



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2007133808/08, 08.03.2006

(24) Дата начала отсчета срока действия патента:
08.03.2006

Приоритет(ы):

(30) Конвенционный приоритет:
08.03.2005 US 60/659,616
29.04.2005 US 11/118,544

(43) Дата публикации заявки: 20.03.2009 Бюл. № 8

(45) Опубликовано: 27.06.2011 Бюл. № 18

(56) Список документов, цитированных в отчете о
поиске: RU 2236751 C2, 20.09.2004. US
2002/0107684 A1, 08.08.2002. US 2004/0102956
A1, 27.05.2004. US 6,092,036 A, 18.07.2000.
Kevin Knight et al. «Building a large-scale
knowledge base for machine translation»,
опубликован 1994, найден в Интернет URL:
<http://www.aai.org/Papers/AAAI/1994/AAAI94-118.pdf>.(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 10.09.2007(86) Заявка РСТ:
US 2006/012901 (08.03.2006)(87) Публикация заявки РСТ:
WO 2006/132700 (14.12.2006)

Адрес для переписки:

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

(72) Автор(ы):

КОЛЬМАЙЕР Бернхард (US),
БРАУНЕЛЛ Лори А. (US),
ВУ Вей (US),
Е Шэнхуа (Эдвард) (US),
МАРТИ Джорди Мола (US),
НЕЛЬСОН Ян Андерс (US),
ЭЛЬ-ГАММАЛ Мохаммед (US),
БЕННЕТ Джули Д. (US)

(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)

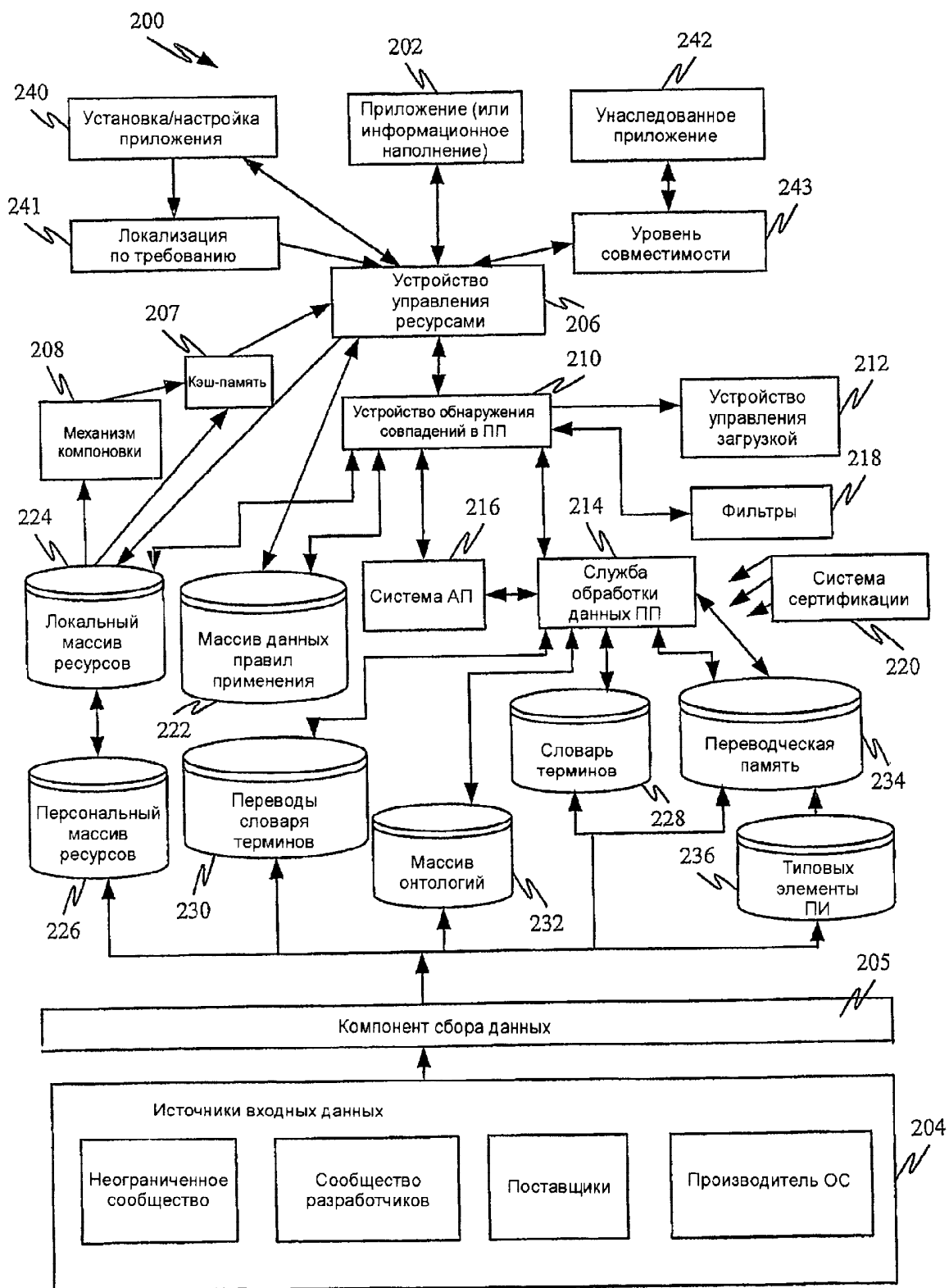
(54) СОЗДАНИЕ РЕСУРСА С ИСПОЛЬЗОВАНИЕМ ОНТОЛОГИЙ

(57) Реферат:

Изобретение относится к системам
локализации информационного наполнения
и/или программного обеспечения. Техническим
результатом является повышение
быстродействия за счет упрощениялокализации информационного наполнения
программных продуктов. Платформа
локализации включает: устройства управления
ресурсами, множество компонентов
локализации, массив онтологии, хранящий
информацию онтологии; и компонент

обнаружения совпадений в переводах, выполненный с возможностью приема запроса на локализацию и одного или более ресурсов от устройства управления ресурсами, при этом упомянутые один или более ресурсов содержат ресурсы от каждого из множества приложений, предназначенных для локализации, получения доступа к множеству компонентов локализации и массиву онтологии для

устранения неоднозначности данных, ассоциированных с конкретными ресурсами, предназначенными для локализации на основании запроса на локализацию, и выбора и вывода локализованных данных от одного или более из множества компонентов локализации на основании предварительно заданных критериев и данных с устраненной неоднозначностью. 3 н. и 11 з.п. ф-лы, 6 ил.



Фиг. 2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21)(22) Application: 2007133808/08, 08.03.2006

(24) Effective date for property rights:
08.03.2006

Priority:

(30) Priority:
08.03.2005 US 60/659,616
29.04.2005 US 11/118,544

(43) Application published: 20.03.2009 Bull. 8

(45) Date of publication: 27.06.2011 Bull. 18

(85) Commencement of national phase: 10.09.2007

(86) PCT application:
US 2006/012901 (08.03.2006)

(87) PCT publication:
WO 2006/132700 (14.12.2006)

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

(72) Inventor(s):

**KOL'MAJER Bernkhard (US),
BRAUNELL Lori A. (US),
VU Vej (US),
E Shehnhua (Ehdvard) (US),
MARTI Dzhordi Mola (US),
NEL'SON Jan Anders (US),
EhL'-GAMMAL Mokhammed (US),
BENNET Dzhuli D. (US)**

(73) Proprietor(s):

MAJKROSOFT KORPOREJShN (US)

(54) RESOURCE CREATION USING ONTOLOGY

(57) Abstract:

FIELD: information technology.
SUBSTANCE: localisation platform includes: resource management devices, multiple localisation components, an ontology array which stores ontology information; and component for detecting coincidences in translations made with possibility of receiving a localisation request and one or more resources from the resource management device, wherein said one or more resources contain resources from each of the multiple applications meant for localisation, obtaining access to multiple

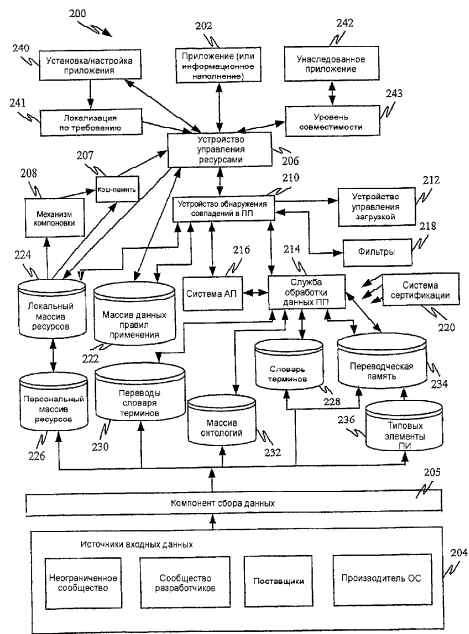
localisation components and the ontology array in order to eliminate data ambiguity associated with specific resources meant for localisation based on the localisation request, and selection and output of localised data from one or more of multiple localisation components based on predetermined criteria and data with the eliminated ambiguity.

EFFECT: faster operation owing to easier localisation of information content of program products.

14 cl, 6 dwg

RU 2 4 2 2 8 9 0 C 2

RU 2 4 2 2 8 9 0 C 2



Фиг. 2

УРОВЕНЬ ТЕХНИКИ

Настоящее изобретение касается обработки данных.

Локализация представляет собой процесс изменения продуктов или служб с учетом различий для отдельных рынков сбыта. Самый распространенный пример
5 локализации имеет место, когда приложение создается на исходном языке, а внедряется на рынке сбыта, который использует целевой язык, отличный от языка оригинала. Например, если приложение было создано на английском языке, а затем внедряется в Китае, то оно должно быть локализовано путем перевода
10 разнообразных англоязычных строк и элементов ПИ (пользовательского интерфейса) (меню, пиктограммы и т.д.), программных компонентов и справки пользователя в приложении на китайский язык. Компонировка и стиль ПИ (в том числе тип шрифта, размер, управляющие позиции и т.д.) тоже, вероятно, будут изменены, чтобы
15 соответствовать целевому языку. Конечно, понятие локализации является более широким, чем простое изменение языка. Рынки сбыта, которые используют один и тот же язык, могут выделяться по другим соображениям. Например, приложение может быть "локализовано" для различных возрастных групп. Оно может обладать одной конфигурацией слога языка и визуального представления для взрослых и другой для
20 детей или подростков. Таким образом, локализация осуществляет иллюстративное приспособление к широкому спектру различий на отдельных рынках сбыта.

Несмотря на то что, как описано выше, локализация затрагивает намного большее, чем просто переведенные строки, для облегчения понимания документа, мы
сосредоточимся на описании этого сценария. Аналогично, большинство примеров
25 взято из области локализации программного обеспечения, но настоящее изобретение не ограничивается локализацией программного обеспечения. Точно так же, в то время как большая часть настоящего обсуждения сосредотачивается на локализации продукта, настоящее изобретение этим не ограничивается и столь же применимо к
30 службам, и, соответственно, термин "продукт" включает в себя "службы" для этого описания. В прошлом не предпринималось попыток обеспечения локализации на системном уровне, а вместо этого локализация выполнялась, используя конкретные компоненты для решения конкретных задач. Например, устройство локализации может быть дополнено определенными технологиями автоматического перевода,
35 чтобы повысить скорость, точность и согласованность локализации. Однако другое устройство локализации может использовать только системы переводческой памяти для того, чтобы увеличить коэффициент повторного использования, многократно используя предыдущие переводы, тем самым обеспечивая наиболее подходящий
40 результат с более высокой скоростью, чем без инструментальных средств.

Кроме того, все это делается обычно в автономном режиме. Другими словами, автор создает приложение целиком или большую часть компонента, и такой компонент или приложение предоставляется устройству локализации для локализации.
Обычный процесс является таким, что взаимодействие устройства локализации с
45 первым автором является минимальным или отсутствует. Это делает его трудным для внесения изменений в первоначальное информационное наполнение, что могло бы облегчить его локализацию. Фактически во многих случаях локализация не выполняется до тех пор, пока весь продукт не будет разработан и не будет принято
50 стратегическое сбытовое решение о распространении этого продукта на другом рынке сбыта, который использует другой язык или отличается в другом отношении. Во всех этих случаях, обычно при разработке приложения предельвается очень небольшая предварительная работа, направленная на локализацию или хотя бы на оптимизацию

локализации. Точно так же присутствует очень незначительная поддержка на стадии разработки/создания для разработки приложения или другого продукта/службы, которые будет относительно просто локализовать, даже при том, что создание продукта или службы, которые легко поддаются локализации, не является более сложной, чем создание продукта или службы, которые не являются простыми для локализации.

Следовательно, продукты, службы и приложения обычно переводятся на различные языки или иным образом локализируются в результате сложного, ручного и трудоемкого процесса. Стоимость этой локализации программных продуктов и перевод связанного с продуктом информационного наполнения представляют собой существенное препятствие, которое должно быть преодолено для того, чтобы выйти на новые рынки сбыта. Это особенно относится к малым и средним независимым поставщикам программного обеспечения или к авторам информационного наполнения.

Задача локализации также пропорционально меняется в зависимости от конкретной территории, на которой разрабатывается программное обеспечение. Для разработчиков, которые создают программное обеспечение на территориях, которые имеют большие рынки сбыта, локализация программного обеспечения для других (и, вероятно, более малых) рынков сбыта не является большой необходимостью. Однако если разработчик работает на территории (и использует язык), которая имеет относительно малый рынок сбыта, общая жизнеспособность продукта может зависеть от возможности локализовать этот продукт для языков, используемых на больших рынках сбыта. Это требует от производителей затрачивать чрезмерно большой объем ресурсов на локализацию. Часто это отнимает ресурсы, имеющиеся для разработки.

Другой задачей, связанной с предварительными мерами по локализации, является то, что не было хорошего способа привлечь к работе множество дополнительных источников локализации. Например, широкий спектр поставщиков локализируют свои продукты для различных рынков сбыта. Аналогичные приложения, разработанные различными поставщиками, могут быть применимы для локализации совпадающих, или очень похожих, строк или программного обеспечения для одних и тех же рынков сбыта. Однако в настоящее время не существует подходящего способа для двоих, чтобы заимствовать, или совместно использовать, работу друг друга. Следовательно, имеется много повторяющихся усилий при локализации продуктов.

Более того, существует много различных моделей программирования (таких, как Win32 (32-разрядный интерфейс прикладного программирования для Windows), CLR (Common Language Runtime - общезыковая исполняющая среда) и написание сценариев для глобальной гипертекстовой сети) с различными типами устройств управления ресурсами, форматов ресурсов и массивов данных. Им необходимы различные синтаксические анализаторы и наборы инструментальных средств для управления локализацией, что приводит к сложным и дорогостоящим процессам и несоответствующему качеству локализации.

РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

Настоящее изобретение предоставляет систему для обработки данных.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Фиг. 1 является одним иллюстративным вариантом осуществления окружения, в котором может использоваться настоящее изобретение.

Фиг. 2 является более детализированной структурной схемой среды выполнения с применением аспектов настоящего изобретения.

Фиг. 3 является блок-схемой, иллюстрирующей общее функционирование платформы (или системы), показанной на Фиг. 2, в которой перевод является недоступным.

Фиг. 4 является блок-схемой, иллюстрирующей функционирование компонента обнаружения совпадений, показанного на Фиг. 2.

Фиг. 5 является более детализированной структурной схемой платформы (или системы) локализации, показанной на Фиг. 2, которая используется в процессе разработки приложения или другого информационного наполнения.

Фиг. 6 является блок-схемой, иллюстрирующей, как продукт может быть разработан, реализуя технологии, которые помогают при последующей локализации в соответствии с одним вариантом осуществления настоящего изобретения.

ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

Настоящее изобретение предоставляет операционную систему с интегрированной платформой локализации, или платформу локализации, которая имеет компоненты, тесно взаимодействующие с операционной системой, хотя и не полностью интегрированную. Настоящая платформа локализации также может предоставляться в виде службы, которая вызывается операционной системой. Настоящее изобретение использует платформу локализации для локализации информационного наполнения и программного обеспечения. Настоящее изобретение также может использоваться в процессе разработки продуктов или служб, реализуя практические методики, которые предоставляют возможность более быстрой и более эффективной локализации продукта или службы. Настоящее обсуждение будет продолжено главным образом относительно перевода во время процесса локализации, но настоящее изобретение этим не ограничивается, и локализация, кроме этого, включает в себя широкое разнообразие других адаптаций для различных рынков сбыта. Прежде чем давать более подробное описание настоящего изобретения, будет описан один вариант осуществления окружения, в котором может использоваться настоящее изобретение.

Фиг. 1 иллюстрирует пример подходящего вычислительного системного окружения 100, в котором может быть реализовано настоящее изобретение. Вычислительное системное окружение 100 является только одним примером подходящего вычислительного окружения и не подразумевает наложения какого-либо ограничения относительно области применения или функциональных возможностей настоящего изобретения. Также вычислительное окружение 100 не должно пониматься как обладающее какой-либо зависимостью или требованиями относительно любого компонента или комбинации компонентов, демонстрирующихся в иллюстративном операционном окружении 100.

Настоящее изобретение действительно для многих других вычислительных системных окружений или конфигураций, общего назначения или специального назначения. Примерами известных вычислительных систем, окружений и/или конфигураций, которые могут быть подходящими для использования с настоящим изобретением, включают в себя, но не ограничиваются этим, персональные компьютеры, обслуживающие компьютеры, переносные или портативные устройства, многопроцессорные системы, микропроцессорные системы, телевизионные приставки, программируемую бытовую электронику, сетевые ПК, мини-компьютеры, универсальные компьютеры, системы телефонной связи, распределенные вычислительные окружения, которые включают в себя любое из вышеупомянутых систем или устройств, и тому подобное.

Настоящее изобретение может быть описано в общем контексте компьютерных

исполняемых инструкций, таких как программные модули, исполняемые компьютером. Как правило, программные модули включают в себя процедуры, программы, объекты, компоненты, структуры данных и т.д., которые выполняют конкретные задачи или реализуют конкретные абстрактные типы данных. Настоящее изобретение предназначено для практического использования в распределенных вычислительных окружениях, когда задачи выполняются удаленными обрабатывающими устройствами, которые связаны при помощи сети передачи данных. При распределенном вычислительном окружении программные модули размещаются в компьютерных запоминающих средах и локального и удаленного компьютера, включающих в себя запоминающие устройства хранения.

Что касается Фиг. 1, иллюстративная система для реализации настоящего изобретения включает в себя вычислительное устройство общего назначения в виде компьютера 110. Компоненты компьютера 110 могут включать в себя, но не ограничиваясь этим, обрабатывающее устройство 120, системную память 130 и системную шину 121, которая соединяет различные системные компоненты, в том числе системную память с обрабатывающим устройством 120. Системная шина 121 может быть любого из нескольких типов шинных структур, включающих в себя шину памяти или устройство управления памятью, шину периферийных устройств и локальную шину, использующую любую из множества шинных архитектур. В качестве примера, но не ограничиваясь этим, такие архитектуры включают в себя шину архитектуры промышленного стандарта (ISA - Industry Standard Architecture), шину микроканальной архитектуры (MCA - Micro Channel Architecture), шину расширенной архитектуры промышленного стандарта (EISA - Enhanced ISA), локальную шину стандарта Ассоциации по стандартам в области видеoeлектроники (VESA - Video Electronics Standards Association) и шину соединения периферийных устройств (PCI - Peripheral Component Interconnect), также известную как шина расширения.

Компьютер 110 обычно включает в себя множество компьютерочитаемых сред. Компьютерочитаемые среды могут быть любыми имеющимися в распоряжении средами, к которым может получить доступ компьютер 110, и включают в себя и энергозависимые и энергонезависимые среды, и съемные и несъемные среды. В качестве примера, но не ограничиваясь этим, компьютерочитаемые среды могут содержать компьютерные запоминающие среды и среды передачи данных.

Компьютерные запоминающие среды включают в себя и энергозависимые и энергонезависимые, и съемные и несъемные среды, реализованные для любых способа или технологии хранения информации, такой как компьютерочитаемые инструкции, структуры данных, программные модули или другие данные. Компьютерные запоминающие среды включают в себя, но не ограничиваются этим, ОЗУ, ПЗУ, ЭСППЗУ, флэш-память или другую технологию памяти, компакт-диски (CD-ROM), цифровые диски универсального назначения (DVD) или другое запоминающее устройство на оптическом диске, магнитные кассеты, магнитную ленту, запоминающее устройство на магнитных дисках или другие магнитные устройства хранения, или любую другую среду, которая может использоваться для хранения необходимой информации и к которой может получить доступ компьютер 110. Среда передачи данных обычно объединяют компьютерочитаемые инструкции, структуры данных, программные модули или другие данные в модулированный сигнал передачи данных, такой как несущая волна или другой транспортный механизм, и включают в себя любые среды доставки информации. Термин "модулированный сигнал передачи

данных" обозначает сигнал, у которого одна или более из его характеристик установлены или изменены таким образом, чтобы закодировать информацию в сигнале. В качестве примера, но не ограничиваясь этим, среды передачи данных включают в себя проводные среды, такие как проводная сеть или прямое проводное соединение, и беспроводные среды, такие как звуковые волны, радиочастоты, инфракрасное излучение и другие беспроводные среды. Комбинации любого из вышеупомянутого также должны входить в понятие компьютерочитаемых сред.

Системная память 130 включает в себя компьютерную запоминающую среду в виде энергозависимой и/или энергонезависимой памяти, такой как постоянное запоминающее устройство (ПЗУ) 131 и оперативное запоминающее устройство (ОЗУ) 132. Базовая система 133 ввода/вывода (БСВВ), содержащая основные процедуры, которые помогают передавать информацию между элементами внутри компьютера 110, например, во время запуска, обычно хранится в ПЗУ 131. ОЗУ 132 обычно содержит данные и/или программные модули, к которым обрабатывающее устройство 120 получает непосредственный доступ и/или которыми оно в настоящий момент оперирует. В качестве примера, но не ограничиваясь этим, Фиг. 1 иллюстрирует операционную систему 134, прикладные программы 135, другие программные модули 136 и программные данные 137.

Кроме того, компьютер 110 может включать в себя другие съемные/несъемные энергозависимые/энергонезависимые компьютерные запоминающие среды. Только в качестве примера, Фиг. 1 иллюстрирует накопитель 141 на жестком диске, который осуществляет считывание или запись на несъемные, энергонезависимые магнитные среды, накопитель 151 на магнитном диске, который осуществляет считывание или запись на съемный, энергонезависимый магнитный диск 152, и накопитель 155 на оптическом диске, который осуществляет считывание или запись на съемный энергонезависимый оптический диск 156, такой как CD-ROM или другие оптические среды. Другие съемные/несъемные, энергозависимые/энергонезависимые компьютерные запоминающие среды, которые могут использоваться в иллюстративном операционном окружении, включают в себя, но не ограничиваются этим, кассеты с магнитной лентой, платы флэш-памяти, цифровые диски универсального назначения, ленту с цифровой видеозаписью, твердотельное ОЗУ, твердотельное ПЗУ и тому подобное. Накопитель 141 на жестком диске обычно подсоединяется к системной шине 121 при помощи интерфейса несъемного запоминающего устройства, такого как интерфейс 140, а накопитель 151 на магнитном диске и накопитель 155 на оптическом диске обычно подсоединяется к системной шине 121 посредством интерфейса съемного запоминающего устройства, такого как интерфейс 150.

Накопители и связанные с ними компьютерные запоминающие среды, обсуждаемые выше и проиллюстрированные на Фиг. 1, обеспечивают хранение компьютерочитаемых инструкций, структур данных, программных модулей и других данных для компьютера 110. На Фиг. 1, например, накопитель 141 на жестком диске проиллюстрирован как хранящий операционную систему 144, прикладные программы 145, другие программные модули 146 и программные данные 147. Заметим, что эти компоненты могут или совпадать или отличаться от операционной системы 134, прикладных программ 135, других программных модулей 136 и программных данных 137. Операционной системе 144, прикладным программам 145, другим программным модулям 146 и программным данным 147 в данном описании присвоены другие ссылочные значения, чтобы пояснить, что, как минимум, они

являются другими копиями.

Пользователь может вводить команды и информацию в компьютер 110 при помощи устройств ввода, таких как клавиатура 162, микрофон 163 и координатно-указательное устройство 161, например манипулятор мышь, шаровой манипулятор или сенсорная панель. Другие устройства ввода (не показаны) могут включать в себя рычажный указатель, игровой планшет, антенну спутниковой связи, сканирующее устройство или тому подобное. Эти и другие устройства ввода часто соединяются с обрабатывающим устройством 120 при помощи интерфейса 160 пользовательского ввода, который подсоединяется к системной шине, но могут соединяться посредством других интерфейса и шинных структур, таких как параллельный порт, игровой порт или универсальная последовательная шина (USB - universal serial bus). Монитор 191 или другой тип устройства отображения также соединяется с системной шиной 121 через интерфейс, такой как видеоинтерфейс 190. В дополнение к монитору компьютеры могут также включать в себя другие периферийные устройства вывода, такие как динамики 197 и устройство 196 печати, которые могут быть подсоединены при помощи интерфейса 195 периферийных устройств вывода.

Компьютер 110 функционирует в сетевом окружении, используя логические соединения с одним или более удаленными компьютерами, такими как удаленный компьютер 180. Удаленный компьютер 180 может быть персональным компьютером, переносным устройством, обслуживающим компьютером, маршрутизатором, сетевым ПК, равноправным устройством или другим публичным сетевым узлом и, как правило, включает в себя многие или все элементы, описанные выше касательно компьютера 110. Логические соединения, изображенные на Фиг. 1, включают в себя локальную вычислительную сеть (ЛВС) 171 и глобальную вычислительную сеть (ГВС) 173, но могут также включать в себя другие сети. Такие сетевые окружения являются обычными в офисах, корпоративных вычислительных сетях, внутренних сетях и глобальной сети на базе технологии Интернет.

При использовании в сетевом окружении ЛВС компьютер 110 соединяется с ЛВС 171 при помощи сетевого интерфейса или сопрягающего устройства 170. При использовании в сетевом окружении ГВС компьютер 110, как правило, включает в себя модем 172 или другое средство для установления связи по ГВС 173, такой как Интернет. Модем 172, который может быть внутренним или внешним, может подсоединяться к системной шине 121 через интерфейс 160 пользовательского ввода или по другой подходящей схеме. В сетевом окружении программные модули, изображенные касательно компьютера 110, или их части, могут храниться на удаленном запоминающем устройстве хранения. В качестве примера, но не ограничиваясь этим, Фиг. 1 иллюстрирует удаленные прикладные программы 185 как постоянно хранящиеся на удаленном компьютере 180. Нужно понимать, что показанные сетевые соединения являются иллюстративными и может использоваться другое средство установления соединения между компьютерами.

Фиг. 2 иллюстрирует платформу (или систему) 200 локализации, которая интегрирована в операционную систему 134, показанную на Фиг. 1. Платформа локализации 200 изображена с некоторым количеством различных запоминающих устройств, и нужно понимать, что эти запоминающие устройства могут быть воплощены в любом из множества устройств хранения, которые показаны на Фиг. 1, или в других устройствах хранения. Система 200 изображена во взаимодействии с приложением 202, которое будет подвергаться локализации, и с множеством источников 204 входных данных, которые вводят различные переводы в массивы

данных в системе 200 при помощи компонента 205 сбора данных.

Сама система 200 включает в себя устройство 206 управления ресурсами, механизм 208 компоновки, устройство 210 обнаружения совпадений в переводческой памяти (ПП), устройство 212 управления загрузкой, службу 214 обработки данных переводческой памяти (ПП), систему 216 автоматического перевода (АП), фильтры 218, систему 220 сертификации и множество различных массивов данных. Массивы данных, показанные на Фиг. 2, включают в себя массив 222 данных правил применения, локальные и персональные массивы 224 и 226 соответственно, массив 228 данных словаря терминов и массив 230 данных переводов словаря терминов, массив 232 онтологий, переводческую память 234 и массив 236 данных типовых элементов ПИ. В общем случае, правила применения используются ресурсами, чтобы определить связанные с приложением обогащенные данные, в то время как фильтры используются, чтобы гарантировать, что модули (или источники 204 входных данных), вносящие вклад в данные, имеют возможность изменять только данные, внесенные ими, и что все добавленные данные отфильтрованы, исходя из определенных правил. Конечно, массивы данных могут быть сгруппированы, иным образом объединены или разделены, любым заданным способом. Функционирование системы 200 подробно описывается ниже в настоящем описании изобретения со ссылками на Фиг. 2-6. Тем не менее, приведем здесь краткое обсуждение функционирования системы 200 и элементов, показанных на Фиг. 2.

Во время рабочего цикла устройство 206 управления ресурсами загружает ресурсы, относящиеся к приложению 202, в кэш-память 207 (подобные сценарии могут встречаться для информационного наполнения и другого типа локализуемых данных). Кроме того, устройство 206 управления ресурсами принимает заявки на ресурсы от приложения 202. В ответ устройство 206 управления ресурсами сначала загружает строку на языке, который соотносится с языковой настройкой, в настоящее время выбранной платформой 200. Устройство 206 управления ресурсами решает, какой вариант ресурса загрузить, основываясь на множестве критериев, таких как доступность, степень достоверности, пользовательские параметры настройки, и т.д. Устройство 206 управления ресурсами может принять решение вызвать устройство 210 обнаружения совпадений в ПП. Чтобы сделать это, устройство 206 управления ресурсами обращается к устройству 210 обнаружения совпадений в ПП для получения перевода (или локализации) указанного ресурса. Устройство 210 обнаружения совпадений в ПП также может иметь предварительно заполненные данные ресурсов, чтобы предотвратить задержки во время выполнения загрузки ресурсов; предварительно заполненная информация будет помечена метками с соответствующими сведениями, например, исходила ли она от системы 216 АП или от устройства 210 обнаружения совпадений в ПП, степень достоверности ее источника, и т.д. Это обсуждается более подробно ниже. Если устройство 210 обнаружения совпадений в ПП не выдает в результате совпадения для ресурса, устройство 206 управления ресурсами осуществляет каскадный вывод при помощи систем восстановления ресурсов с тем, чтобы перевод мог быть предоставлен на другом запасном языке. Это также описывается более подробно ниже. Кроме того, в одном варианте осуществления, и устройство 206 управления ресурсами, и устройство 210 обнаружения совпадений в ПП обладают системами восстановления. Они могут отличаться друг от друга и, хотя результаты работы устройства 210 обнаружения совпадений в ПП могут влиять на устройство 206 управления ресурсами, устройство 206 управления ресурсами может принять решение о восстановлении без

обращения к устройству 210 обнаружения совпадений в ПП.

Нередко изменения, сделанные при локализации строк, приводят к изменениям в элементах отображения, содержащих эти строки. Общие примеры включают в себя потребность в большем пространстве (более длинные строки), изменения в движении диалогового окна (языки с записью справа налево в сравнении с языками с записью слева направо) и т.д. Процесс регулирования может происходить во время локализации и/или во время исполнения. Самым распространенным случаем является, когда требуются перегруппировка и изменение размеров элементов в диалоговом окне, и этот процесс называется автоматической компоновкой. Объем и корректность информации, предоставленной во время процесса создания, будут влиять на результаты, полученные во время исполнения.

Устройство 210 обнаружения совпадений в ПП располагается между различными источниками переводов и устройством 206 управления ресурсами, которое запрашивает о совпадении информационного наполнения (или о переводе на другой язык или об иной локализации). Устройство 210 обнаружения совпадений в ПП производит поиск доступных источников (и тех источников, которые являются надежными для приложения 202), чтобы получить перевод, который имеет наивысшую связанную с ним оценку достоверности.

И опять более подробное обсуждение функционирования устройства 210 обнаружения совпадений в ПП изложено ниже со ссылкой на Фиг. 4. Однако, кратко, устройство 210 обнаружения совпадений в ПП имеет доступ к правилам, связанным с приложением или информационным наполнением 202 в массиве 222 данных правил применения, чтобы гарантировать, что независимо от того, какая локализация возвращается устройству 206 управления ресурсами, это не вызывает сбоя в приложении 202. Устройство 210 обнаружения совпадений в ПП сначала производит поиск в локальном массиве 224 данных и персональном массиве 226 данных, чтобы понять, существует ли уже локализация. Локальный массив 224 данных содержит в себе переводы или другие локализации ресурсов, которые уже были локализованы, на тот случай, если запрос, сделанный устройством 206 управления ресурсами, является таким же, как произведенный ранее.

Устройство 210 обнаружения совпадений в ПП также вызывает службу 214 обработки данных ПП. Служба 214 обработки данных ПП имеет доступ к переводческой памяти 234, которая вмещает переводы или другие локализации широкого множества различных сущностей, таких как распространенные элементы пользовательского интерфейса (ПИ), хранящиеся в массиве 236, а также определенных терминов, хранящихся в словаре 228 терминов, а массив 232 онтологий предоставляет контекстную информацию для приложения 202 и соответствующей предметной области.

Если нет уже существующей локализации или в локальных массивах 224 или 226, или в удаленном массиве 234, устройство 210 обнаружения совпадений в ПП может обратиться к доступной при необходимости системе 216 автоматического перевода (АП) (или к одной из множества доступных систем АП), чтобы выполнить автоматический перевод информационного наполнения, которое нужно локализовать. Решение устройства 210 обнаружения совпадений в ПП вызвать систему 216 АП может основываться на пользовательских настройках, доступности перевода, качестве доступных переводов и предоставленных АП переводов и т.д. Система 216 автоматического перевода, в одном варианте осуществления, является какой-нибудь серийно выпускаемой системой АП и только иллюстративно используется для

переводов, если не присутствует совпадений с более высокой степенью достоверности от какого-нибудь из других источников с памятью (таких как служба 214 обработки данных ПП и локальный и персональный массивы 224 и 226 данных). Тот факт, что система 216 АП может быть по существу любой системой АП, или любой системой, способной использовать данные в различных массивах (например, предыдущие переводы) для создания переводов, значительно повышает расширяемость системы, поскольку новые или отличные системы АП легко могут применяться устройством 210 обнаружения совпадений в ПП.

Конечно, расширяемость системы АП является не единственной возможностью расширяемости в системе, а представляет собой только один пример. Некоторые другие примеры расширяемости системы включают в себя программы редактирования при создании информационного наполнения, языковые синтаксические анализаторы, переводческую память от третьей стороны, программы коллективного редактирования, средства создания и разработки, подключение к сетевым службам и т.д. Расширяемость в этих и других областях поддерживается с помощью АРІ (интерфейс прикладного программирования) в платформе 200.

В любом случае каждый из переводческих ресурсов, к которым обращается устройство 210 обнаружения совпадений в ПП, иллюстративно будет предоставляться устройству 210 обнаружения совпадений в ПП с оценкой достоверности (или скоростью перевода, коэффициентом повторного использования или другим подобным критерием), обозначающей достоверность или коэффициент повторного использования, имеющие отношение к запрошенной локализации. Затем устройство 210 обнаружения совпадений в ПП может выбрать локализацию или перевод из надежного источника, который предоставляет перевод, соответствующий самой высокой оценке. После этого переводы загружаются устройством 212 управления загрузкой в локальный массив 224, если они уже не хранятся в нем.

В одном иллюстративном варианте осуществления переводческие входные данные могут предоставляться на различные запоминающие устройства в системе 200 от широкого спектра источников 204 входных данных. Например, поставщики или производители программного обеспечения, разрабатывающие операционные системы, в которые встроена платформа 200 локализации, иллюстративно предоставляют надежные термины на различных запоминающих устройствах. Однако в одном варианте осуществления источники 204 входных данных могут включать в себя неограниченное сообщество и сообщество разработчиков. Неограниченное сообщество позволяет любому, кто выполняет услуги по локализации с использованием системы 200, предоставлять предложенную локализацию части информационного содержимого на соответствующее запоминающее устройство при помощи компонента 205 сбора данных. Также источники 204 входных данных могут включать в себя сообщество разработчиков, которое будет (в противоположность неограниченному сообществу) сохранять переводы, предоставленные только разработчиками, официально признанными таковыми. Таким образом, можно заметить, что платформа 200 дает возможность процессу локализации быть открытым для участия сообщества в очень широких масштабах, а также и настраиваться на гранулированный уровень (например, пользователь решает, что считать надежным, а что нет), удовлетворяющий потребностям пользователя и ожидаемым результатам.

Дополнительно, компонент сбора данных может динамически (во время установки, рабочего цикла, настройки и т.д.) периодически собирать новые данные от множества источников 204 входных данных. Новые данные, конечно, могут быть новыми

данными ресурсов, а также и предварительно локализованными данными или данными переводческой памяти для поддержки новых приложений, новых предметных областей и для выполнения автоматического преобразования запущенной на выполнение операционной системы. Это охватывает возможность самостоятельной 5 настройки и точность системы.

Однако конкретный разработчик приложения, системный администратор или пользователь могут пожелать, чтобы только определенная локализация была назначена надежной. Разработчик может, например, не захотеть доверять 10 информационному наполнению локализации от неограниченного сообщества или от сообщества разработчиков, а только от поставщиков производителей ОС. Это дает возможность разработчику или пользователю устанавливать пределы, в которых платформа 200 делает общедоступными задачи локализации для различных источников. Это может быть установлено даже на построочном уровне.

В одном таком варианте осуществления система 200 включает в себя систему 220 сертификации, которая присваивает сертификаты информационному наполнению, предоставленному определенными источниками. При этом разработчик приложения 202 может пожелать, чтобы надежным считалось только 20 информационное наполнение локализации с присвоенным сертификатом. Например, разработчики и другие участники работ могли бы иметь возможность получить право подписывать свое информационное наполнение, которое относится к уровню, на котором они находятся, относительно их приложения. Это означает, что сертификат разработчика приложения помещает этого разработчика на уровень разработчика 25 для этого определенного приложения, а все другие участники работ находятся на один уровень ниже в "других независимых поставщиках", на уровне сообщества или пользователя. Сертификаты могут быть надежными относительно уровня участника работ. Например, заданный производитель мог всегда иметь высший уровень, но ниже этого уровня пользователь может определить надежный уровень и в пределах уровня надежных источников. Указание того, какое информационное наполнение локализации должно быть объявлено надежным каким-либо заданным 30 приложением 202 или каким-либо пользователем, может храниться в массиве 222 данных правил применения. Сертификаты помогают ввести доверительную систему для переводов.

Кроме того, пользователь или разработчик могут выбирать иерархические надежные источники. Например, основываясь на первоисточнике информационного наполнения локализации, пользователь может решить, считать ли надежным только 40 производителя ОС, а если он недоступен, тогда считать надежными переводы независимых поставщиков программного обеспечения, а если таковые недоступны, тогда считать надежным только сообщество разработчиков, и так далее. В одном иллюстративном варианте осуществления эта иерархия установлена по умолчанию, но пользователь может легко принять или отменить ее в персональных параметрах 45 настройки. Такие параметры настройки могут быть реализованы в правилах применения, связанных с каждым приложением или с отдельной машиной, выполняющей приложение.

Кроме того, нужно заметить, что персональный массив 226 аналогичен локальному массиву 224 в том, что он постоянно хранится в массиве данных пользователя. Однако локальный массив 224 является иллюстративным общесистемным массивом, в то время как персональный массив 226 является массивом, который может 50 использоваться для указания надежного информационного наполнения локализации,

соответствующего отдельному пользователю, включающим в себя данные, введенные пользователем для своего собственного персонального использования.

Следовательно, каждый отдельный пользователь может идентифицировать различное надежное информационное наполнение локализации. Это информационное
5 наполнение отождествляется с этим отдельным пользователем в персональном массиве 226. Это дает возможность отдельному пользователю дополнительно настроить процесс локализации, помимо настроек, связанных с отдельным приложением. Например, пользователь может сделать изменения в его или ее
10 собственных приложениях, которые необходимы и допустимы для приложения. Также пользователь может в ответ предоставить сообществу для совместного использования локальные или персональные массивы.

В добавление, определенные переводы могут быть заблокированы с тем, чтобы они не изменялись. Это может быть критически важным для того, чтобы избежать
15 возможных намеренно неверных переводов. Например, один из источников входных данных мог предоставить намеренно неверный перевод, который переводит выражение "Нажмите для форматирования вашего жесткого диска" как "Нажмите для продолжения".

Как только устройство 210 обнаружения совпадений в ПП получает требуемую локализацию для запроса от устройства 206 управления ресурсами, оно применяет фильтры 218 и правила применения из массива 222. Фильтры 218 иллюстративно определяют термины, которые не должны заноситься ни в какое информационное
20 наполнение локализации ни на каком заданном языке. Например, некоторые пользователи могут пожелать отфильтровывать непристойные выражения или любое другое нежелательное информационное наполнение. Поэтому фильтры 218 применяются до использования локализованного информационного наполнения
25 запрошенного устройством 206 управления ресурсами. Заметим, что применение фильтров (также как и любого другого вида проверки правильности) является неким этапом локализации и может применяться на многих этапах процесса, таких как редактирование, загрузка, использование и т.д.

Например, проверка правильности в большинстве случаев происходит всякий раз, когда два компонента обмениваются данными, за исключением тех случаев, когда
35 компоненты являются полностью надежными друг для друга, и, кроме того, канал является надежным (например, по отношению к скрытым действиям). Поэтому, например, если ресурсы загружаются из надежного источника и ресурсы содержат достоверную подпись, нет необходимости в другой проверке правильности (кроме проверки достоверности подписи). Если источник является надежным, но файл не
40 подписан, тогда оценивается канал, используемый для транспортировки информации, чтобы определить, является ли он также надежным. Например, неподписанный файл, который был загружен из сети Интернет, с большей вероятностью будет испорченным, чем неподписанный файл, который пользователь скопировал из другой
45 папки на том же самом компьютере. В зависимости от нескольких показателей может быть принято решение выполнить быструю функциональную проверку правильности, или может быть проведена некоторая дополнительная проверка правильности информационного наполнения, такая как проверка на оскорбительные слова, или для
50 уверенности, что термины первоначального ресурса представлены на целевом ресурсе (например, чтобы "меню" не было переведено как "папка").

Компонент 240 установки/настройки приложения используется во время установки и настройки после установки. Во время установки приложение 202 регистрирует свои

ресурсы в локальном массиве 224 ресурсов. Компонент 240 иллюстративно показывает пользовательский интерфейс, который дает возможность пользователю локализовать приложение для целевого языка, поддерживаемого платформой, если оно уже не локализовано. Пользовательский интерфейс может разрешить
5 пользователю явно выбрать целевой язык. Компонент 241 локализации по требованию пересылает эти запросы устройству 206 управления ресурсами, а также, возможно, устройству 210 обнаружения совпадений в ПП.

Унаследованное приложение 242 представляет собой приложения, которые были
10 разработаны без осведомления о платформе 200. Уровень 243 совместимости поддерживает существующие интерфейсы ресурсов, используемых приложением 242, и переадресовывает заявки устройству 206 управления ресурсами.

Фиг. 3 является блок-схемой, которая иллюстрирует функционирование
15 платформы 200 более подробно в соответствии с одним вариантом осуществления настоящего изобретения. Во время работы пользователь сначала выбирает или запускает приложение 202. Это обозначено блоком 300 на Фиг. 3. Затем, в этом варианте осуществления, устройство 206 управления ресурсами загружает
20 необходимые ресурсы, связанные с приложением 202, в кэш-память 207 (или в другое запоминающее устройство, все они будут именоваться в настоящем описании как "кэш", в данном контексте). Это обозначено блоком 302.

В фоновом режиме операционная система (а именно платформа 200 локализации) начинает переводить ресурсы в кэше 207 на требуемый язык. При этом платформа 200
25 реализует правила 222 применения, связанные с приложением 202, и доверительную модель (т.е. надежные источники локализации), связанную с приложением 202 или пользователем (что бы ни применялось). Устройство 206 управления ресурсами вызывает устройство 210 обнаружения совпадений в ПП для ресурсов, хранящихся в кэше 207, чтобы получить их локализованное значение. Локализация ресурсов,
30 загруженных в кэш 207, в фоновом режиме обозначена блоком 304 на Фиг. 3.

Также нужно заметить, что начальная загрузка локализованных ресурсов для приложения не должна дожидаться рабочего цикла. Вместо этого она может быть
35 сделана во время установки или в другое время. В дополнение, некоторые части локализации могут быть запущены на выполнение в рабочем цикле, в то время как другие могут быть совершены при установке или в другое время. Например, некоторые из наиболее затратных в вычислительном отношении или требующих
40 много времени задач могут быть выполнены в фоновом режиме, во время установки или в другое время, в то время как более быстрореализуемые задачи могут быть выполнены в рабочем цикле. В еще одном варианте осуществления задачи могут быть назначены (пользователем) на выполнение в какое-либо определенное время, и, кроме того, состояние этих задач может отображаться.

Поскольку ресурсы локализуются в фоновом режиме, локализованные значения хранятся не только в кэше 207, но также они размещаются в локальном массиве 224.
45 Поэтому при следующем запуске приложения 202 процесс локализации будет происходить намного быстрее, потому что все те значения ресурса, которые были локализованы, будут уже размещены в локальном массиве 224. Таким образом, устройству 210 обнаружения совпадений в ПП не придется искать службу 214
50 обработки данных ПП или получать значения, переведенные с использованием системы 216 автоматического перевода, потому что локализации будут храниться в памяти системы. Вместо этого они просто могут быть извлечены из локального массива 224. Сохранение переводов в локальном массиве 224 обозначено блоком 306

на Фиг. 3. Кроме того, по запросу все переводы могут быть загружены в локальный массив 224 с использованием устройства 212 управления загрузкой.

Конечно, настоящая система может также проверять обновления. Например, изначально мог быть выбран неидеальный перевод, поскольку ничего лучшего не было обнаружено. Однако впоследствии более совершенный перевод может быть загружен на одно из запоминающих устройств одним из источников 204. Тогда ресурс может периодически обновляться для приведения его в соответствие.

Точно так же обновления кода могут сделать ранее локализованные ресурсы недействительными. Поэтому в одном варианте осуществления изменения кода инициируют повторную локализацию некоторых или всех ресурсов.

По мере того, как образуются переводы ресурса, механизм 208 компоновки идентифицирует определенные переводы, для которых потребуются изменения в компоновке при отображении для приложения. В дополнение, шрифты, используемые в исходном языке в приложении, отображаются в шрифты в целевом языке. Это может быть выполнено с помощью механизма 208 компоновки или другими компонентами в рабочем цикле или заранее. Компоновка и подстановка шрифтов обозначены блоком 308 на Фиг. 3.

Таким образом, платформа 200 локализации в операционной системе продолжает выполнение локализации ресурсов, связанных с приложением 202 и сохраненных в кэше 207. Может случиться, однако, что приложение осуществляет вызов ресурса, который еще не локализован в фоновом режиме. Такой запрос обозначен блоком 310 на Фиг. 3. Устройство 206 управления ресурсами сначала определяет, был ли запрошенный ресурс уже локализован и сохранен в кэше 207. Это обозначено блоком 312 на Фиг. 3. Если это так, локализация запрошенного ресурса просто возвращается из кэш-памяти 207 приложению 202. Это обозначено блоком 314.

Однако, если ресурс еще не был локализован в фоновом режиме и сохранен в кэше 207, устройство 206 управления ресурсами запрашивает локализацию ресурса у устройства 210 обнаружения совпадений в ПП. Затем устройство 210 обнаружения совпадений в ПП запрашивает локализацию ресурса у различных источников, к которым у него есть доступ, и возвращает локализацию запрошенного ресурса. Вызов устройства 210 обнаружения совпадений в ПП обозначен блоком 316 на Фиг. 3. Функционирование устройства 210 обнаружения совпадений в ПП более подробно описывается со ссылкой на Фиг. 4.

Благодаря локализации ресурсов в кэше 207 в фоновом режиме платформа 200 локализации существенно выходит за рамки систем предшествующего уровня техники. Системы предшествующего уровня техники загружают кэш ресурсами. Однако это приводит к существенным неудобствам при локализации. Если платформа локализации ждет, пока приложение запросит ресурс для того, чтобы локализовать его, динамически вызывая АП, это, вероятно, приведет к нежелательному замедлению в работе приложения 202, или результаты будут ограничены набором алгоритмов, которые могут использоваться в таком ограниченном окружении (требования к рабочим характеристикам могут ограничивать точность). Современная технология автоматического перевода требует приблизительно половину секунды для перевода предложения. В то время как согласно настоящему изобретению обязательно рассматриваются динамические перевод и другой тип локализации, перевод всех ресурсов динамически с использованием системы 216 автоматического перевода, вероятно, приведет к чрезмерному замедлению в работе приложения. Конечно, если значительное число ресурсов было локализовано заранее и сохранено в локальном

массиве 224 или в персональном массиве 226 или если они были сохранены в переводческой памяти 234 и только относительно малая часть ресурсов должна была быть переведена системой 216 АП, тогда локализация содержимого кэша 207 в фоновом режиме не представляет собой значительного риска для работы приложения 202.

В любом случае устройство 210 обнаружения совпадений в ПП продолжает выполнение локализации значений в кэше 207, любым способом, в фоновом режиме и/или динамически. Устройство 210 обнаружения совпадений в ПП продолжает работу, пока все ресурсы не будут локализованы.

Вообще устройство 210 обнаружения совпадений в ПП может работать с одним или более массивами данных. Оно может при желании интегрироваться с другими средствами повторного использования (такими, как устройство обнаружения совпадений в ПП или системы АП). Оно может при желании воспользоваться установленными словами и выражениями, наряду с их переводами и взаимосвязями (онтологиями). Оно может использовать метаданные, чтобы решить, какие части данных в массивах предположительно составляют пространство поиска (например, участок предметной области, автор, статус подтверждения или другие метаданные). Оно может самостоятельно проверить ограничения (правила применения/фильтры) для строки и воспользоваться ими для получения наилучшего соответствия или соответствия, достоверность которого легче подтвердить. Конечно, все эти признаки являются необязательными и при желании могут быть добавлены иные, или эти могут быть изменены или не использоваться в работе устройства обнаружения совпадений в ПП.

Фиг. 4 является блок-схемой, более подробно иллюстрирующей один вариант осуществления функционирования устройства 210 обнаружения совпадений в ПП, только для примера. Сначала устройство 206 управления ресурсами предоставляет устройству 210 обнаружения совпадений в ПП запрос на локализацию. Это обозначено блоком 350 на Фиг. 4. Затем устройство 210 обнаружения совпадений в ПП получает информацию доверительной модели для текущего контекста. В одном варианте осуществления информация доверительной модели хранится с самими данными, но также могла бы храниться в персональном массиве 226 или в каком-нибудь другом источнике, который распознает любые другие элементы контекста, исходя из которых определяются надежные источники локализации при наличии данного контекста. Информация доверительной модели может включать в себя указание на то, какие источники являются надежными, или список требований, которые должны быть удовлетворены для того, чтобы локализованный элемент был правильными и обоснованным. Это помогает также и в сценариях обновления. Обновления в коде могут изменить набор ограничений, которым ресурсы должны удовлетворять. Эти изменения потребуют повторной проверки правильности ранее признанных правильными ресурсов. Это обозначено блоком 352 на Фиг. 4.

Затем устройство 210 обнаружения совпадений в ПП иллюстративно осуществляет запрос на локализацию ресурса ко всем надежным источникам локализации, но может сделать это в заданном порядке. Порядок и число выбранных источников иллюстративно изменяются в зависимости от условий (например, происходит ли локализация в рабочем цикле, во время установки, в пакетной обработке и т.д.). Это обозначено блоком 354. Другими словами, в одном варианте осуществления устройство 210 обнаружения совпадений в ПП даже не запрашивает перевод у ненадежных источников. В этом случае предполагается, что пользователь не желает

доверять переводам от неограниченного сообщества или от сообщества разработчиков в источниках 204 входных данных. Устройство 210 обнаружения совпадений в ПП интересуется только информационным наполнением локализации, которое образовано из надежных источников, и таким образом исключает все
5 информационное наполнение локализации, предоставленное неограниченным сообществом и сообществом разработчиков. Конечно, устройство 210 обнаружения совпадений в ПП будет изменять источники, у которых запрашиваются результаты локализации, исходя из надежных источников с учетом заданного контекста.

10 В ответ на запрос от устройства 210 обнаружения совпадений в ПП различные источники, которые получили запросы на информационное наполнение локализации, возвращают относящиеся к ним оценки степени достоверности. Например, обычно результаты перевода имеют соответствующий уровень достоверности, указывающий, насколько вероятно, что перевод является правильным с учетом входных данных на
15 исходном языке. Для целей настоящего изобретения не важно, какой конкретно показатель используется для указания степени достоверности, связанной с переводом или другой локализацией. В иллюстративном варианте осуществления оценки от всех источников или используют один и тот же показатель или показатели, которые
20 находятся в определенном соотношении друг с другом, так, чтобы устройство 210 обнаружения совпадений в ПП могло определить относительную величину оценок. Возвращение оценок степени достоверности, связанных с переводами (или информационным наполнением локализации), обозначено блоком 356 на Фиг. 4.

25 Затем устройство 210 обнаружения совпадений в ПП получает локализованное информационное наполнение (например, перевод) от надежного источника с наивысшей оценкой. Это обозначено блоком 358. Кроме того, устройство 210 обнаружения совпадений в ПП может выбирать локализованное информационное наполнение, основываясь также на других или дополнительных критериях, таких как
30 время, требуемое для получения локализованного информационного наполнения, насколько точно соответствует локализованное информационное наполнение целевому рынку сбыта, критерии проверки правильности или другие критерии или любая комбинация критериев. Кроме того, устройство обнаружения совпадений в ПП может извлекать информационное наполнение локализации из множественных
35 источников и комбинировать их.

После того как перевод принят, устройство 210 обнаружения совпадений в ПП применяет фильтры 218 и другую проверку правильности, чтобы гарантировать, что
40 информационное наполнение локализации (или перевод) является подходящим для заданных приложения и контекста, и любые фильтры или критерии проверки правильности по желанию пользователя. Применение фильтров и другая проверка правильности обозначено блоком 360 на Фиг. 4. Также устройство 210 обнаружения совпадений в ПП иллюстративно применяет правила применения так, что
45 информационное наполнение локализации, извлеченное для данного ресурса, не будет препятствовать работе приложения или являться причиной некорректных действий или приводить к иному сбою в приложении. Также нужно заметить, что некоторые технологические процессы фильтрации и проверки правильности могут быть заранее
50 рассчитаны, вместо того, чтобы ждать выполнения всех необходимых этапов в этой точке процесса.

После того как это сделано, устройство 210 обнаружения совпадений в ПП возвращает результат устройству 206 управления ресурсами. Это обозначено блоком 362. Нужно понимать, что этот процесс будет выполняться всякий раз, когда

устройство 210 обнаружения совпадений в ПП вызывается устройством 206 управления ресурсами. Следовательно, процесс будет выполняться, когда ресурсы в кэше 207 локализируются в фоновом режиме, и будет выполняться, когда приложение 202 подает заявку на пока еще не локализованный ресурс.

5 И в том и в другом случае устройство 210 обнаружения совпадений в ПП может прийти к заключению, что не существует имеющегося в наличии перевода на целевой язык или возвращенная оценка степени достоверности ниже заданной пороговой величины и что перевод должен быть совершен с использованием системы 216
10 автоматического перевода. На этом этапе, в одном варианте осуществления, устройство 206 управления ресурсами выдает предупреждение и может подать запрос устройству 210 обнаружения совпадений в ПП на поиск перевода на запасном языке. Например, если целевым языком является каталанский язык, приложение 202 может запросить перевод ресурса, но его может не иметься в наличии на каталанском языке.
15 Однако устройство 206 управления ресурсами может быть настроено так, что испанский язык является первым запасным языком по отношению к каталанскому языку. В этом случае устройство 210 обнаружения совпадений в ПП возвращает устройству 206 управления ресурсами извещение, что не имеется в наличии перевода
20 на каталанском языке и что для получения такого перевода должна быть вызвана система 216 автоматического перевода (в качестве альтернативы, обращение к системе 216 АП может происходить по первой заявке от устройства 210 обнаружения совпадений в ПП). Устройство 206 управления ресурсами вместо того, чтобы
25 получить ухудшения рабочих характеристик, связанные с динамическим автоматическим переводом, может просто запросить устройство 210 обнаружения совпадений в ПП вернуть перевод требуемого ресурса на испанский язык. Если перевод имеется в наличии, он возвращается устройством 210 обнаружения совпадений в ПП.

30 Конечно, как указано выше, запасные языковые настройки/языки могут образовывать иерархическую каскадную структуру любого необходимого порядка так, чтобы различные языковые настройки являлись запасными для различных других языковых настроек, причем крайний запасной вариант в этой структуре может быть
35 исходным языком. Таким образом, если достигается крайний запасной вариант, вместо предоставления перевода запрошенного ресурса, устройство 206 распределения ресурсов просто возвращает приложению 202 запрошенный ресурс в крайнем запасном варианте, который обеспечивается платформой, таким образом приложение не будет нарушаться или терять точку взаимодействия с пользователем вследствие
40 работы платформы.

Также пользователь может иллюстративно установить, как работает иерархия запасных вариантов. Например, пользователь может указать, что при совместном отображении группы ресурсов (например, диалоговое окно с несколькими строками)
45 частично диалоговое окно может показываться на одном языке (или быть локализованным для одного рынка сбыта), а частично на другом (например, на запасном). Другие пользователи могут указать, что отображаемая группа ресурсов должна быть вся на одном языке (или быть локализованной для одного и того же рынка сбыта).

50 Заметим также, что от пользователя не требуется запускать приложение на выполнение, чтобы получить его перевод с помощью платформы 200. Пользователь может просто выбрать приложение или другой компонент программного обеспечения для локализации, и платформа 200 может осуществить его локализацию в фоновом

режиме. Например, в одном варианте осуществления пользователь нажимает правую кнопку манипулятора мышь на приложении и из меню выбирает пункт "Локализовать". После этого платформа 200 начинает локализацию приложения в фоновом режиме, сохраняя локализованное информационное наполнение в локальном массиве. Информационное наполнение также может быть локализовано в фоновом режиме. В одном варианте осуществления пользователь нажимает правую кнопку манипулятора мышь на документе, который хранится в файловой системе или на обслуживающем компьютере, и выбирает пункт "Локализовать". Платформа локализации будет выполнять локализацию документа в фоновом режиме. То же самое применимо к документам, которые загружены из сети Интернет.

В дополнение, при начальной настройке приложения (или позднее) пользователь может выбрать множество различных языков для подключения. Тогда эти языки загружаются в локальный массив 224 данных. Переводы могут храниться на прилагаемом компакт-диске или другом носителе.

Фиг. 5 является структурной схемой, иллюстрирующей компоненты платформы 200 в рамках разработки. Многие элементы аналогичны тем, которые показаны на Фиг. 2 и так же пронумерованы.

Фиг. 5 показывает вспомогательную платформу 380 и компонент 382 разработки. Компонент 382 разработки может быть любым подходящим компонентом разработки, таким как Visual Studio, поставляемой компанией Microsoft Corporation, Редмонд, Вашингтон, или любой другой компонент разработки, подходящий для создания программного обеспечения. Вспомогательная платформа 380 помогает в создании информационного наполнения, взаимодействуя с одним или более компонентами платформы 200, например переводческая платформа 381 дает возможность поставщикам поставлять продукт, который имеет высокий коэффициент повторного использования и, следовательно, сильно сниженную стоимость локализации на различные языки. И компонент 382 разработки, и вспомогательная платформа 380 используют службы, предоставляемые платформой 200. При этом компонент 382 разработки и вспомогательная платформа 380 вместе с другими компонентами встраиваются в определенные процедуры на клиентской стороне процесса разработки/создания, что, вероятно, значительно расширит возможности локализуемого продукта при выводе на многие непохожие рынки сбыта и таким образом существенно повысит окупаемость разработки.

Компонент 384 формирования использует информацию, предоставленную платформой локализации и/или вспомогательной платформой 380 или любым другим средством создания информационного наполнения, которое может получить доступ к платформе локализации посредством API и к правилам применения в массиве 222 данных, и формирует приложение 202, а также тексты информационного наполнения.

Также заметим, что настоящее обсуждение Фиг. 5 протекает в основном касательно разработчика, разрабатывающего приложение или автора, создающего информационное наполнение. Однако система может использоваться для множества сущностей независимо от типа информационного наполнения или программного обеспечения, которые разрабатывает разработчик, или создает автор.

В соответствии с одним вариантом осуществления настоящего изобретения существуют два способа для разработчика, или автора соответственно, чтобы использовать компонент 382 разработки для разработки программного обеспечения или вспомогательную платформу 380 для разработки информационного наполнения. Первый - это использовать такую особенность своего окружения

создания/разработки, что при посредстве интерфейса прикладного программирования (API) платформы или другого подобного механизма устанавливается, было ли уже какое-нибудь подобное информационное наполнение или программное обеспечение разработано и сохранено на каком-либо из 5 запоминающих устройств платформы 200. Вторым способом для разработчика является просто разрабатывать программное обеспечение или информационное наполнение (оригинальные или нет). В последнем случае платформа 200 просто предоставляет разработчику обратную связь в отношении коэффициента повторного 10 использования (например, объем информационного наполнения/программного обеспечения, которое может быть переведено на различные языки на платформе 200 с учетом предыдущих переводов) информационного наполнения/программного обеспечения и рекомендует действия, которые возможно повысят коэффициент повторного использования, а также рекомендует действия, которые препятствуют 15 неправильной работе приложения после локализации.

В соответствии с первым вариантом осуществления автор информационного наполнения предоставляет входные данные посредством вспомогательной платформы 380, а вспомогательная платформа 380 обращается к API или другому 20 подобному механизму для создания ресурса. Это обозначено блоком 500 на Фиг. 6. Затем вспомогательная платформа 380 вызывает устройство 210 обнаружения совпадений в ПП для получения доступа к локальному массиву 224 и возвращает все типы ресурсов, для которых уже имеются переводы. Это обозначено блоком 502 на Фиг. 6. Например, платформа разработчика может вызвать API, указывая, что 25 разработчик информационного наполнения хочет создать ресурс. В ответ платформа 380 возвращает указание всех различных классов ресурсов, которые содержатся в платформе 200 и для которых уже есть переводы на множество различных языков. Предположим, например, что результатом, полученным 30 платформой и предоставленным пользователю платформой разработчика, являются классы "меню" и "сообщения об ошибках".

При помощи доступных функциональных средств в API разработчик может выбрать классы "сообщения об ошибках" и вспомогательная платформа 380 35 запрашивает эту информацию у платформы 200 (она может храниться на любом запоминающем устройстве), затем возвращает все различные классы сообщений об ошибках, которые содержатся в платформе 200. Затем разработчик выбирает один из классов сообщений об ошибках и платформа 380 возвращает все конкретные сообщения об ошибках, связанные с этим классом. После этого пользователь может 40 просто выбрать одно из сообщений об ошибках, которое уже было создано и для которого локализация уже будет высокоэффективной (например, потому что оно было уже переведено на множество различных языков или иначе локализовано). Выбор разработчиком типа, класса и определенного ресурса, при наличии такового, обозначен блоком 504 на Фиг. 6.

Конечно, также нужно принимать во внимание, что определенное, точное 45 сообщение, которое разработчик хочет создать, возможно, еще не имеется в наличии. В этом случае пользователь может выбрать наиболее близкое сообщение и изменить его. Измененное сообщение, вероятно, во многих случаях, может быть пригодно для 50 весьма эффективной локализации, поскольку большая его часть уже присутствует в платформе 200 локализации. В любом случае, выбор похожего, не идентичного сообщения приведет к ухудшению достоверности. Изменение выбранного ресурса обозначено блоком 506 на Фиг. 6.

Такой тип повторного использования предоставляет автору более высокую степень гибкости по сравнению с устройством локализации предшествующего уровня техники, которое применяется после разработки. Например, автор может изменить число замещающих элементов в программной строке, в то время как традиционное устройство локализации значительно более ограничено. Кроме того, автор может полностью заменить исходное предложение (при условии, что ключевые термины сохранены), в то время как традиционное устройство локализации не может предоставить такую возможность. Рассмотрим предложение для сообщения об ошибке: "файл {0} не существует". Автор может заменить его на "файл {0} на диске {1} не может быть найден", тогда как традиционное устройство локализации не может использовать перевод первого предложения в качестве перевода для второго, потому что, не имея доступа к коду, традиционное устройство локализации не имеет возможности изменить основные ограничения в коде относительно числа замещающих элементов. Повторное использование может осуществляться не только по отношению к тексту, найденному в сообщениях, но также может быть осуществлено по отношению к общим элементам ПИ в массиве 236 данных. Многократное использование предыдущих элементов ПИ также улучшает коэффициент повторного использования и локализацию продукта.

Во время разработки разработчик может также пожелать предоставления массива 232 соответствия онтологий. Онтология является четким определением концептуализации и описывает взаимосвязи между терминами, которые дают возможность семантического кодирования информационного наполнения и строк. Один вариант осуществления онтологии изложен в патентах США Номер 6253170 и 6098033. Другое описание онтологии изложено в публикации Грубера "A TRANSLATION APPROACH TO PORTABLE ONTOLOGIES", журнал Knowledge Acquisition, 5(2):199-220 (1993).

В общем случае, слова переводятся по-разному в зависимости от контекста. В частности, слова или термины могут быть переведены по-разному в зависимости от контекста приложения при локализации программного обеспечения. Платформа локализации получает доступ к семантическим данным в форме массива 282 данных онтологий, где описывается взаимосвязь между терминами. Онтологии могут быть сгенерированы вручную, путем ввода семантической информации вместе со словом или термином или автоматически, с применением алгоритмов, которые в состоянии определить семантический контекст по близости слов или термина другим словам или терминам. Массив 282 данных иллюстративно наполняется структурами и контекстом приложения, полученными из формата ресурса. Тип семантического кодирования в массиве 282 онтологий предоставляет возможность 200 устранить неоднозначность семантики для ресурсов, которые будут переводиться, а кроме того, предоставляет мощные возможности поиска, в силу того, что учитывает также устранение противоречий в запросе. Переводческая платформа 381 определяет, необходимо ли устранение неоднозначности, в блоке 508 на Фиг. 6. Если это так, можно получить доступ к массиву 232 онтологий, чтобы получить устраняющую неоднозначность семантическую информацию, или вспомогательная платформа 380 может запросить устраняющую неоднозначность семантическую информацию у автора, таким же образом компонент 382 разработки может осуществить запрос разработчику. Это обозначено блоком 510 на Фиг. 6.

Поскольку разработчик продолжает разрабатывать продукт, каждый из созданных ресурсов подвергается оценке коэффициента локализации со стороны переводческой

платформы 381 в платформе 200. При оценке коэффициента локализации созданных ресурсов переводческая платформа 381 определяет процентное соотношение тех ресурсов, которые уже переведены (или иначе локализованы), и тех, которые хранятся где-либо в платформе 200. Кроме того, переводческая платформа 381 учитывает
5 требования для автоматического перевода ресурсов с помощью системы 216, а также принимает во внимание оценки достоверности, связанные с каждым из переводов, хранящимся в платформе 200. Эти элементы комбинируются для предоставления
10 коэффициента повторного использования, который является мерой того, насколько простой будет локализация создаваемого продукта для рынков сбыта, использующих различные языки. Обеспечение выполнения, с суммарным коэффициентом повторного использования, обозначено блоком 512 на Фиг. 6.

Заметим, что коэффициент повторного использования может широко
15 использоваться другими разнообразными способами. Например, если разработчик разрабатывает только компонент целой системы, может потребоваться приведение компонента в соответствие определенному пороговому значению коэффициента повторного использования прежде, чем он будет допущен для включения в систему. Точно так же коэффициент повторного использования вполне может быть
20 показателем ценности продукта с учетом вероятности вывода продукта на другие рынки сбыта. Конечно, этот коэффициент повторного использования может также широко использоваться и другими разнообразными способами.

Также нужно заметить, что система может вычислять коэффициент повторного использования для кода, который не разрабатывается на платформе. Код может
25 просто быть предоставлен платформе 200, и устройство 210 обнаружения совпадений в ПП может вычислить коэффициент повторного использования для кода, в целом, и возратить его, при необходимости. Это может использоваться при принятии решения, разворачивать ли продукт на новых рынках сбыта или же приобрести права
30 на продукт.

Далее обсуждение продолжается относительно варианта осуществления, в котором разработчик не выбирает уже существующие ресурсы. При условии, что разработчик не хочет выбирать из предварительно созданных ресурсов, разработчик может
35 создать оригинальные ресурсы или информационное наполнение. Это обозначено блоком 514 на Фиг. 6. В этом случае разработчик просто создает часть информационного наполнения, а вспомогательная платформа 380 обращается к устройству 210 обнаружения совпадений в ПП для определения коэффициента повторного использования для созданного информационного наполнения. Это
40 обозначено блоком 516 на Фиг. 6. Затем устройство 210 обнаружения совпадений в ПП получает доступ к своим источникам локализации и возвращает коэффициент повторного использования для вновь созданного ресурса или информационного наполнения. Это обозначено блоком 518 на Фиг. 6.

На этом этапе платформа 200 также может использоваться для уведомления автора
45 о действиях, которые могут помочь автору улучшить коэффициент повторного использования находящегося в процессе разработки продукта, или предупредить автора об определенных действиях, которые могут вызвать сбой в приложении после того, как оно будет локализовано. Возвращение этих подсказок обозначено
50 блоком 520 на Фиг. 6. В одном варианте осуществления эти подсказки предоставляются, даже если источник продукта используется многократно. То, что он используется многократно, может, конечно, означать, что некоторые из проверок были уже выполнены.

Некоторые примеры действий, которые могли бы оказать влияние на коэффициент повторного использования или прикладную функцию, включают в себя, например, запись длинных строк. Длинные строки не только труднее для понимания, они также труднее для перевода. Кроме того, программное обеспечение представляет риск переполнения буфера, что приведет к сбою приложения. Другие действия, которые могут оказать влияние на коэффициент повторного цикла или прикладную функцию, включают в себя использование специальных символов или использование некорректной или неформальной грамматики. Использование корректных грамматических конструкций приводит к значительному увеличению возможностей перевода текста. Таким образом, устройство 210 обнаружения совпадений в ПП обеспечивает обратную связь в реальном времени в форме коэффициента повторного использования кода и практических советов.

Кроме того, платформа 200 может предпринять определенные действия для снижения отрицательного влияния этих проблемных участков в анализируемом информационном наполнении. Эти действия снизят вероятность того, что произойдет сбой приложения после проведения локализации. Платформа 200 может предупреждать пользователя о том, что будут предприняты эти корректирующие действия.

Нужно отметить ряд других элементов в отношении настоящего изобретения. Во-первых, это иллюстративное предоставление автоматизации функциональных клавиш. Переводческая платформа 381 иллюстративно отслеживает функциональные клавиши, которые отображаются в любом данном ПИ для того, чтобы предотвратить назначения двух функций на одну и ту же функциональную клавишу в данном ПИ. Переводческая платформа 381 может быть вызвана для автоматического выбора функциональной клавиши, или она может быть вызвана просто для проверки, чтобы гарантировать, что в данном ПИ не происходит дублирования функциональных клавиш. Чтобы иметь возможность выбрать или предложить пользователю функциональную клавишу, платформе требуется информация о том, какие элементы являются одновременно доступными пользователю, с тем, чтобы можно было избежать повторений. Эта информация тесно связана с тем, что требуется для воспроизведения информации на компьютере пользователя.

Настоящее изобретение может также обеспечить проверку правильности и моделирование среды выполнения. После создания компонента разработчик может вызвать API на платформе 382 разработки и платформа 200 смоделирует среду выполнения, в рамках которой требуется локализация. Моделирование рабочего цикла позволяет разработчику физически видеть отображение данных на экране дисплея после локализации. Это дает разработчику возможность осуществить поиск дефектов, оценить эстетическую привлекательность изображения на экране и при необходимости внести изменения.

Таким образом, настоящее изобретение расширяет возможности многократного использования разработчиком уже разработанного информационного наполнения. Это существенно улучшает коэффициент повторного использования и локализацию продукта.

Дополнительно, настоящее изобретение предоставляет систему, которая грамотно комбинирует локализованное информационное наполнение от операционной системы, приложений, сообщества, и третьих сторон, и от автоматического перевода. Устройство управления ресурсами загружает ресурсы приложения, возвращая запрошенные ресурсы, в соответствии с требуемым языком или другими культурными

или рыночными критериями, из множества источников или массивов данных (также именуемых компонентами информационного наполнения локализации).

Кроме того, платформа локализации может использоваться для предоставления локализованных данных широкому множеству объектов, осуществивших запрос, таких как приложения, программы просмотра, поисковые службы, программы загрузки ресурсов, авторские инструментальные средства и т.д.

Кроме того, позволяя сообществу и неограниченному сообществу предлагать варианты локализации, настоящее изобретение дает возможность установить стандарт локализации для сообщества, что иногда облегчает локализацию для некоторых очень малых рынков сбыта. Например, настоящее изобретение позволяет пользователю локализовать приложение на английском языке для языка суахили. Пользователь может предоставить эту локализацию обратно для платформы 200 в качестве предлагаемого перевода различных ресурсов в приложении. Другие южноафриканские пользователи могут выбирать, считать это локализованное наполнение надежным и загрузить его для языка суахили, или локализовать его самостоятельно, или изменить локализацию и предоставить свою собственную локализацию обратно для платформы 200. Таким образом, приложение может быть локализовано, и исправлено, и предоставлено для такого рынка сбыта, на котором иначе оно не могло бы появиться из-за малого объема размера.

В одном варианте осуществления, когда источник 204 входных данных предоставляет данные, компонент 205 сбора данных также записывает метаданные, такие как происхождение, использование, контекстная информация (например, контроль версий ресурса), уровень достоверности, атрибуты, подсказки автоматического перевода и т.д. Другие данные также могут быть собраны и отсортированы. Кроме того, в одном варианте осуществления только источник 204 входных данных, который предоставил данные, может изменять эти данные.

Настоящее изобретение предусматривает не только загрузку переводов от сообщества, но также и передачу данных им. Кроме того, подобным образом могут быть переданы/загружены исходные строки.

Также заметим, что третьи компании и сообщества могут добавлять новые источник и переводческую память в платформу 200. В одном иллюстративном варианте осуществления платформа 200 предоставляет страницу сети или сетевую службу, которая дает возможность добавления нового исходного термина и его метаданных. Пользователи будут иллюстративно проходить проверку подлинности для того, чтобы платформа 200 могла определить источник каждого вновь добавленного элемента.

Кроме того, платформа 200 будет иллюстративно работать в соответствии с распространенной схемой (например, XML-схема), которая дает возможность пользователям передавать несколько строк сразу. Опять же источник каждой строки известен, так как каждая операция может требовать проверки подлинности. Такие исходные строки могут быть иллюстративно добавлены в один или более переводов. Платформа 200 аналогично работает в отношении терминов в базах данных терминов.

Кроме того, в соответствии с одним вариантом осуществления сообщества имеют возможность предоставлять переводы для исходных строк в платформе 200. При редактировании переводов метаданные, имеющиеся в источнике, являются иллюстративно доступными "переводчику" и метаданные применяются до одобрения перевода. Это может быть сделано с помощью простого сетевого интерфейса. Эти переводы могут использоваться другими людьми, если пользователь решит

предоставить для совместного использования сообществом его или ее переводов.

Настоящее изобретение также иллюстративно предоставляет возможность сообществу просматривать переводы, которые уже присутствуют на платформе 200. Это позволяет источнику переводов определить, сколько раз его или ее переводы
5 выбирались другими пользователями. Это даст возможность источникам создать себе имя в сообществе, основанное на качестве работы, и это побуждает источники входных данных выполнять рабочее задание лучше, чтобы стать более ценным в сообществе.

10 В соответствии с еще одним вариантом осуществления настоящего изобретения когда разработчик передает приложение для перевода, разработчику предоставляется возможность позволить конечным пользователям загружать свою собственную локализованную версию продукта. Если эта дополнительная возможность
15 задействована, пользователь может записывать информацию в платформу 200 и указывать, какие язык, механизмы и переводы использовать для того, чтобы сгенерировать собственную версию продукта.

Например, пользователь может запросить "хуз" версию продукта, оговаривая только точные совпадения механизмов перевода, использующих переводческие
20 памяти, одобренные сообществом (или конкретным пользователем). Этот процесс может периодически повторяться для получения значительной доли перевода ресурса. Платформа 200 также может предусматривать уведомления пользователей посредством почтового отправления или другой службы оповещения о том, что
25 доступны новые переводы.

В соответствии с другим вариантом осуществления третья сторона может добавить
30 новый механизм подтверждения подлинности источника (или перевода) в платформу 200. Как обсуждалось выше, разработчик может представить приложение на рассмотрение платформе 200, чтобы определить, насколько приложение пригодно для локализации. Третьи стороны могут написать новые механизмы для поддержки
35 этого. Если третья сторона разрабатывает механизм, который соответствует критериям для выполнения подтверждения подлинности источника (например, когда он реализует требуемый интерфейс), то третья сторона может передать его на платформу 200 и предусмотреть поддержку для пользователей при загрузке
40 механизма. Как и в случае других частей процессов, описанных в отношении настоящего изобретения, добавление подписи и проверка подлинности иллюстративно используются, чтобы гарантировать, что источник механизма известен и пользователь
45 принял решение считать этот источник надежным.

Механизмы перевода могут быть добавлены подобным образом, но могут
40 иллюстративно выполняться на обслуживающем компьютере, реализующем платформу 200 с мерами, принятыми, чтобы гарантировать, что механизм перевода не разрушает никакие уже существующие переводческие данные. Кроме того, настоящее изобретение выполняет глобализацию и лингвистические проверки оказания помощи
45 авторам. Настоящее изобретение также предоставляет возможность сообществу помочь автору. Например, у сообщества можно попросить исходную строку в диалоговом окне для определенной предметной области в случае, если исходная строка может быть легко локализована. Это помогает автору на начальном этапе
50 создания хорошо локализуемого информационного наполнения.

Также нужно заметить, что, в то время как в описанном варианте осуществления определенные функции назначаются определенным компонентам, это не является
требованием. Функции могут быть выполнены различными компонентами, то есть

некоторые компоненты могут быть исключены, изменены или добавлены, всецело в рамках объема настоящего изобретения.

Следовательно, можно видеть, что настоящее изобретение обобщает данные для локализации, исходя из прошлых локализаций широкого спектра продуктов и широкого разнообразия источников входных данных. Вместо того чтобы каждое приложение или продукт имели свою собственную память переводов, к которой не имеют доступа другие продукты, платформа 200 делает эту информацию доступной, по запросу разработчика или пользователя, для последующей локализации различных продуктов. Это также вносит изменения в процесс локализации так, что конечные пользователи получают доступ к более широким возможностям для адаптирования к специальным требованиям (локализации) своих компьютерных работ. Это также предоставляет возможность участия в процессе сообщества. Как следствие, собственно задача локализации становится не зависящей от продукта, связанной с продуктом только при посредстве контекста приложения. Другие приложения могут получить доступ к локализованным строкам в любое время, если доверительная модель и контекст приложения предусматривают это.

Хотя настоящее изобретение было описано в отношении конкретных вариантов осуществления, специалистам в данной области техники будет понятно, что могут быть сделаны изменения в форме и деталях, без отступления от сущности и объема настоящего изобретения.

Формула изобретения

1. Платформа локализации информационного наполнения и/или программного обеспечения, функционально связанная с операционной системой, причем упомянутая платформа локализации выполнена с возможностью обеспечения локализации данных для множества приложений, причем упомянутая платформа локализации содержит:

множество компонентов локализации, предоставляющих локализованные данные, которые локализованы для одного или более отдельных рынков сбыта;

массив онтологии, хранящий информацию онтологии; и

компонент обнаружения совпадений в переводах, выполненный с возможностью приема запроса на локализацию и ввода данных из компонента создания,

используемого разработчиком приложения для разработки множества ресурсов, ассоциированных с приложением, при этом вводимые данные содержат упомянутое множество ресурсов, ассоциированных с приложением;

получения доступа к множеству компонентов локализации и массиву онтологии для устранения неоднозначности данных, ассоциированных с конкретными ресурсами, предназначенными для локализации на основании запроса на локализацию, и выбора и вывода локализованных данных от одного или более из множества компонентов локализации на основании предварительно заданных критериев и данных с устраненной неоднозначностью; и

приема входных данных от компонента создания во время создания приложения, и при этом платформа локализации выполнена с возможностью обеспечения обратной связи с разработчиком приложения, причем обратная связь включает в себя указание коэффициента повторного использования множества ресурсов, при этом коэффициент повторного использования указывает информацию во множестве компонентов локализации, которая уже была локализована.

2. Платформа локализации по п.1, в которой обратная связь включает в себя указание множества различных классов ресурсов, которые содержатся во множестве

компонентов локализации и включают в себя уже локализованные данные.

3. Платформа локализации по п.1, в которой компоненты локализации, массив онтологии и компонент обнаружения совпадений в переводах интегрированы с операционной системой.

5 4. Платформа локализации по п.1, в которой компоненты локализации предоставляют оценки достоверности, ассоциированные с локализованными данными, с учетом входных данных, и в которой компонент обнаружения совпадений в переводах выполнен с возможностью выбора локализованных данных на основании оценок достоверности, при этом выбор включает в себя определение относительной величины оценки достоверности ассоциированной с первым элементом локализованных данных в сравнении с оценкой достоверности, связанной со вторым элементом локализованных данных.

15 5. Платформа локализации по п.1, в которой компоненты локализации содержат: множество массивов данных, хранящих данные, локализованные для одного или более отдельных рынков сбыта.

20 6. Платформа локализации по п.1, в которой компоненты локализации содержат: по меньшей мере, одну систему автоматического перевода, которая обеспечивает определенный вид локализованных данных для выбора упомянутым компонентом обнаружения совпадений в переводах, при этом упомянутый определенный вид представляет собой автоматически генерируемый перевод данных с первого языка на отличающийся второй язык.

25 7. Способ локализации данных, содержащий этапы, на которых: принимают запрос и вводят данные из компонента создания, используемого разработчиком приложения для разработки множества ресурсов, ассоциированных с приложением, при этом вводимые данные содержат упомянутое множество ресурсов, ассоциированных с приложением;

30 получают доступ к множеству компонентов локализации, которые предоставляют локализованные данные, которые локализованы для одного или более отдельных рынков сбыта;

35 получают доступ к массиву онтологии, хранящему информацию онтологии для устранения неоднозначности данных, ассоциированных с конкретными ресурсами, предназначенными для локализации на основании запроса на локализацию;

выбирают локализованные данные от одного или более из множества компонентов локализации на основании предварительно заданных критериев и данных с устраненной неоднозначностью и

40 обеспечивают обратную связь с разработчиком приложения, причем обратная связь включает в себя указание коэффициента повторного использования множества ресурсов, при этом коэффициент повторного использования указывает информацию в множестве компонентов локализации, которая уже была локализована.

45 8. Способ по п.7, в котором компоненты локализации предоставляют оценки достоверности, ассоциированные с локализованными данными, с учетом входных данных, и в котором этап, на котором выбирают данные, выполняется компонентом обнаружения совпадений в переводах и содержит этап, на котором:

50 выбирают локализованные данные на основании оценок достоверности, при этом выбор включает в себя определение относительной величины оценки достоверности связанной с первым элементом локализованных данных в сравнении с оценкой достоверности, связанной со вторым элементом локализованных данных.

9. Способ по п.7, в котором компоненты локализации содержат множество

массивов данных, хранящих данные, локализованные для одного или более отдельных рынков сбыта.

10. Способ по п.9, в котором компоненты локализации содержат, по меньшей мере, одну систему автоматического перевода, которая обеспечивает определенный вид локализованных данных для выбора упомянутым компонентом обнаружения совпадений в переводах, при этом упомянутый определенный вид представляет собой автоматически генерируемый перевод данных с первого языка на отличающийся второй язык.

11. Способ по п.7, в котором обратная связь включает в себя указание множества различных классов ресурсов, которые содержатся во множестве компонентов локализации и включают в себя уже локализованные данные.

12. Реализованная на компьютере система локализации информационного наполнения и/или программного обеспечения, содержащая:

процессор, ассоциированный с упомянутым компьютером;

приложение, имеющее один или более ресурсов, предназначенных для локализации;

и

платформу локализации информационного наполнения и/или программного обеспечения, реализованную с использованием упомянутого процессора, причем упомянутая платформа локализации содержит:

устройство управления ресурсами, выполненное с возможностью приема входных данных, соответствующих одному или более ресурсам, предназначенных для локализации;

множество компонентов локализации, предоставляющих локализованные данные, которые локализованы для одного или более отдельных рынков сбыта;

массив онтологии, хранящий информацию онтологии; и

компонент обнаружения совпадений в переводах, выполненный с возможностью приема от устройства управления ресурсами запроса на локализацию и входных данных, соответствующих одному или более ресурсам, предназначенных для локализации, при этом упомянутый компонент обнаружения совпадений в переводах выполнен с возможностью выполнения доступа к упомянутому множеству компонентов локализации и массиву онтологии для устранения неоднозначности входных данных, ассоциированных с приложением, предназначенным для локализации, основываясь на запросе на локализацию, и причем упомянутый компонент обнаружения совпадений в переводах выполнен с возможностью выбора и вывода локализованных данных для упомянутых одного или более ресурсов из одного или более из упомянутого множества компонентов локализации,

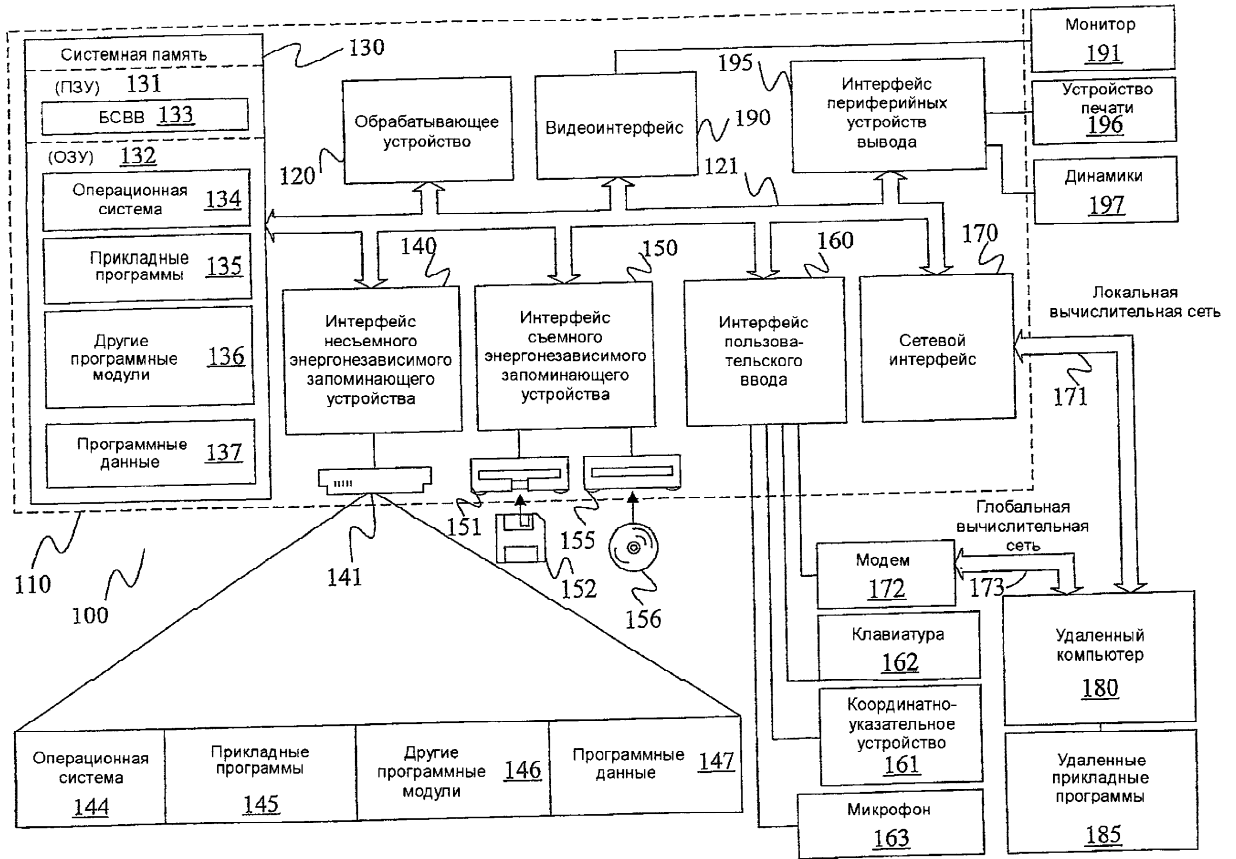
при этом компоненты локализации предоставляют оценки достоверности, ассоциированные с локализованными данными, с учетом информации входных данных, причем компонент обнаружения совпадений в переводах выполнен с возможностью выбора локализованных данных на основании оценок достоверности, при этом выбор включает в себя определение относительной величины оценки достоверности связанной с первым элементом локализованных данных в сравнении с оценкой достоверности, связанной со вторым элементом локализованных данных, причем

первый и второй элементы локализованных данных получают из различных компонентов упомянутого множества компонентов локализации, и

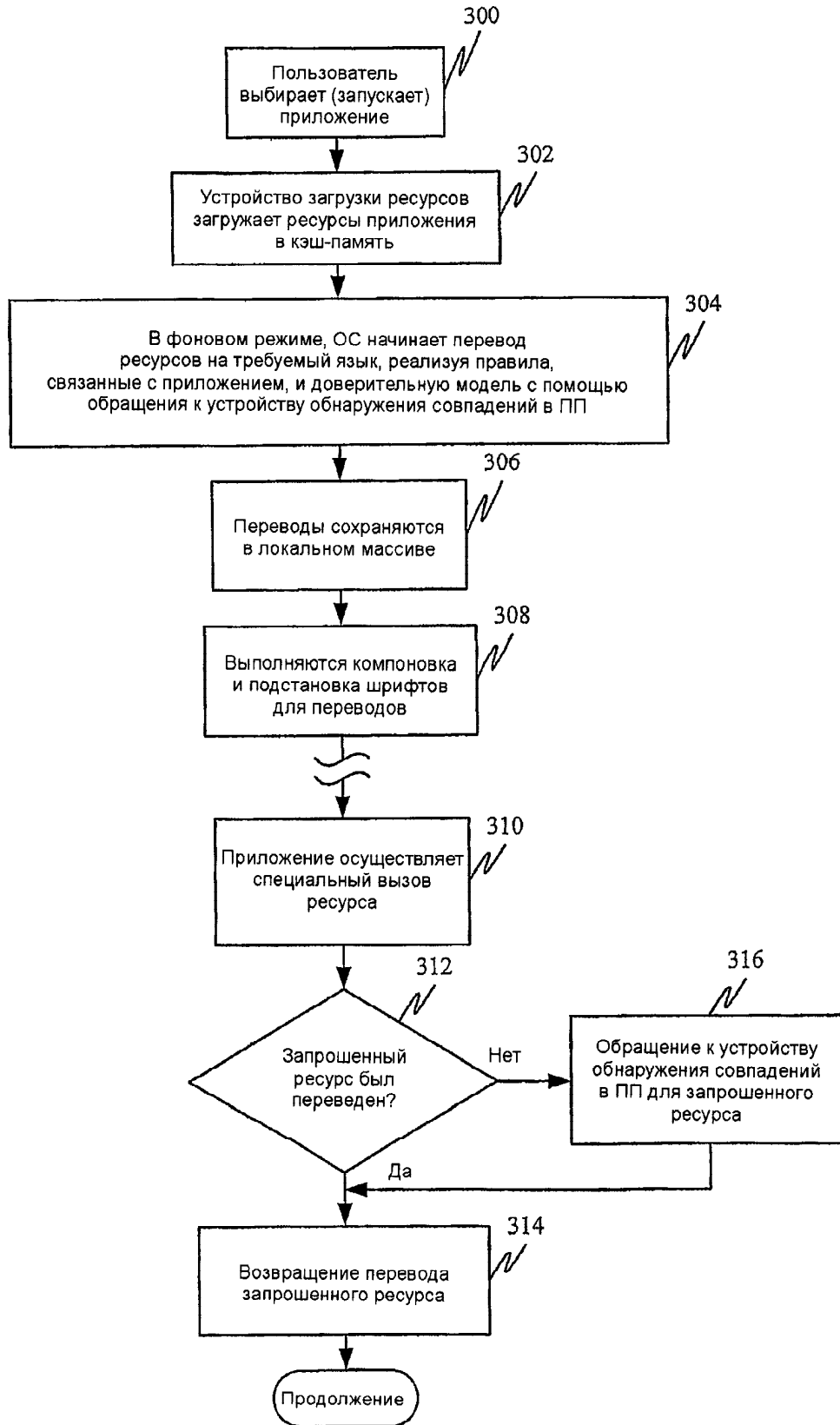
первая и вторая оценки достоверности содержат различные меры достоверности, которые коррелируют друг с другом.

13. Система по п.12, дополнительно содержащая второе приложение, отличающееся от упомянутого приложения, причем второе приложение содержит один или более ресурсов, предназначенных для локализации, причем компонент обнаружения совпадений в переводах, выполнен с возможностью приема от устройства управления ресурсами вторых входных данных, соответствующих одному или более ресурсам, ассоциированным со вторым приложением, доступа к упомянутому множеству компонентов локализации и массиву онтологии для устранения неоднозначности вторых входных данных, и причем упомянутый компонент обнаружения совпадений в переводах выполнен с возможностью выбора и вывода вторых локализованных данных для упомянутых одного или более ресурсов второго приложения из одного или более из упомянутого множества компонентов локализации.

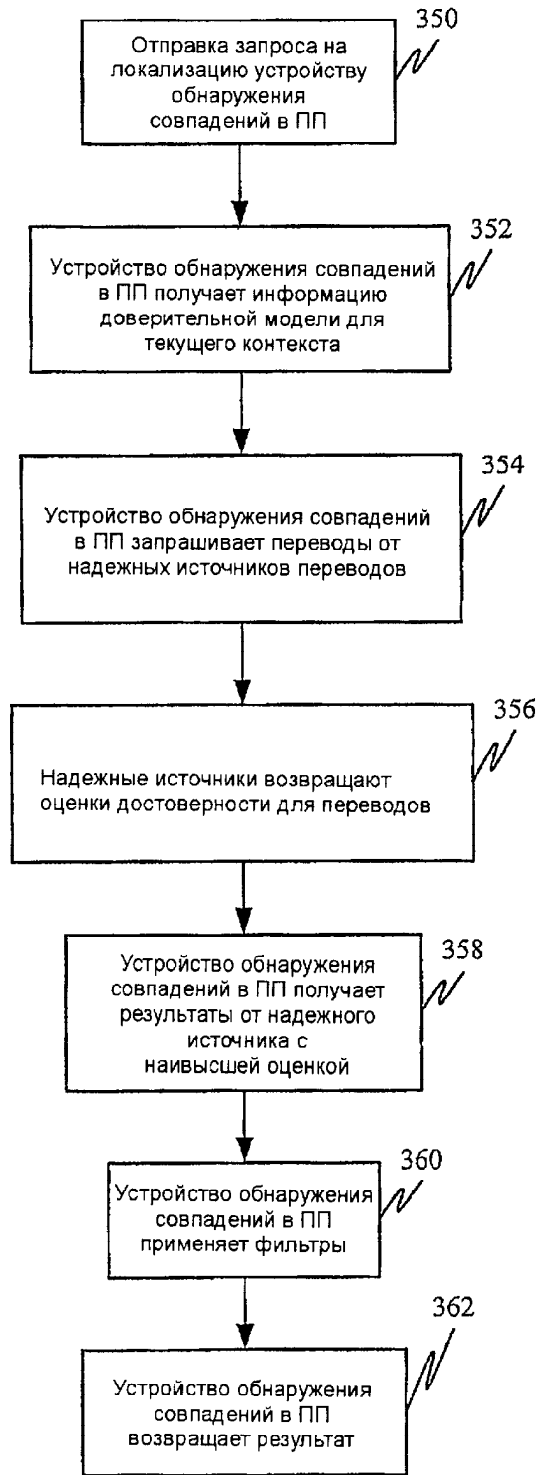
14. Система по п.13, в которой уже локализованные данные упомянутых одного или более ресурсов упомянутого приложения используются компонентом обнаружения совпадений в переводах во время выполнения локализации упомянутых одного или более ресурсов упомянутого второго приложения.



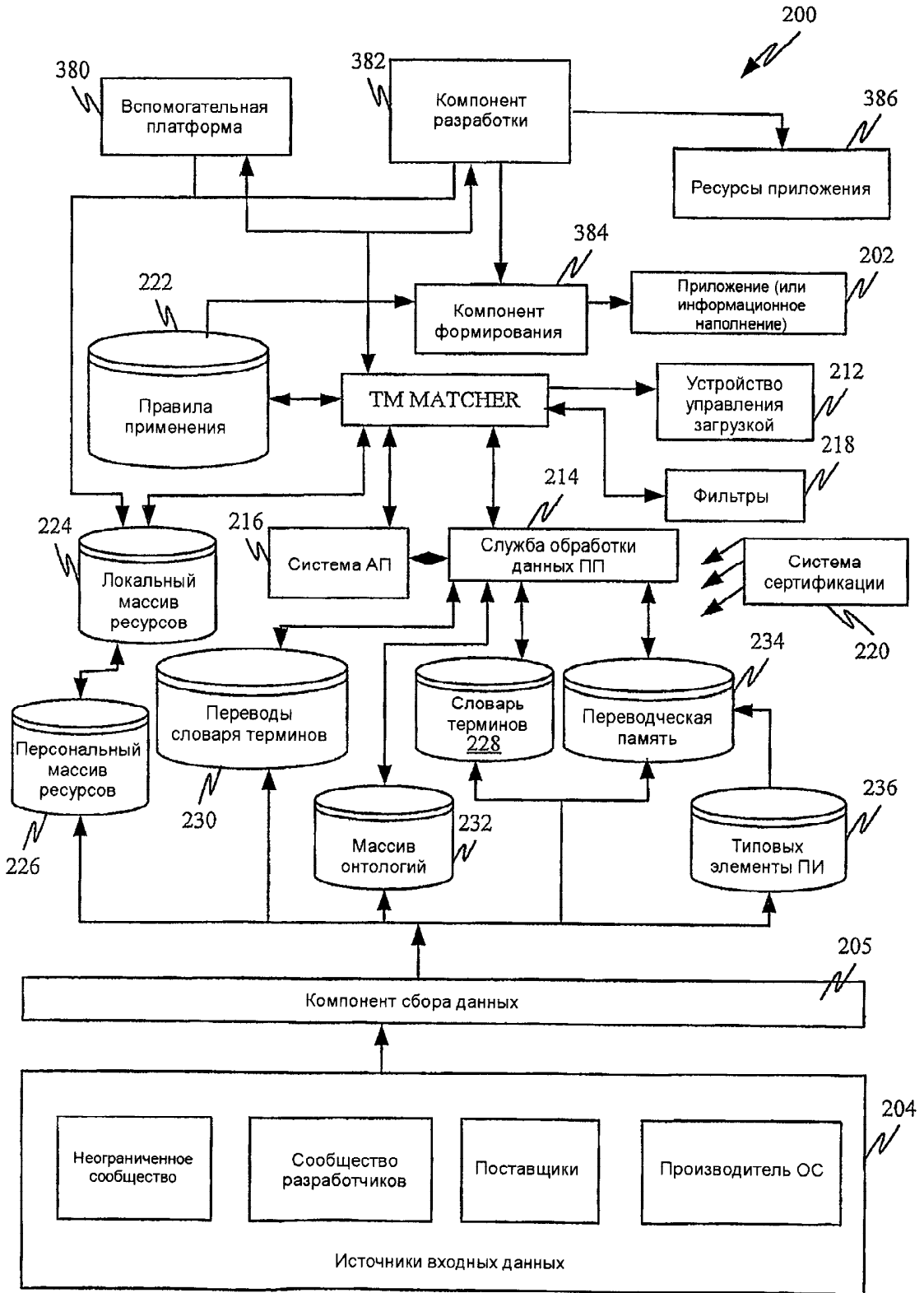
ФИГ. 1



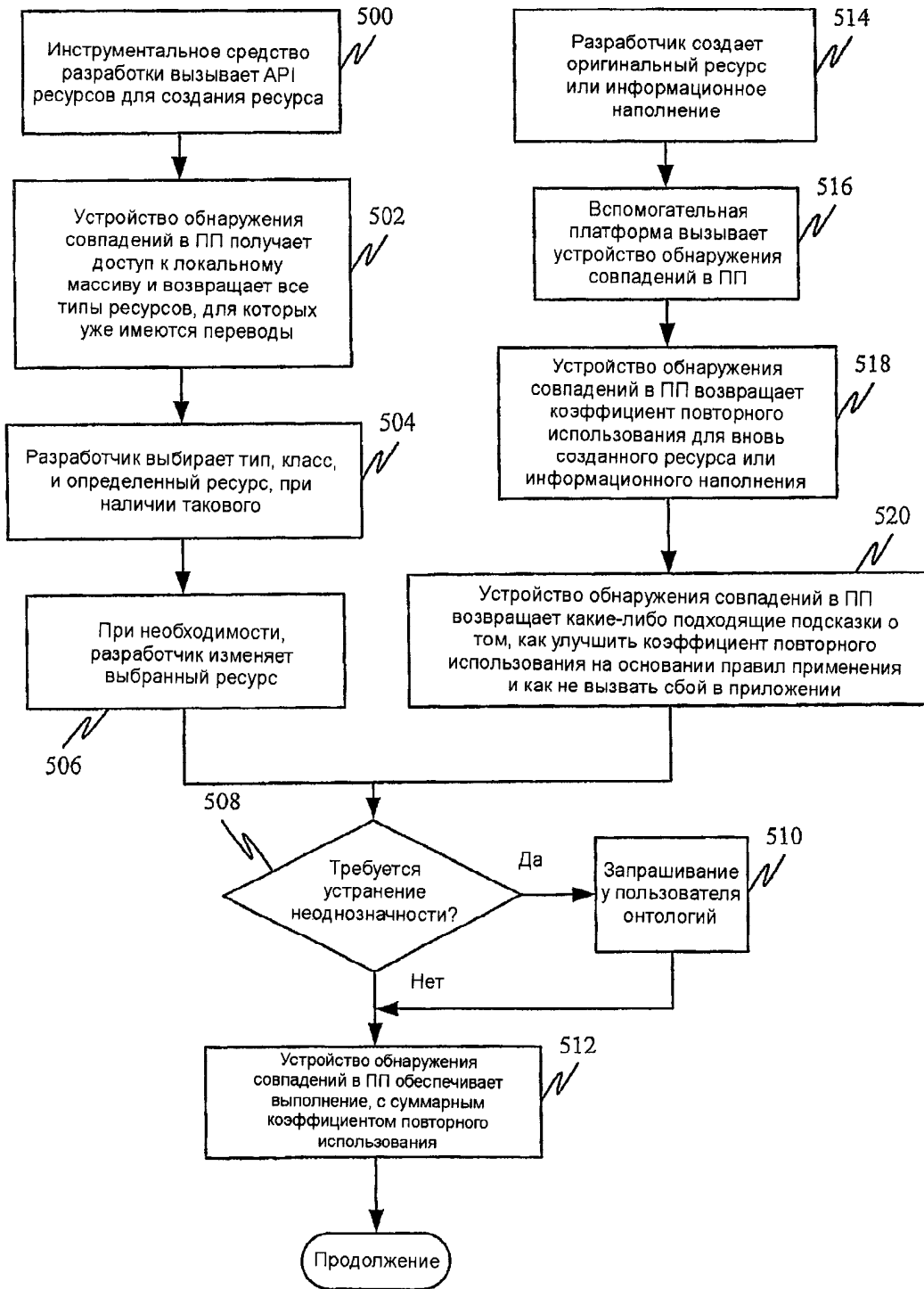
ФИГ. 3



ФИГ. 4



Фиг. 5



Фиг. 6