно быть установлено до установки обновления программного обеспечения.

16. Способ по п.15, в котором элемент Relationships [соотношения] содержит совокупность подэлементов предварительного условия, идентифицирующих совокупность обновлений программного обеспечения, связанных друг с другом посредством булевых операторов в логическое выражение, так что оценка

логического выражения определяет пригодность обновления программного обеспечения для установки на компьютер-клиент.

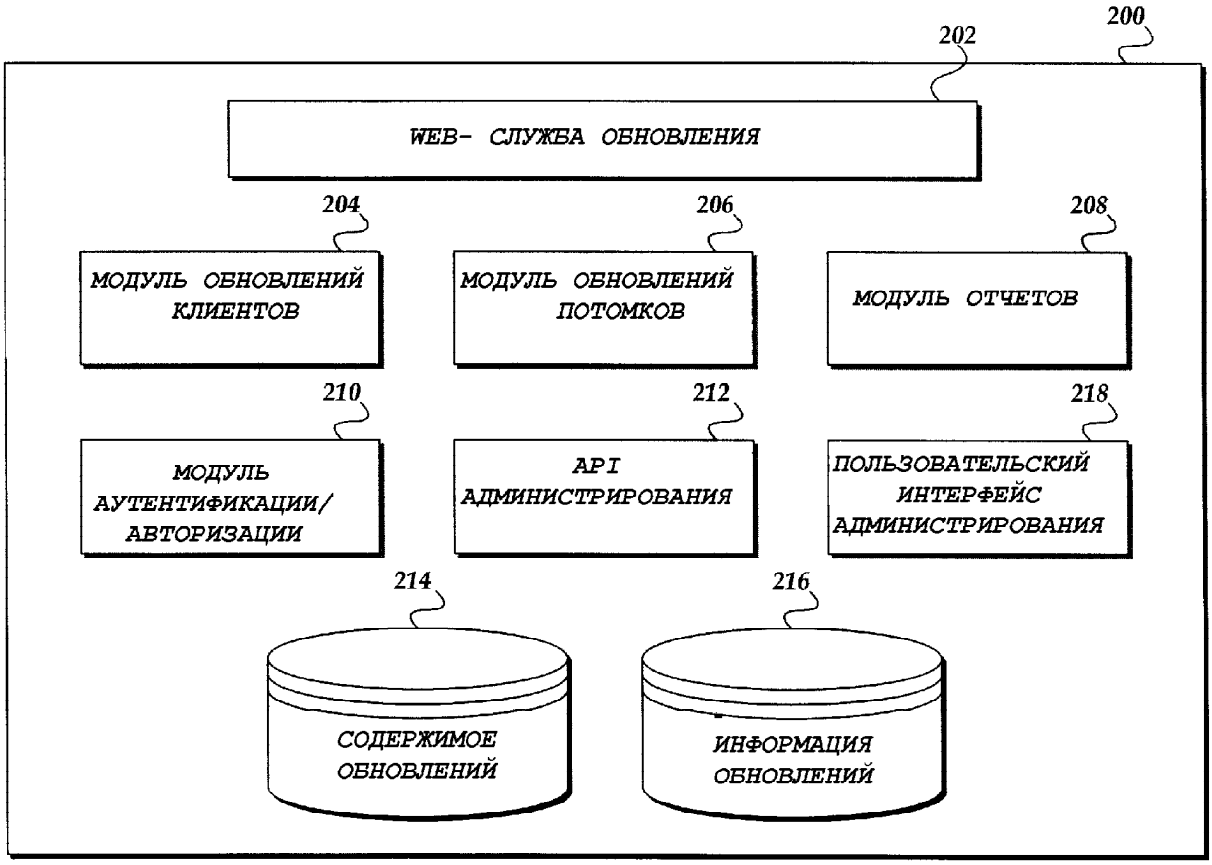
17. Способ по п.14, в котором элемент Relationships [Соотношения] содержит подэлемент опережения, который идентифицирует, по меньшей мере, одно другое обновление программного обеспечения, которое установлено в опережение обновления программного обеспечения.

18. Способ по п.14, в котором элемент Relationships [Соотношения] содержит любое количество подэлементов предварительного условия, которые идентифицируют другие обновления программного обеспечения, которые должны быть установлены до установки обновления программного обеспечения, подэлементов связки, которые идентифицируют совокупность обновлений программного обеспечения, которые нужно устанавливать совместно, и подэлементов опережения, которые идентифицируют, по меньшей мере, одно другое обновление программного обеспечения, которое установлено в опережение обновления программного обеспечения.

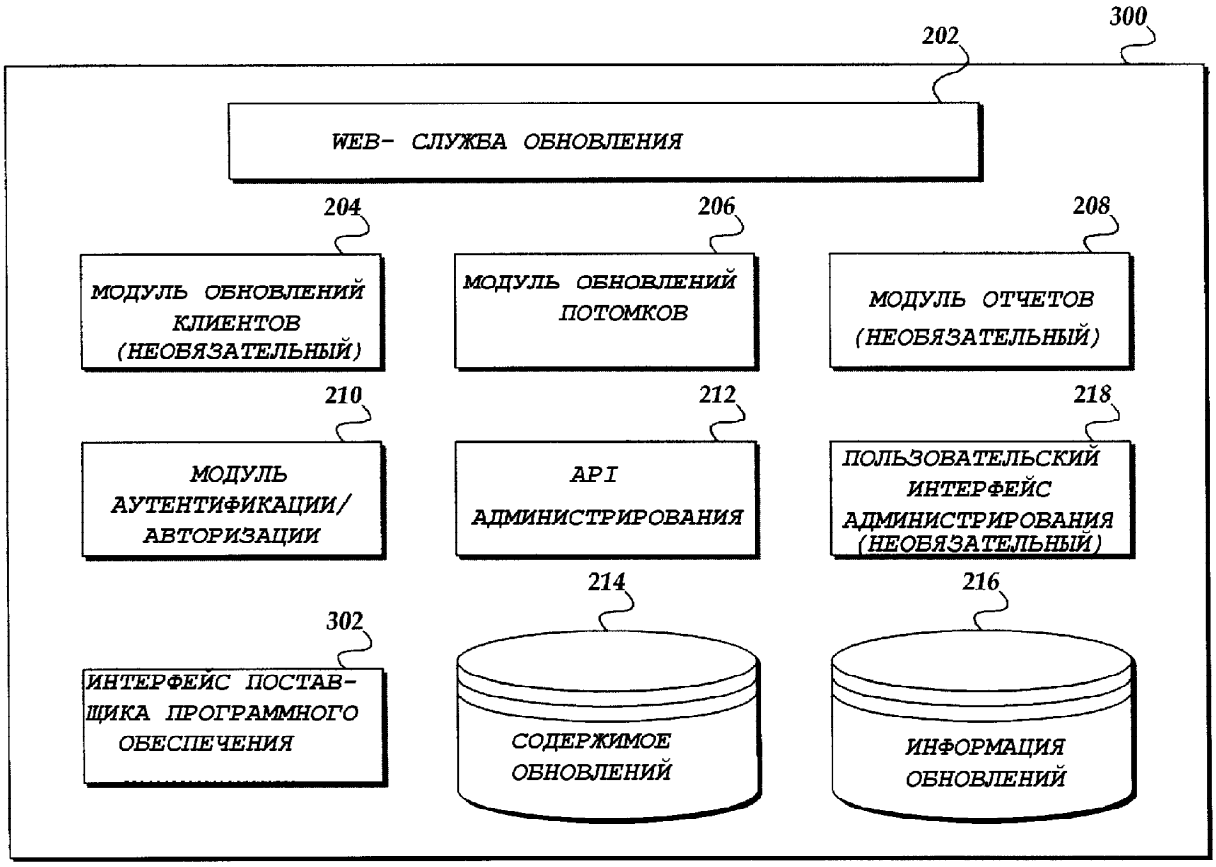
19. Способ по п.14, в котором элемент Files [файлы] содержит информацию, идентифицирующую полезную нагрузку обновления программного обеспечения для установки заплаток на существующие файлы на компьютере-клиенте.

20. Способ по п.14, в котором элемент Files [файлы] содержит информацию, идентифицирующую полезную нагрузку обновления программного обеспечения для замены существующих файлов на компьютере-клиенте.

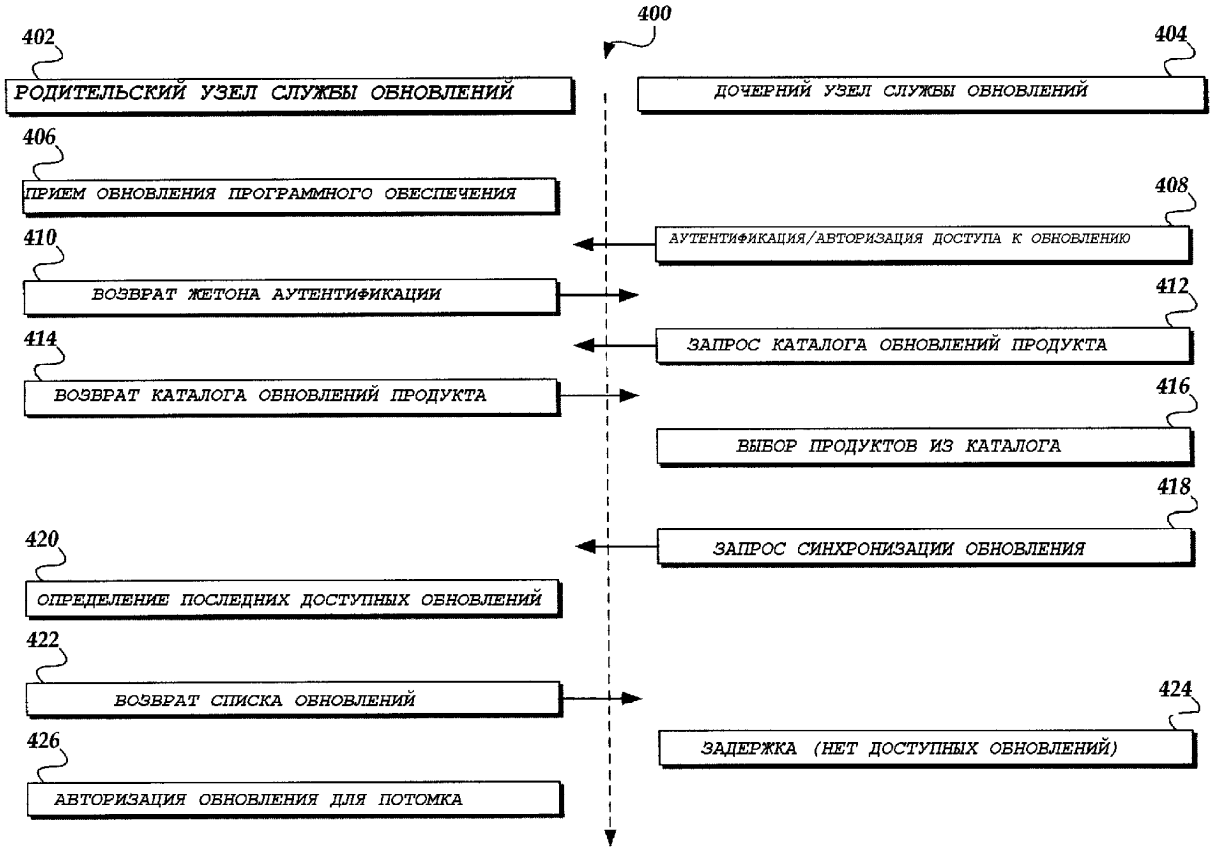
21. Способ по п.20, в котором элемент Files [файлы] содержит информацию, идентифицирующую полезную нагрузку обновления программного обеспечения как для установки заплаток на существующие файлы на компьютере-клиенте, так и для замены существующих файлов на компьютере-клиенте.


200
202
WEB- СЛУЖБА ОБНОВЛЕНИЯ
204
МОДУЛЬ ОБНОВЛЕНИЙ КЛИЕНТОВ
206
МОДУЛЬ ОБНОВЛЕНИЙ ПОТОМКОВ
208
МОДУЛЬ ОТЧЕТОВ
210
МОДУЛЬ АУТЕНТИФИКАЦИИ/ АВТОРИЗАЦИИ
212
API АДМИНИСТРИРОВАНИЯ
218
ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС АДМИНИСТРИРОВАНИЯ
214
СОДЕРЖИМОЕ ОБНОВЛЕНИЙ
216
ИНФОРМАЦИЯ ОБНОВЛЕНИЙ

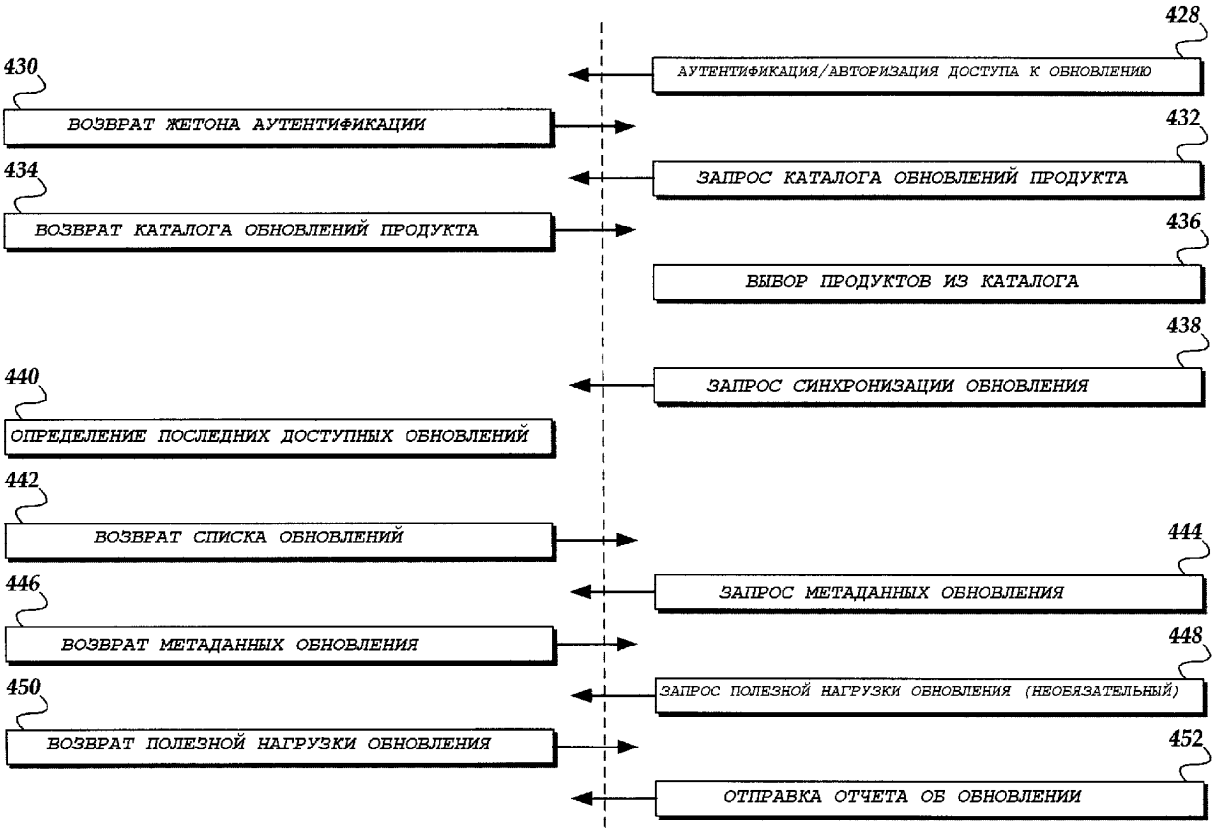
ФИГ. 2


300
202
WEB- СЛУЖБА ОБНОВЛЕНИЯ
204
МОДУЛЬ ОБНОВЛЕНИЙ КЛИЕНТОВ (НЕОБЯЗАТЕЛЬНЫЙ)
206
МОДУЛЬ ОБНОВЛЕНИЙ ПОТОМКОВ
208
МОДУЛЬ ОТЧЕТОВ (НЕОБЯЗАТЕЛЬНЫЙ)
210
МОДУЛЬ АУТЕНТИФИКАЦИИ/ АВТОРИЗАЦИИ
212
API АДМИНИСТРИРОВАНИЯ
218
ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС АДМИНИСТРИРОВАНИЯ (НЕОБЯЗАТЕЛЬНЫЙ)
214
216
302
ИНТЕРФЕЙС ПОСТАВЩИКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
СОДЕРЖИМОЕ ОБНОВЛЕНИЙ
ИНФОРМАЦИЯ ОБНОВЛЕНИЙ

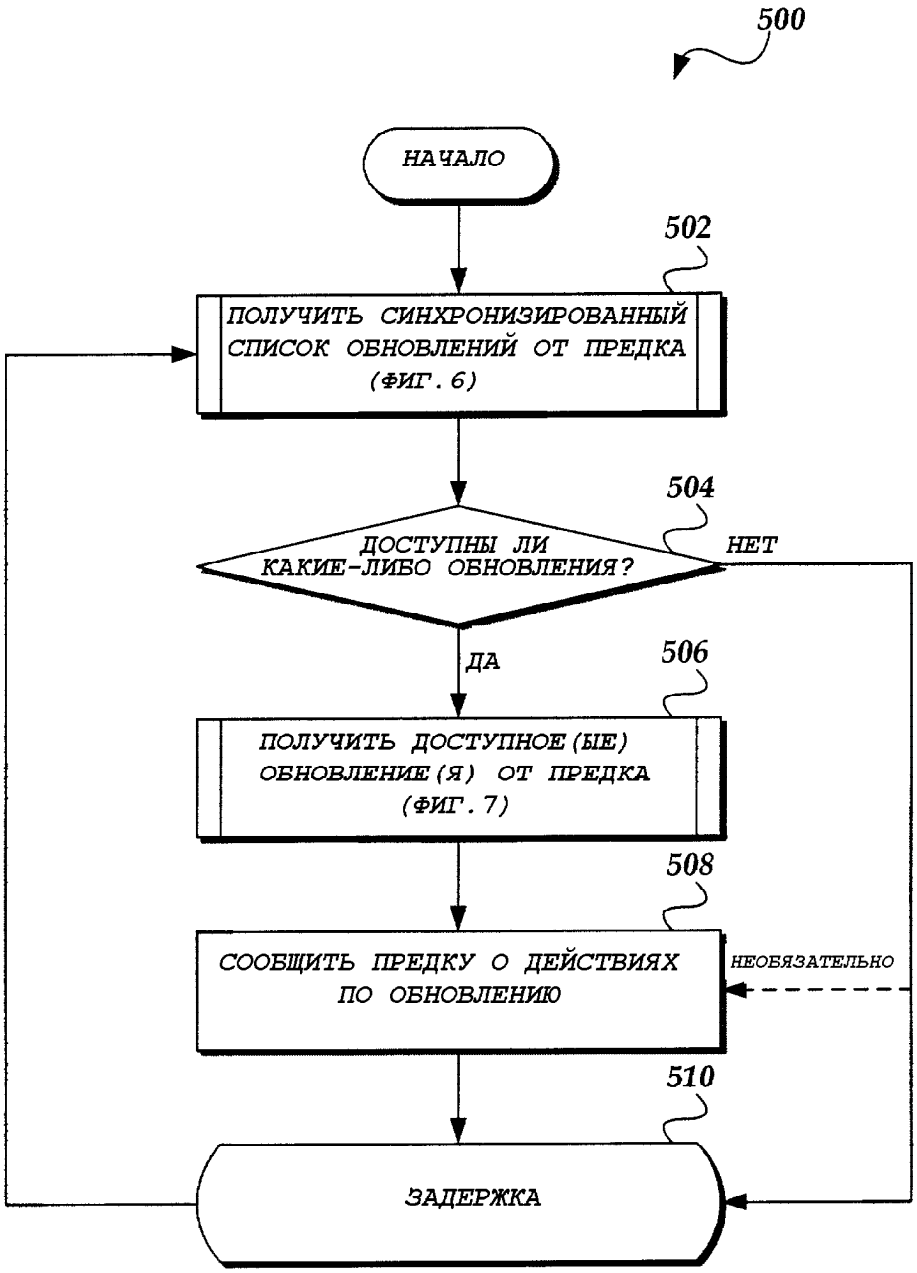
ФИГ. 3


400
402
РОДИТЕЛЬСКИЙ УЗЕЛ СЛУЖБЫ ОБНОВЛЕНИЙ
404
ДОЧЕРНИЙ УЗЕЛ СЛУЖБЫ ОБНОВЛЕНИЙ
406
ПРИЕМ ОБНОВЛЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
408
АУТЕНТИФИКАЦИЯ/АВТОРИЗАЦИЯ ДОСТУПА К ОБНОВЛЕНИЮ
410
ВОЗВРАТ ЖЕТОНА АУТЕНТИФИКАЦИИ
412
ЗАПРОС КАТАЛОГА ОБНОВЛЕНИЙ ПРОДУКТА
414
ВОЗВРАТ КАТАЛОГА ОБНОВЛЕНИЙ ПРОДУКТА
416
ВЫБОР ПРОДУКТОВ ИЗ КАТАЛОГА
418
ЗАПРОС СИНХРОНИЗАЦИИ ОБНОВЛЕНИЯ
420
ОПРЕДЕЛЕНИЕ ПОСЛЕДНИХ ДОСТУПНЫХ ОБНОВЛЕНИЙ
422
ВОЗВРАТ СПИСКА ОБНОВЛЕНИЙ
424
ЗАДЕРЖКА (НЕТ ДОСТУПНЫХ ОБНОВЛЕНИЙ)
426
АВТОРИЗАЦИЯ ОБНОВЛЕНИЯ ДЛЯ ПОТОМКА

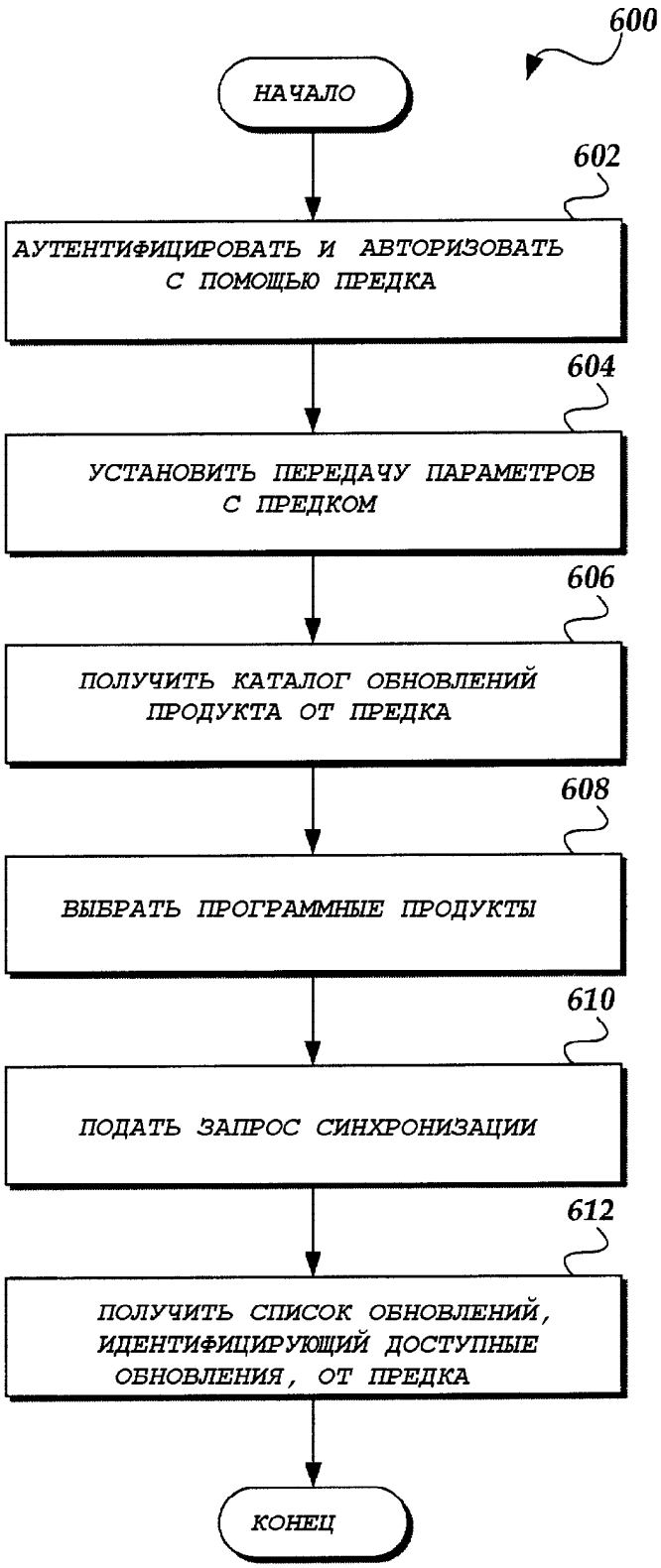
ФИГ. 4A


428
АУТЕНТИФИКАЦИЯ/АВТОРИЗАЦИЯ ДОСТУПА К ОБНОВЛЕНИЮ
430
ВОЗВРАТ ЖЕТОНА АУТЕНТИФИКАЦИИ
432
ЗАПРОС КАТАЛОГА ОБНОВЛЕНИЙ ПРОДУКТА
434
ВОЗВРАТ КАТАЛОГА ОБНОВЛЕНИЙ ПРОДУКТА
436
ВЫБОР ПРОДУКТОВ ИЗ КАТАЛОГА
438
ЗАПРОС СИНХРОНИЗАЦИИ ОБНОВЛЕНИЯ
440
ОПРЕДЕЛЕНИЕ ПОСЛЕДНИХ ДОСТУПНЫХ ОБНОВЛЕНИЙ
442
ВОЗВРАТ СПИСКА ОБНОВЛЕНИЙ
444
ЗАПРОС МЕТАДАННЫХ ОБНОВЛЕНИЯ
446
ВОЗВРАТ МЕТАДАННЫХ ОБНОВЛЕНИЯ
448
ЗАПРОС ПОЛЕЗНОЙ НАГРУЗКИ ОБНОВЛЕНИЯ (НЕОБЯЗАТЕЛЬНЫЙ)
450
ВОЗВРАТ ПОЛЕЗНОЙ НАГРУЗКИ ОБНОВЛЕНИЯ
452
ОТПРАВКА ОТЧЕТА ОБ ОБНОВЛЕНИИ

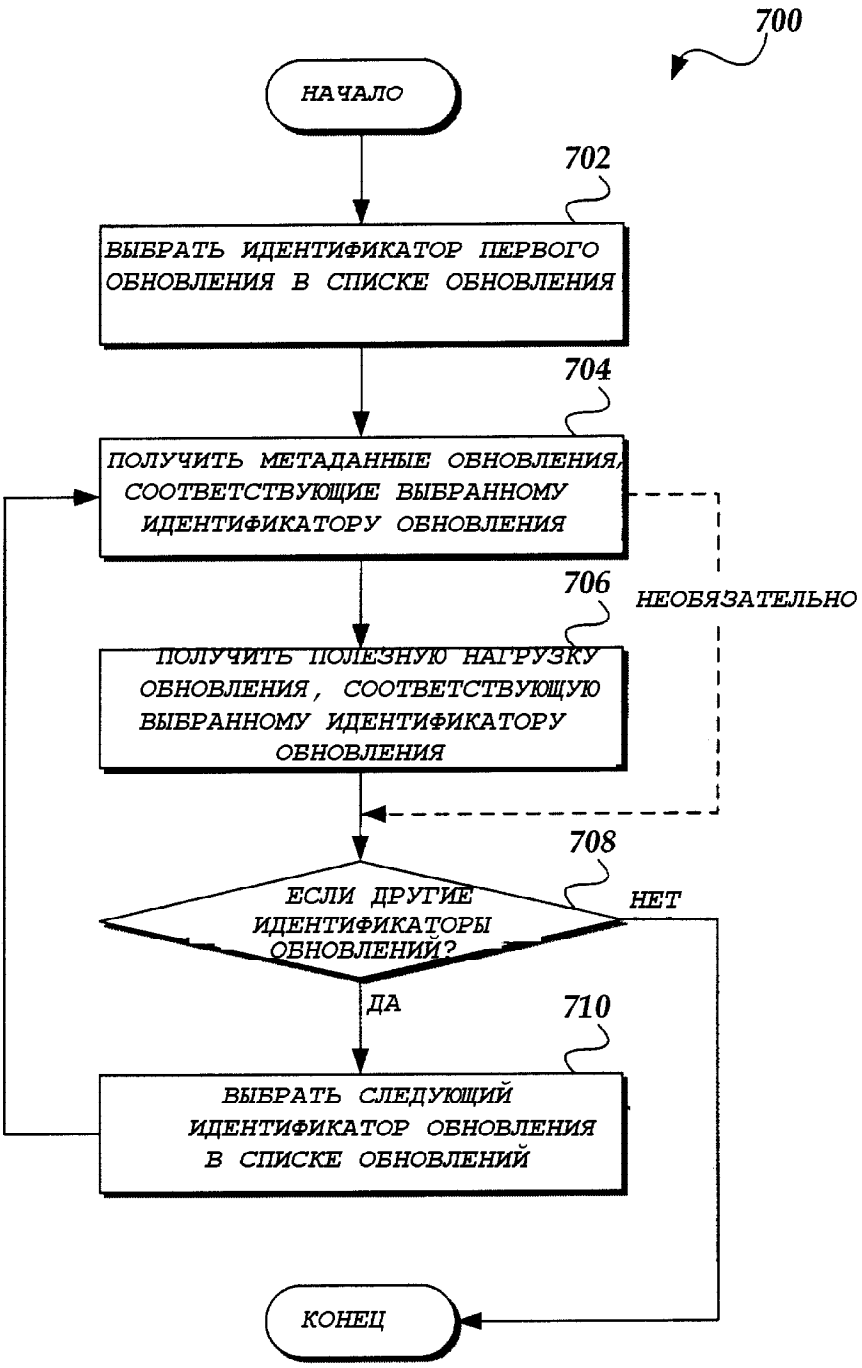
ФИГ. 4B


500
НАЧАЛО
502
ПОЛУЧИТЬ СИНХРОНИЗИРОВАННЫЙ СПИСОК ОБНОВЛЕНИЙ ОТ ПРЕДКА (ФИГ. 6)
504
ДОСТУПНЫ ЛИ КАКИЕ-ЛИБО ОБНОВЛЕНИЯ?
НЕТ
ДА
506
ПОЛУЧИТЬ ДОСТУПНОЕ(ЫЕ) ОБНОВЛЕНИЕ(Я) ОТ ПРЕДКА (ФИГ. 7)
508
СООБЩИТЬ ПРЕДКУ О ДЕЙСТВИЯХ ПО ОБНОВЛЕНИЮ
НЕОБЯЗАТЕЛЬНО
510
ЗАДЕРЖКА

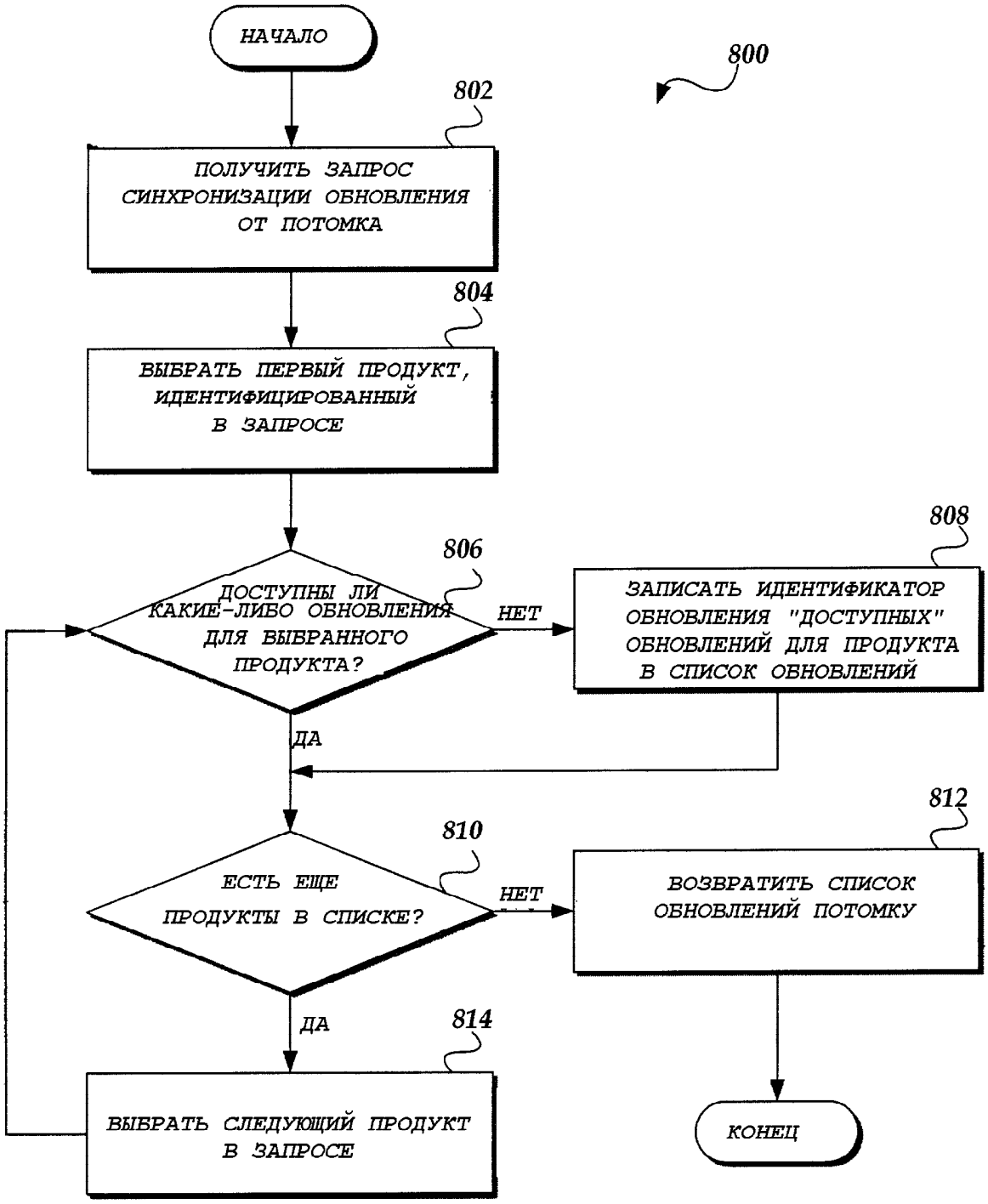
ФИГ. 5


600
НАЧАЛО
602
АУТЕНТИФИЦИРОВАТЬ И АВТОРИЗОВАТЬ С ПОМОЩЬЮ ПРЕДКА
604
УСТАНОВИТЬ ПЕРЕДАЧУ ПАРАМЕТРОВ С ПРЕДКОМ
606
ПОЛУЧИТЬ КАТАЛОГ ОБНОВЛЕНИЙ ПРОДУКТА ОТ ПРЕДКА
608
ВЫБРАТЬ ПРОГРАММНЫЕ ПРОДУКТЫ
610
ПОДАТЬ ЗАПРОС СИНХРОНИЗАЦИИ
612
ПОЛУЧИТЬ СПИСОК ОБНОВЛЕНИЙ, ИДЕНТИФИЦИРУЮЩИЙ ДОСТУПНЫЕ ОБНОВЛЕНИЯ, ОТ ПРЕДКА
КОНЕЦ

ФИГ. 6


700
НАЧАЛО
702
ВЫБРАТЬ ИДЕНТИФИКАТОР ПЕРВОГО ОБНОВЛЕНИЯ В СПИСКЕ ОБНОВЛЕНИЯ
704
ПОЛУЧИТЬ МЕТАДАННЫЕ ОБНОВЛЕНИЯ, СООТВЕТСТВУЮЩИЕ ВЫБРАННОМУ ИДЕНТИФИКАТОРУ ОБНОВЛЕНИЯ
706
НЕОБЯЗАТЕЛЬНО
ПОЛУЧИТЬ ПОЛЕЗНУЮ НАГРУЗКУ ОБНОВЛЕНИЯ, СООТВЕТСТВУЮЩУЮ ВЫБРАННОМУ ИДЕНТИФИКАТОРУ ОБНОВЛЕНИЯ
708
ЕСЛИ ДРУГИЕ ИДЕНТИФИКАТОРЫ ОБНОВЛЕНИЙ?
НЕТ
ДА
710
ВЫБРАТЬ СЛЕДУЮЩИЙ ИДЕНТИФИКАТОР ОБНОВЛЕНИЯ В СПИСКЕ ОБНОВЛЕНИЙ
КОНЕЦ

ФИГ. 7


НАЧАЛО
800
802
ПОЛУЧИТЬ ЗАПРОС СИНХРОНИЗАЦИИ ОБНОВЛЕНИЯ ОТ ПОТОМКА
804
ВЫБРАТЬ ПЕРВЫЙ ПРОДУКТ, ИДЕНТИФИЦИРОВАННЫЙ В ЗАПРОСЕ
806
ДОСТУПНЫ ЛИ КАКИЕ-ЛИБО ОБНОВЛЕНИЯ ДЛЯ ВЫБРАННОГО ПРОДУКТА?
НЕТ
808
ЗАПИСАТЬ ИДЕНТИФИКАТОР ОБНОВЛЕНИЯ "ДОСТУПНЫХ" ОБНОВЛЕНИЙ ДЛЯ ПРОДУКТА В СПИСОК ОБНОВЛЕНИЙ
ДА
810
ЕСТЬ ЕЩЕ ПРОДУКТЫ В СПИСКЕ?
НЕТ
812
ВОЗВРАТИТЬ СПИСОК ОБНОВЛЕНИЙ ПОТОМКУ
ДА
814
ВЫБРАТЬ СЛЕДУЮЩИЙ ПРОДУКТ В ЗАПРОСЕ
КОНЕЦ

ФИГ. 8

```
<?xml version="1.0" encoding="UTF-8" ?>

<schema targetNamespace="http://schemas.microsoft.com/msus/2002/12/Update"                    900
     xmlns="http://www.w3.org/2001/XMLSchema"
     xmlns:upd="http://schemas.microsoft.com/msus/2002/12/Update"
     xmlns:bt="http://schemas.microsoft.com/msus/2002/12/BaseTypes"
     elementFormDefault="qualified" attributeFormDefault="unqualified">

  <element name="Update" type="upd:Update" />

  <complexType name="Update">          902                                          918

     <sequence>

 904          <element name="UpdateIdentity" type="upd:UpdateIdentity" />
 906          <element name="Properties" type="upd:Properties" minOccurs="0" />
 908          <element name="LocalizedPropertiesCollection"
                       type="upd:LocalizedPropertiesCollection" minOccurs="0" />
 910          <element name="Relationships" type="upd:Relationships" minOccurs="0" />
 912          <element name="ApplicabilityRules" type="upd:ApplicabilityRules"
                       minOccurs="0" />
 914          <element name="Files" type="upd:Files" minOccurs="0" />
 916          <element name="HandlerSpecificData" type="upd:HandlerSpecificData"
                       minOccurs="0" />

     </sequence>

  </complexType>
     •
     •
     •
</schema>
```

ФИГ. 9