РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 419 986** (13) **C2**

(51) МПК
H04H 60/29 (2008.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2008143372/08, 03.04.2007**

(24) Дата начала отсчета срока действия патента:
03.04.2007

Приоритет(ы):
(30) Конвенционный приоритет:
05.04.2006 US 11/398,863

(43) Дата публикации заявки: **10.05.2010** Бюл. № 13

(45) Опубликовано: **27.05.2011** Бюл. № **15**

(56) Список документов, цитированных в отчете о поиске: **RU 2257014 C2, 20.07.2005. RU 2216036 C2, 10.11.2003. US 2005/0235318 A1, 20.10.2005. US 2005/0243366 A1, 03.11.2005.**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **05.11.2008**

(86) Заявка PCT:
US 2007/065886 (03.04.2007)

(87) Публикация заявки PCT:
WO 2007/118096 (18.10.2007)

Адрес для переписки:
129090, Москва, ул.Б.Спасская, 25, стр.3, ООО "Юридическая фирма Городисский и Партнеры", пат.пов. А.В.Мищу, рег.№ 364

(72) Автор(ы):
АГИЛАР-МАСИАС Эктор (US),
МАНТРИ Гириш (US)

(73) Патентообладатель(и):
АРКСАЙТ, ИНК. (US)

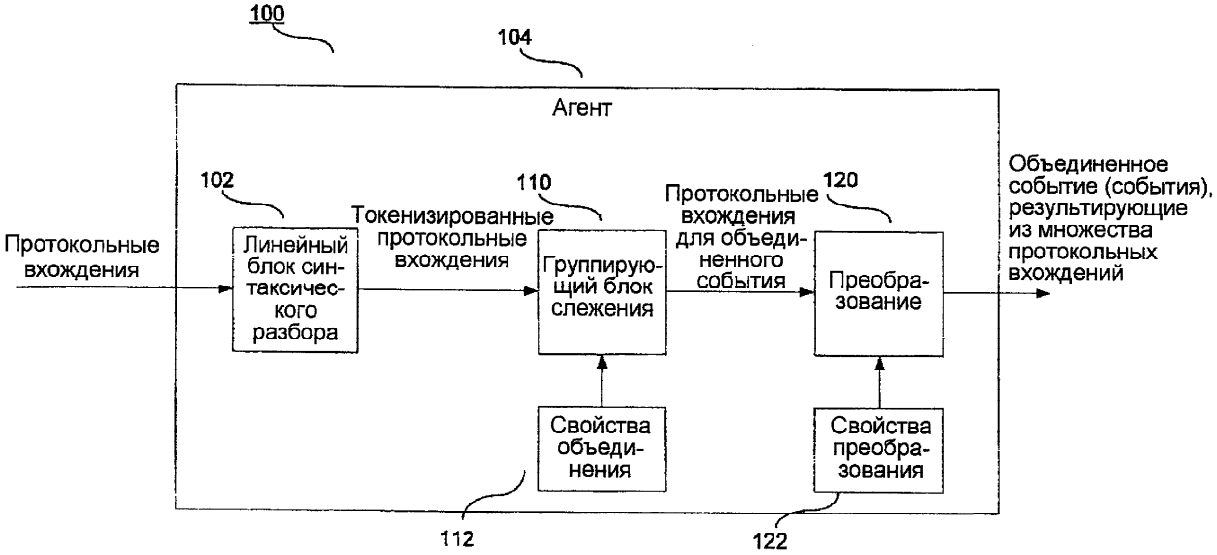
RU 2419986 C2

(54) ОБЪЕДИНЕНИЕ МНОГОСТРОЧНЫХ ПРОТОКОЛЬНЫХ ВХОЖДЕНИЙ

(57) Реферат:

Изобретение относится к области мониторинга сетевой активности. Техническим результатом является повышение эффективности обработки протокольных вхождений, принятых от множественных устройств. Способ построения объединенных событий из протокольных вхождений, принятых системой обработки данных, содержит этапы, на которых осуществляют прием множества протокольных вхождений; для каждого принятого протокольного вхождения определяют, что протокольное вхождение содержит ID (идентификатор) общий для объединенного события, в соответствии со свойствами объединения; если протокольное вхождение представляет собой начальное протокольное вхождение потенциального объединенного события в соответствии со свойствами объединения, начинают новое объединенное событие и преобразуют протокольное вхождение в новое объединенное событие в соответствии со свойствами преобразования для объединенного события; и если протокольное вхождение является оканчивающим протокольным

вхождением существующего объединенного события в соответствии со свойствами объединения, преобразуют протокольное вхождение в существующее объединенное событие в соответствии со свойствами преобразования для существующего объединенного события и осуществляют окончание существующего объединенного события. 3 н. и 16 з.п. ф-лы, 6 ил.



Фиг.1

RU 2419986 C2

RU 2419986 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(19) **RU** (11) **2 419 986** (13) **C2**

(51) Int. Cl.
H04H 60/29 (2008.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2008143372/08, 03.04.2007**

(24) Effective date for property rights:
03.04.2007

Priority:
(30) Priority:
05.04.2006 US 11/398,863

(43) Application published: **10.05.2010** Bull. 13

(45) Date of publication: **27.05.2011 Bull. 15**

(85) Commencement of national phase: **05.11.2008**

(86) PCT application:
US 2007/065886 (03.04.2007)

(87) PCT publication:
WO 2007/118096 (18.10.2007)

Mail address:
129090, Moskva, ul.B.Spasskaja, 25, str.3, OOO "Juridicheskaja firma Gorodisskij i Partnery", pat.pov. A.V.Mitsu, reg.№ 364

(72) Inventor(s):
AGILAR-MASIAS Ehktor (US),
MANTRI Girish (US)

(73) Proprietor(s):
ARKSAJT, INK. (US)

RU 2419986 C2

(54) COMBINING MULTILINE PROTOCOL ACCESSES

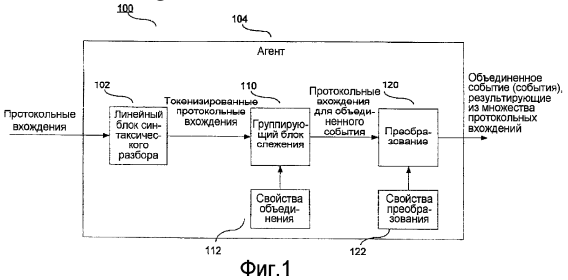
(57) Abstract:

FIELD: information technologies.

SUBSTANCE: method to build combined events from protocol accesses, received by a system of data processing, comprises stages, at which multiple protocol accesses are received; for each received protocol access: it is identified that the protocol access includes ID (an identifier), common for a combined event, in compliance with the combination properties; if a protocol access is an initial protocol access of a potential combined event in compliance with the combination properties: a new combined event is started; and the protocol access is transformed into a new combined event in compliance with the transformation properties for a combined event; and if a protocol access is an ending protocol access of the existing combined event in compliance with the combination properties: the protocol access is transformed into the existing combined event in compliance with the transformation properties for the existing combined event, and the existing combined event is terminated.

EFFECT: improved efficiency to process protocol accesses received from multiple devices.

19 cl, 6 dwg



Фиг.1

RU 2419986 C2

Область техники

Раскрытые варианты осуществления относятся, в общем, к мониторингу сетевой активности. Более конкретно, раскрытые варианты осуществления относятся к системе и способу для объединения множества вхождений, представляющих относящуюся активность сети.

Уровень техники

Является желательным мониторировать протокольные вхождения, принятые от различных устройств и частей программного обеспечения в сети. Часто эти другие устройства или части программного обеспечения могут создавать несколько сообщений протоколирования по причинам удобства, скорости или надежности. Это делается, например, чтобы некоторая информация достигла центрального пункта для события, даже если не вся информация достигает. Например, может быть желательно посылать протокольное сообщение до того, как работа выполнена, чтобы быть уверенным, что что-то записано, даже если система позже терпит аварию, до полного окончания данной работы.

В дополнение, некоторые типы протокольных событий происходят в устройстве с течением времени. Считается желательным посылать протоколируемые события по мере того, как они происходят, вместо того, чтобы ждать пока все протоколируемые случаи произойдут для события в устройстве.

Если многочисленные устройства посылают протокольные вхождения в один или несколько центральных пунктов сбора в сети, протокольные вхождения для различных событий от различных устройств будут, наиболее вероятно, прибывать разбросанными между собой. Различные протокольные вхождения могут быть не соседними в протоколе. Они могут чередоваться с очень похожими событиями. Они могут быть раскиданы по нескольким протокольным файлам. Последовательность вхождений может не быть полной (возможно, что датчик сломался до того, как операция была закончена).

Что необходимо - это способ для автоматического сбора информации о событиях высокого уровня из протокольных вхождений, которые были сгенерированы при проблемных условиях, описанных выше.

Сущность изобретения

Предпочтительные варианты осуществления настоящего изобретения определяют агент, содержащий блок синтаксического разбора, модуль группирующего блока слежения и модуль преобразования. Блок синтаксического разбора разделяет поступающие протокольные вхождения на токены. Группирующий блок слежения анализирует эти токены, чтобы определить, к каким объединенным событиям токены принадлежат (если есть какие-либо). В описанном варианте осуществления группирующий блок слежения работает в соответствии с конфигурируемыми свойствами объединения, хотя другие варианты осуществления могут иметь жестко закодированные свойства. Свойства объединения делают возможной конфигурацию различных свойств, ассоциированных с действием группирования протокольных вхождений в объединенные события высокого уровня. В описанном варианте осуществления эти свойства включают в себя некоторое или все из: какие типы протокольных вхождений будут рассматриваться для каждого объединенного события, какие ID используются для идентификации каждого объединенного события, какие вхождения начинают и оканчивают объединенное событие, значение таймаута, которое автоматически оканчивает сбор вхождений для существующего объединенного события, даже если никакое вхождение окончания не найдено.

В описанном варианте осуществления, модуль преобразования принимает протокольные вхождения, ассоциированные с конкретными объединенными событиями, и преобразует их в поля в структуре данных объединенного события в соответствии со свойствами преобразования (хотя эти свойства преобразования также могут быть жестко закодированы).

Описанные варианты осуществления изобретения используют регулярные выражения в свойствах объединения для описания значений, которые ищутся в принятых протокольных вхождениях. Например, регулярное выражение может определять, какие протокольные вхождения являются частью события со многими вхождениями, может определять первое вхождение в событии со многими вхождениями, может определять последнее вхождение в событии со многими вхождениями. Свойства объединения также определяют, какое поле во вхождениях должно содержать определенное значение в порядке объединения (например, запись может иметь какой-либо численный ID или ссылаться на какой-либо IP адрес). Описанный вариант осуществления настоящего изобретения может обрабатывать протокольные вхождения для событий, которые перемешены между собой.

Краткое описание чертежей

На ФИГ.1 показана блок-схема системы в соответствии с вариантом осуществления настоящего изобретения.

На ФИГ.2 показан алгоритм варианта осуществления способа выполнения обработки протокольных вхождений в соответствии со свойствами объединения.

На ФИГ.3 изображена блок-схема варианта осуществления способа выполнения добавления протокольного вхождения в объединенное событие в соответствии со свойствами объединения.

На ФИГ.4 изображена блок-схема, показывающая функцию one_of, используемую в свойствах преобразования в варианте осуществления настоящего изобретения.

На ФИГ.5 показан пример, на котором множественные объединенные события строятся по мере того, как принимаются разбросанные вхождения для различных объединенных событий.

На ФИГ. 6 показан пример формата объединенного события.

Варианты осуществления

Варианты осуществления настоящего изобретения здесь описаны со ссылками на чертежи, где схожие номера ссылок показывают одинаковые или функционально схожие элементы.

На ФИГ.1 показана блок-схема системы 100, соответствующей варианту осуществления настоящего изобретения. Система 100 предпочтительно содержит агент 104 в одном или более центральных точках сети. Агент 104 принимает протокольные вхождения от разнообразных устройств и частей программного обеспечения сети, такой как, например, интернет, локальная сеть, WAN, беспроводная сеть, мобильная сеть или любой другой подходящий механизм, который позволяет удаленным устройствам посылать протокольные вхождения для агента 104.

Протокольные вхождения принимаются блоком 102 синтаксического разбора и разделяются посредством синтаксического разбора на токены способом, известным специалистам в данной области техники. В другом варианте осуществления синтаксический разбор выполняется, как описано в заявке США No. 11/070024 от Hector Aguilar-Macias et al, под названием "Message Parsing In A Network Security System," поданной 1 марта 2005 года, которая приведена в настоящем описании посредством ссылки.

Принятое протокольное вхождение может иметь любой подходящий формат, для которого блок 102 синтаксического разбора может осуществлять синтаксический разбор. Блок 102 синтаксического разбора выдает токены на основании принятых записей системного журнала. Эти токены принимаются модулем 110 группирующего блока слежения.

Модуль 110 группирующего блока слежения соединен, чтобы принимать свойства объединения из памяти или другого модуля хранения или устройства 112. Свойства объединения определяют, как должны интерпретироваться принятые протокольные вхождения, когда они используются для построения объединенных событий. Модуль группирующего блока слежения выдает протокольные вхождения, которые ассоциированы с конкретными объединенными событиями в модуль преобразования, где протокольные вхождения преобразуются в объединенные события, которые строятся из принятых протокольных вхождений. Это преобразование происходит в соответствии со свойствами 122 преобразования. Выход модуля 120 преобразования это одно или более объединенных событий, являющихся результатом множества протокольных вхождений. Процесс, в общем виде описанный на ФИГ.1, будет описан более детально ниже в связи с примером.

Пример

Здесь пример того, как объединение событий работает в варианте осуществления данного изобретения:

Предположим, что следующие строки протокольных вхождений (они также иногда называются "сообщения"):

[18/Jul/ 2005:12:30:20 -0400] conn=8 op=0 msgId=82 - BIND uid=admin
[18/Jul/ 2005:12:30:25 -0400] conn=7 op=-1 msgId=-1 - LDAP
connection from 10.0.20.122 to 10.0.20.122
[18/Jul/ 2005:12:30:30 -0400] conn=8 op=0 msgId=82 - RESULT err=0

Блок 102 синтаксического разбора осуществляет синтаксический разбор этих принятых протокольных вхождений в пары ключ-значение. Для каждого протокольного вхождения выдается набор токенов. Например, протокольное вхождение:

[18/JuI/2005:12:30:20 -0400]] conn=8 op=0 msgId=82 - BIND uid=admin

Результирующие токены имеют следующие пары ключ/значение:

Date=18/Jul/2005 12:30:20
Connection=8
Operation=0
MessageId=82
OperationName=BIND
UserId=admin

Аналогично, другие два протокольных вхождения выдают их собственные пары ключ/значение:

[18/JuI/2005:12:30:25 -0400]] conn=7 op=-1 msgId=-1 - LDAP
connection from 10.0.20.122 to 10.0.20.12
Date=18/Jul/2005 12:30:25
Connection=7
Operations
MessageId=-1
OperationName=LDAP
Source=10.0.20.122

Destination=10.0.20.12
[18/[uI/2005:12:30:30 -0400]] conn=8 op=0 msgId=82 - RESULT err=0
Date=18/Jul/2005 12:30:30
Connection=8
Operation=0
MessageId=82
OperationName=RESULT
ResultCode=0

На ФИГ.2 изображен алгоритм 200 варианта осуществления способа выполнения процесса приема протокольных вхождений в соответствии со свойствами объединения 112. В предпочтительном варианте осуществления способ выполняется модулем группирования/блоком слежения 110. Если таймаут 202 достигнут для объединенного события, формирующегося в настоящий момент, объединенное событие оканчивается 204 и управление возвращается элементу 202. Таким образом, даже если никакого явного оканчивающего протокольного вхождения не найдено, объединенное событие будет закрыто, когда наступает его таймаут. Значение таймаута может быть различным для различных типов устройств протоколирования и для различных объединенных событий от одного устройства. Как описано ниже, значение таймаута содержится в свойствах объединения.

Элемент 206 принимает следующее протокольное вхождение для обработки. Если протокольное вхождение рассматривается для объединения 208 (как определено в свойствах объединения 112), обработка продолжается, в противном случае посылается одиночное событие 209 и обработка возвращается к элементу 202.

Если протокольное вхождение является начальным протокольным вхождением для нового объединенного события 210 (как определено в свойствах объединения 112), открывается новое объединенное событие 212 (смотрите ФИГ.5 как пример множественных объединенных событий в процессе построения). В некоторых вариантах осуществления запускаются часы таймаута для объединенного события 212.

Если протокольное вхождение не является начальным вхождением, но оно содержит ID существующего объединенного события, строящегося в текущий момент 214, тогда ошибка протоколируется и посылается одиночное событие 215. В противном случае обработка продолжается, и токены и протокольное вхождение переходят 220 в модуль преобразования таким образом, что его информация может быть добавлена в объединенное событие. В различных вариантах осуществления ID может иметь одно поле в протокольном вхождении или иметь много полей в протокольном вхождении, которые содержат общие значения для всех записей системного журнала объединенного события.

Если протокольное вхождение является протокольным вхождением окончания для нового объединенного события 216 (как определено в свойствах объединения 112), существующее объединенное событие оканчивается и перемещается 218 из модуля группирования/блока слежения (смотрите ФИГ.5 как пример множественных объединенных событий в процессе построения). Если протокольное вхождение показывает конец события, соответствующее объединенное событие будет закончено и перемещено из системы на ФИГ.5.

Для продолжения примера, свойства объединения 112 в этом примере определены так:

merge.count=1
merge[0].pattern.count=1

merge[0].pattern [0].token=OperationName
merge[0].pattern[0].regex=(BIND RESULT)
merge[0].starts.count=1
merge[0].starts[0].token=OperationName
merge[0].starts[0].regex=BIND
merge[0].ends.count=1
merge[0].ends[0].token=OperationName
merge [0].ends[0].regex=RESULT
merge[0].id.tokens=Connection,Operation,MessageId
merge[0].timeout=60000

В первую очередь мы покажем случай, когда мы имеем только одну операцию объединения:

merge.count=1

Потом мы определим, что мы хотим все сообщения с OperationName, установленным как BIND или RESULT, использовать для объединения:

merge[0].pattern.count=1
merge [0]. pattern [0].token=OperationName
merge[0].pattern[0].regex=(BIND RESULT)

Теперь мы установим, какие сообщения имеют OperationName установленными как BIND и начнем операцию объединения:

merge[0].starts.count=1
merge[0].starts[0].token=OperationName
merge[0].starts[0].regex=BIND

И что операция объединения закончится, когда мы найдем сообщение с OperationName установленным как RESULT:

merge[0].ends.count=1
merge[0].ends[0].token=OperationName
merge[0].ends[0].regex=RESULT

Мы также должны определить, как идентифицировать, что события принадлежат к одной группе, мы сделаем это посредством специфицирования, что значения Connection, Operation и MessageId должны быть идентичны (формируя ID для объединенного события):

merge[0].id.tokens=Connection,Operation,MessageId

Наконец, мы определим таймаут так, что если мы не получим сообщение с OperationName, установленным как RESULT после 60 секунд, тогда мы посылаем сообщение:

merge[0].timeout=60000

На ФИГ.3 показана блок-схема варианта осуществления метода выполнения добавления протокольного вхождения в объединенное событие в соответствии со свойствами преобразования. Принятые протокольные вхождения и их токены уже идентифицированы по отношению к не менее чем одному объединенному событию, которое строится. Модуль 120 преобразования преобразует информацию в протокольных вхождениях к одному или более объединенных событий, которые строятся (смотрите ФИГ.5, как пример построения объединенных событий, смотрите ФИГ.6 для примера формата объединенного события).

В этом примере свойства 122 преобразования определены так:

event.deviceReceiptTime=Date
event.name= oneOf(mergedevent.name,OperationName)

event.deviceAction=ResultCode
event.destinationUserId=UserId

Эти свойства показывают, как мы будем использовать Date, как отметку времени для события, ResultCode, как действие устройства и UserId, как id пользователя назначения. Имя определено как:

event.name=oneOf(mergedevent.name,OperationName)

Поскольку эта схема также разрешает вам обращаться к событию «отслеживания», оно используется для хранения конечных данных. В данном случае в работе мы будем использовать или OperationName, или имя события «отслеживания» (если какое-либо). Например, первое событие будет содержать следующие ключ-значения:

[18/[uI/2005:12:30:20 -0400]] conn=8 op=0 msgId=82 - BIND uid=admin
Date=18/Jul/2005 12:30:20
Connection=8
Operation=0
MessageId=82
OperationName=BIND
UserId=admin

И новое событие «отслеживания» будет создано, что закончит со следующими преобразованиями:

mergedevent.name=BIND
mergedevent.deviceReceiptTime=18/Jul/2005 12:30:20
mergedevent.destinationUserId=admin

Название объединенного события будет BIND, поскольку это новое объединенное событие, таким образом, имя объединенного события не существует и используется значение OperationName (BIND). Теперь, когда обрабатывается второе события для объединенной группы:

[18/[uI/2005:12:30:30 -0400]] conn=8 op=0 msgId=82 - RESULT err=0
Date=18/Jul/2005 12:30:30
Connection=8
Operations
MessageId=82
OperationName=RESULT
ResultCode=0

Объединенное событие будет преобразываться, как показано ниже:

mergedevent.name=BIND
mergedevent.deviceReceiptTime=18/Jul/2005 12:30:30
mergedevent.destinationUserId=admin
mergedeve[pi]t.deviceAction=0

Заметим, что mergedevent.name будет установлено BIND, поскольку, когда это событие обрабатывается, оно уже «отслеженное» событие (объединенное событие) с именем, установленным как BIND, таким образом, в этом случае OperationName НЕ БУДЕТ использоваться, и объединенное событие сохранит имя BIND. Заметим, что как mergedevent.deviceReceiptTime теперь установленно 18/Jul/2005 12:30:30, это потому, что определение значения mergedevent было возвращено, таким образом deviceReceiptTime приобретет новое значение.

На ФИГ.4 изображен алгоритм 400, показывающий функцию oneOf 402, использующуюся в свойствах преобразования в варианте осуществления данного изобретения. Для выполнения функции oneOf для, например, имени события, если имя

события в настоящее время пустое 404, используется текущее имя токена 406. Если имя не пустое, не пустое имя сохраняется 408.

Следует понимать, что _oneOf только пример операций, которые могут использоваться в компоненте преобразований. Компонент преобразования может содержать другие «операции», которые могут делать ссылки к полям объединенного события. _oneOf это только пример, в существующем способе преобразования. Другие примеры операций включают в себя конкатенирование, операции преобразования типа и другие.

На ФИГ.5 показан пример 500, в котором множественные объединенные события строятся по мере приема разбросанных протокольных вхождений для различных объединенных событий.

На ФИГ.6 показан пример формата 550 одного объединенного события. Например, одно из различных объединенных событий на ФИГ.5 будет иметь этот формат, хотя не все значения могут быть найдены для этого объединенного события. Различные варианты осуществления настоящего изобретения будут содержать другие примеры операций объединения, включая сюда конкатенирование, преобразование типа, подсчет и другие. Другие варианты осуществления включают в себя сбор объединенного события для того, чтобы могла вестись статистика для номеров различных типов объединенных событий. Эти собранные данные могут быть посланы регистратору по отдельности или как часть в сочетании с другими посылаемыми данными.

Следующие параграфы предоставляют короткое описание примеров свойств 112 объединения, включенных в один вариант осуществления изобретения:

merge.count

Определяет номер операций объединения, которые будут определяться.

merge[{mergeindex}].[gamma]pattern.count

Определяет, как много образцов будет определено. Операции объединения требуют образцы для определения того, какие события будут учитываться в операции объединения, если не дано образцов, тогда учитываются все события.

merge[{mergeindex}].pattern[{patternindex}].token

Определяет метку, которая будет использоваться для этого образца.

merge[{mergeindex}].pattern[{patternindex}].regex

Определяет регулярное выражение, использующееся для этого образца.

merge[{mergeindex}].starts.count

Определяет, сколько образцов начала определено. Операции объединения требуют образцы начала для определения того, какие события начнут операцию объединения, если не дано образцов, тогда все события начинают операцию объединения. Если операция единожды началась, она может закончиться только через таймаут или совпадение с образцом конца.

merge[{mergeindex}].starts[{patternindex}].token

Определяет метку, которая будет использоваться для этого образца начала.

merge[{mergeindex}].starts[{patternindex}].regex

Определяет регулярное выражение, использующееся для этого образца начала.

merge[{mergeindex}].ends.count

Определяет, как много образцов конца будет определено. Операции объединения требуют образцы конца для определения того, какие события закончат операцию объединения, если не дано образцов, тогда не одно событие не закончит операцию объединения, операция закончится только через таймаут.

merge[{mergeindex}].ends[{patternindex}].token

Определяет токен, который будет использоваться для этого образца конца.

merge[{mergeindex}].ends[{patternindex}].regex

Определяет регулярное выражение для использования для этого образца конца.

merge[{mergeindex}].timeout

Определяет таймаут в миллисекундах для операции объединения. Если таймаут достигнут, тогда операция объединения оканчивается и события посылаются. Понятно, что эти события будут посланы через различные каналы связи, таким образом, порядок событий не гарантируется.

merge[{mergeindex}].id.tokens

Определяет список токенов, которые будут использоваться для группировки событий. Это свойство обязательно.

merge[{mergeindex}].id.delimiter

Определяет факультативный разделитель для использования для списка выше, если он не определен, тогда разделителем является запятая (,).

merge[{mergeindex}].sendpartialevents

Это свойство является факультативным и по умолчанию установлено как ложно. По существу, она определяет, должно ли каждое событие в объединенном событии быть послано индивидуально, хотя оно объединено с другими событиями.

merge[{mergeindex}].capacity

Это свойство является факультативным и по умолчанию установлено в 1000. Операция объединения событий требует кэш событий, которая хранит результаты объединения. Этот параметр определяет насколько большим будет кэш, и если кэш переполняется, тогда события будут посылаться, как они есть, и ошибка будет запротоколирована.

Ссылка в описании на «один вариант осуществления» или к «вариант осуществления» означает, что конкретный признак, структура или характеристика, описанные в связи с вариантами осуществления, включены в, по крайней мере, один вариант осуществления изобретения. Применение фразы «в одном варианте осуществления» в различных местах спецификации не обязательно всегда означает один и тот же вариант осуществления.

Некоторые части написанного выше преподносятся в элементах алгоритмов и символьных представлениях операций над битами денных в памяти компьютера. Алгоритмические описания и представления - это способ, использующийся теми, кто является специалистами в области обработки данных, чтобы наиболее эффективно передать сущность своей работы другим сведущим в данной области. Представленный здесь алгоритм, в общем, сконструирован так, чтобы он имел самосогласованную последовательность этапов (инструкций), приводящих к желаемому результату. Этапы требуют физических операций с физическими величинами. Обычно, хотя не необходимо, эти величины имеют вид электрических, магнитных или оптических сигналов, которые можно сохранять, объединять, сравнивать и производить другие манипуляции. Порой удобно, в основном по причине простоты использования, рассматривать эти сигналы как биты, значения, элементы, символы, термы, числа и т.п. К тому же, порой удобно указывать на некоторые компоновки этапов, требующих физических манипуляций с физическими величинами, как части модулей или кодирующих устройств без потери общности.

Как можно иметь в виду, однако, что все из этих и аналогичных терминов должны быть ассоциированы с подходящими физическими величинами и являются просто

удобными обозначениями, применяемыми для этих величин. Если не указанно специально, как видно из последующего обсуждения, следует принять во внимание, что на всем протяжении описания, описания, использующие термины, такие как «обработка», или «расчет», или «вычисление», или «определение», или «отображение», или сходные, указывают на действие и процессы компьютерной системы или сходного электронного вычислительного устройства, которая манипулирует и преобразует данные, представленные в виде физических (электронных) величин в памятях компьютерной системы или регистрах или других устройствах хранения информации, передачи или устройств отображения.

Некоторые аспекты данного изобретения включают в себя этапы обработки и инструкции, описанные здесь в форме алгоритма. Следует заметить, что этапы обработки и инструкции данного изобретения могут быть реализованы в программном обеспечении, встроенном программном обеспечении или аппаратном обеспечении, и в случае реализации в программном обеспечении может быть загружена в место постоянного хранения и работать из различных платформ, используемых различными операционными системами.

Настоящее изобретение также относится к аппаратуре для выполнения операций здесь. Эта аппаратура может быть специально сконструирована для требуемых целей или она может содержать компьютер универсального назначения, выборочно задействованный или переконфигурированный компьютерной программой, записанной в памяти компьютера. Эта компьютерная программа может быть записана на читаемом компьютером носителе, как, хотя это и не полный список, любой тип дисков, включая сюда флоппи диски, оптические диски, компакт диски, магнитооптические диски, постоянные запоминающие устройства (ROM), системы памяти с произвольным доступом (RAM), перепрограммируемые микросхемы памяти (EPROM, EEPROM), магнитные или оптические карты, интегральные схемы специализированного назначения (ASICs), или любые типы устройств, подходящих для хранения электронных инструкций и связанные с системной шиной компьютера. Кроме того, компьютеры, описанные в спецификации, могут включать в себя один процессор или иметь архитектуру, использующую несколько процессоров для увеличения возможностей вычисления.

Алгоритмы и изображения, представленные здесь, не обязательно относятся к какому-либо конкретному компьютеру или другой аппаратуре. Различные системы универсального назначения также могут быть использованы с программами, соответствующими идее, представленной здесь, или может быть удобно сконструировать более специализированную аппаратуру для выполнения требуемых этапов способа. Требуемая структура для многообразия этих систем будет показана в описании ниже. Дополнительно, настоящее изобретение не описано со ссылкой на какой-либо конкретный язык программирования. Следует понимать, что многообразие языков программирования может использоваться для выполнения предмета данного изобретения, как описано здесь, и все описанное ниже по отношению к конкретным языкам представляет возможности и наилучший вариант данного изобретения.

Пока изобретение было подробно показано и описано со ссылкой на предпочтительный вариант осуществления и некоторыми альтернативными вариантами осуществления, понятно, что лицо, осведомленное в данной области, может внести различные изменения в форме и деталях без отклонения от истинного смысла и области данного изобретения.

Наконец, необходимо заметить, что язык, используемый в описании, был принципиально выбран для удобства чтения и назначения инструкций и не может быть выбран для описания ограничений патентоспособности сущности предмета изобретения. Соответственно, опубликование данного изобретения предназначено показать изобретение, но не ограничивать область изобретения, которая установлена далее в пунктах формулы изобретения.

Формула изобретения

1. Способ построения объединенных событий из протокольных вхождений, принятых системой обработки данных, содержащий следующие шаги:

осуществляют прием множества протокольных вхождений;

для каждого принятого протокольного вхождения:

определяют, что протокольное вхождение содержит ID (идентификатор), общий для объединенного события, в соответствии со свойствами объединения;

если протокольное вхождение содержит ID (идентификатор), общий для объединенного события, и если протокольное вхождение представляет собой начальное протокольное вхождение потенциального объединенного события в соответствии со свойствами объединения:

начинают новое объединенное событие; и

преобразуют протокольное вхождение в новое объединенное событие в соответствии со свойствами преобразования для объединенного события; и

если протокольное вхождение содержит ID (идентификатор), общий для существующего объединенного события, и если протокольное вхождение является оканчивающим протокольным вхождением существующего объединенного события в соответствии со свойствами объединения:

преобразуют протокольное вхождение в существующее объединенное событие в соответствии со свойствами преобразования для существующего объединенного события, и

осуществляют окончание существующего объединенного события.

2. Способ по п.1, в котором дополнительно:

осуществляют окончание существующего объединенного события, если наступает таймаут, как определено в свойствах существующего объединения для существующего объединенного события.

3. Способ по п.1, в котором дополнительно определяют, указывает ли протокольное вхождение на начало объединенного события.

4. Способ по п.21, в котором дополнительно определяют, указывает ли протокольное вхождение на окончание объединенного события.

5. Способ по п.1, в котором дополнительно осуществляют идентификацию протокольного вхождения, которое имеет следствием ни начало, ни окончание объединенного события.

6. Способ по п.1, в котором дополнительно определяют возможность рассматривать объединенное событие как объединенное событие, существующее пока, когда осуществляется объединение в новый токен протокольного вхождения.

7. Способ по п.6, в котором используют возможность в операции преобразования.

8. Способ по п.1, в котором дополнительно определяют, будет ли каждое принятое протокольное вхождение приниматься во внимание для объединения в соответствии со свойствами объединения.

9. Способ по п.1, в котором для преобразования протокольного вхождения в

объединенное событие дополнительно определяют время объединенного события, упомянутое время является временем начального протокольного события для объединенного события.

10. Способ по п.1, в котором для преобразования протокольного вхождения в объединенное событие дополнительно определяют время объединенного события, упомянутое время является временем оканчивающего протокольного события для объединенного события.

11. Способ по п.1, в котором для преобразования протокольного вхождения в объединенное событие дополнительно осуществляют преобразование ID события в соответствии со свойствами преобразования.

12. Способ по п.1, в котором для преобразования протокольного вхождения в объединенное событие дополнительно осуществляют преобразование имени события в соответствии со свойствами преобразования.

13. Способ по п.1, в котором для преобразования протокольного вхождения в объединенное событие дополнительно осуществляют преобразование имени, выделенного посредством синтаксического разбора из протокольного вхождения, преобразование выполняют в соответствии с функцией oneOf в свойствах преобразования.

14. Способ по п.1, в котором для преобразования протокольного вхождения в объединенное событие дополнительно осуществляют преобразование действия устройства в соответствии со свойствами преобразования.

15. Способ по п.1, в котором принятые протокольные вхождения содержат протокольные вхождения, соответствующие более чем одному объединенному событию, перемешанные вместе.

16. Способ по п.1, в котором одно принятое протокольное вхождение используют для построения более чем одного объединенного события.

17. Способ по п.1, в котором ID содержит множество полей в протокольном вхождении, множество полей предназначено для идентификации протокольных вхождений, которые вносят вклад в объединенное событие.

18. Система построения объединенных событий из протокольных вхождений, принятых системой обработки данных, содержащая:

модуль для приема множества протокольных вхождений;

блок синтаксического разбора для осуществления синтаксического разбора протокольных вхождений на токены;

модуль группирования, который для каждого принятого протокольного вхождения, разобранного на токены:

определяет, что протокольное вхождение содержит ID (идентификатор), общий для объединенного события, в соответствии со свойствами объединения;

начинает новое объединенное событие, если протокольное вхождение является начальным протокольным вхождением потенциального объединенного события в соответствии со свойствами объединения; и

оканчивает существующее объединенное событие, если протокольное вхождение является оканчивающим протокольным вхождением существующего объединенного события в соответствии со свойствами объединения; и

модуль преобразования, который преобразует каждое протокольное вхождение, содержащее ID, общий для существующего объединенного события, в существующее объединенное событие в соответствии со свойствами преобразования для существующего объединенного события.

19. Машиночитаемый носитель данных с сохраненным на нем компьютерным программным продуктом, содержащим инструкции для побуждения компьютера осуществлять способ построения объединенных событий из протокольных вхождений, содержащий:

прием множества протокольных вхождений;

для каждого принятого протокольного вхождения:

определение того, что протокольное вхождение содержит ID (идентификатор), общий для объединенного события, в соответствии со свойствами объединения;

если протокольное вхождение содержит ID (идентификатор), общий для объединенного события, и если протокольное вхождение представляет собой начальное протокольное вхождение потенциального объединенного события в соответствии со свойствами объединения:

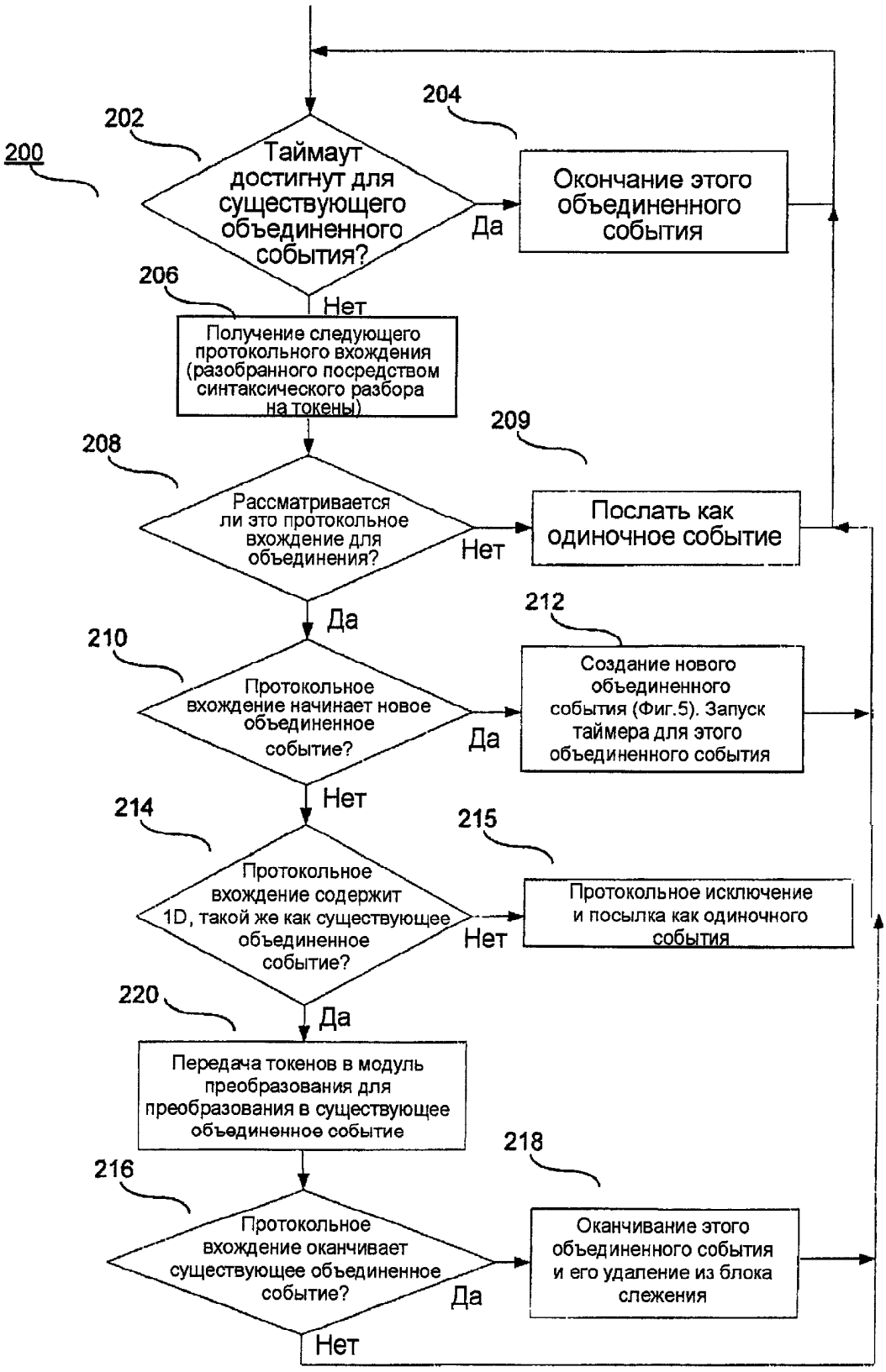
начинание нового объединенного события; и

преобразование протокольного вхождения в новое объединенное событие в соответствии со свойствами преобразования для объединенного события; и

если протокольное вхождение содержит ID (идентификатор), общий для существующего объединенного события, и если протокольное вхождение является оканчивающим протокольным вхождением существующего объединенного события в соответствии со свойствами объединения:

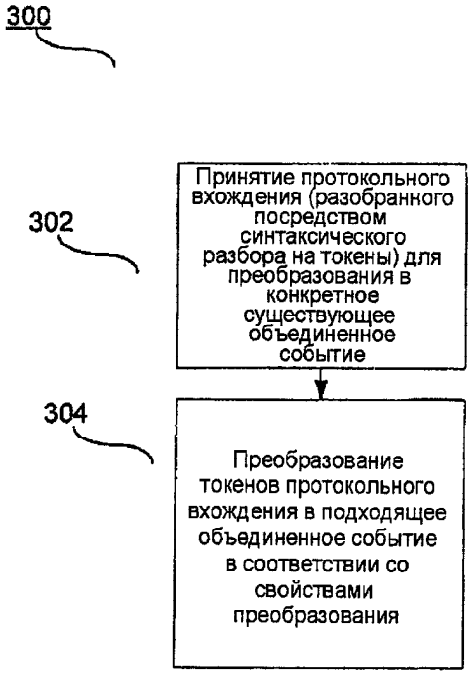
преобразование протокольного вхождения в существующее объединенное событие в соответствии со свойствами преобразования для существующего объединенного события, и

заканчивание существующего объединенного события.



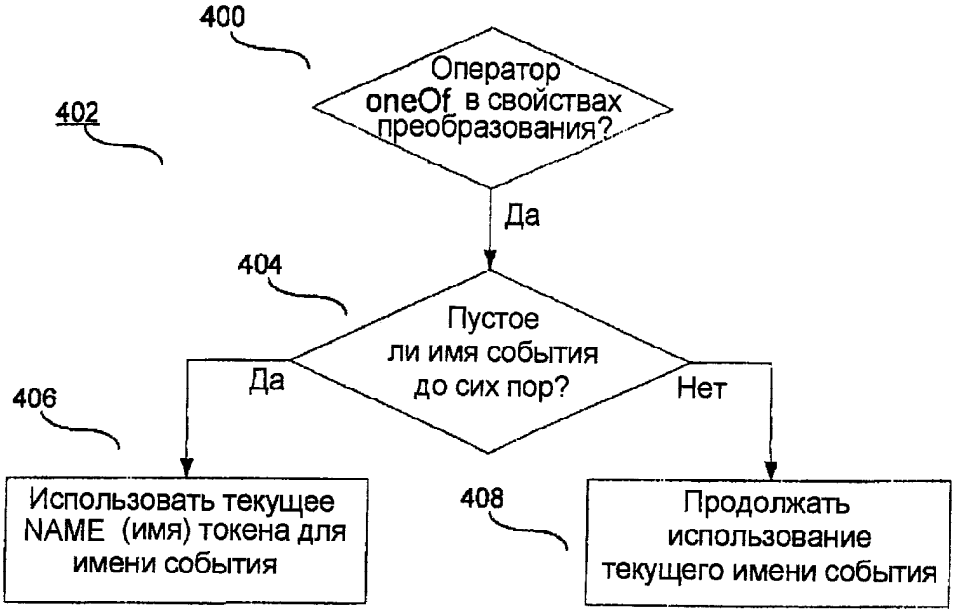
Обработка протокольных вхождений

Фиг.2



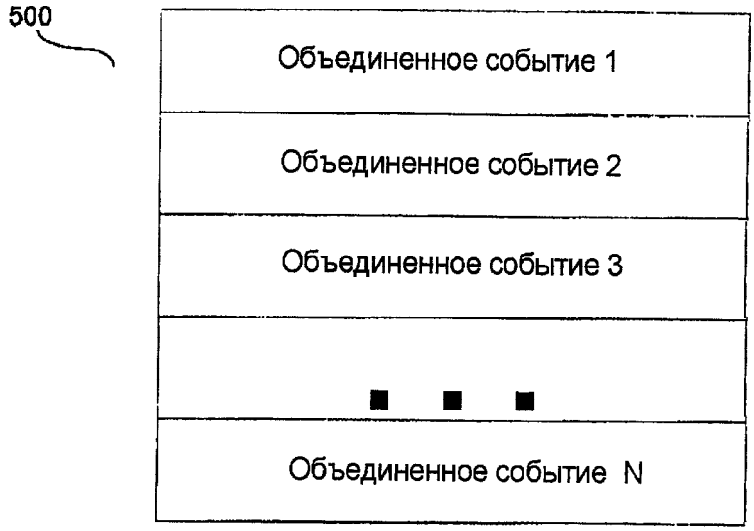
Преобразование протокольного вхождения в объединенное событие

Фиг.3



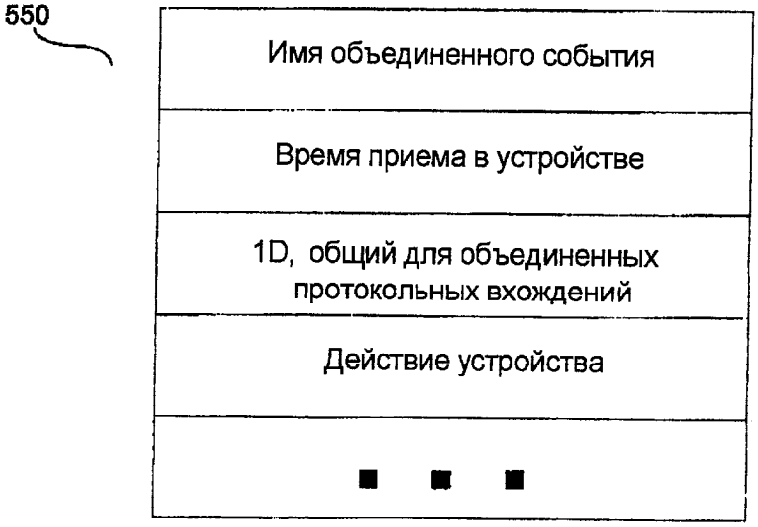
Функция oneOf

Фиг.4



Пример множественных объединенных событий в процессе построения

Фиг.5



Пример формата для одного объединенного события

Фиг.6