РОССИЙСКАЯ ФЕДЕРАЦИЯ



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 697 625**(13) **C2**

(51) МПК
G06F 9/44 (2006.01)
G06F 9/54 (2006.01)
G06F 15/16 (2006.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
G06F 9/44 *(2019.05);* ***G06F 9/54*** *(2019.05);* ***G06F 15/16*** *(2019.05);* ***G06F 16/955*** *(2019.05);* ***G06Q 10/10*** *(2019.05)*

(21)(22) Заявка: **2016150840, 29.06.2015**

(24) Дата начала отсчета срока действия патента:
29.06.2015

Дата регистрации:
15.08.2019

Приоритет(ы):
(30) Конвенционный приоритет:
30.06.2014 US 14/320,111

(43) Дата публикации заявки: **25.06.2018** Бюл. № **18**

(45) Опубликовано: **15.08.2019** Бюл. № **23**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **23.12.2016**

(86) Заявка PCT:
US 2015/038220 (29.06.2015)

(87) Публикация заявки PCT:
WO 2016/003849 (07.01.2016)

Адрес для переписки:
129090, Москва, ул. Б.Спасская, 25, строение 3, ООО "Юридическая фирма Городисский и Партнеры"

(72) Автор(ы):
МИТАЛ Виджай (US),
РУБИН Дэррил Э. (US)

(73) Патентообладатель(и):
МАЙКРОСОФТ ТЕКНОЛОДЖИ ЛАЙСЕНСИНГ, ЭлЭлСи (US)

(56) Список документов, цитированных в отчете о поиске: **US 2005/0177589 A1, 11.08.2005. US 4949254 A, 14.08.1990. US 7260599 B2, 21.08.2007. RU 2386999 C2, 20.04.2010.**

(54) ПРИМЕНЕНИЕ СОСТАВНЫХ ЦЕПОЧЕК ПРЕОБРАЗОВАНИЙ ДЛЯ МНОЖЕСТВА УСТРОЙСТВ

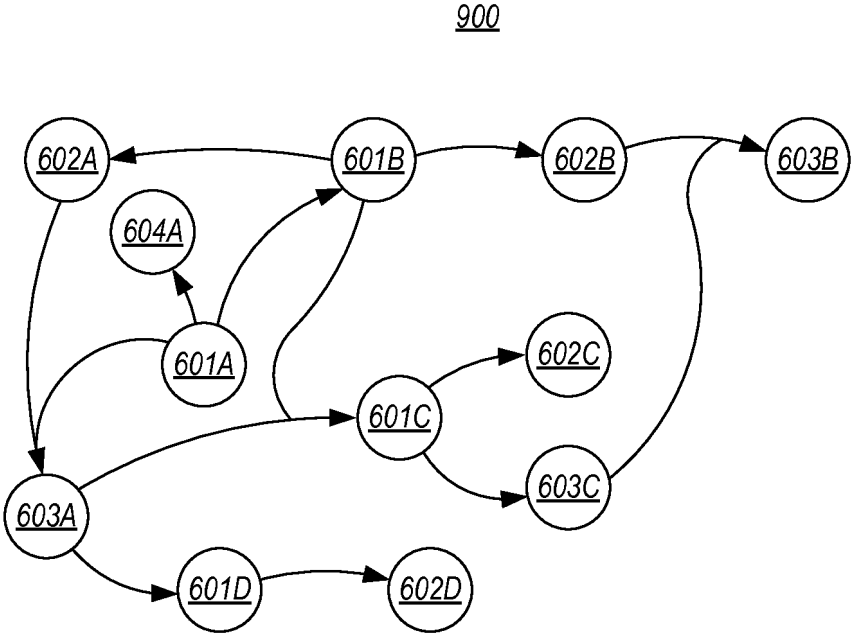
(57) Реферат:

Изобретение относится к области вычислительной техники. Техническим результатом является обеспечение составного приложения на множестве устройств. Раскрыт реализуемый компьютером способ для обеспечения составного приложения на множестве устройств, причем способ содержит этап, на котором идентифицируют зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований посредством, по меньшей мере, получения определения первого класса цепочек преобразований и определения второго класса цепочек преобразований, которые взаимно задают зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований; этап, на котором идентифицируют первое устройство, которое ассоциировано с первым экземпляром первого класса цепочек преобразований; этап, на котором

RU 2697625 C2

RU 2697625 C2

идентифицируют второе устройство, которое ассоциировано со вторым экземпляром второго класса цепочек преобразований; этап, на котором определяют то, что первый экземпляр и второй экземпляр должны объединяться; в ответ на определение того, что первый экземпляр и второй экземпляр должны объединяться, объединение первого экземпляра и второго экземпляра; и этап, на котором обеспечивают протекание данных между первым и вторым экземплярами в соответствии с идентифицированными зависимостями между первым классом преобразований и вторым классом преобразований и в соответствии с состоянием зависимостей между первым и вторым экземплярами. 5 н. и 24 з.п. ф-лы, 37 ил.



ФИГ. 9

RU 2697625 C2

RU 2697625 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 697 625**(13) **C2**

(51) Int. Cl.
G06F 9/44 (2006.01)
G06F 9/54 (2006.01)
G06F 15/16 (2006.01)

(12) ABSTRACT OF INVENTION

(52) CPC
G06F 9/44 *(2019.05);* ***G06F 9/54*** *(2019.05);* ***G06F 15/16*** *(2019.05);* ***G06F 16/955*** *(2019.05);* ***G06Q 10/10*** *(2019.05)*

(21)(22) Application: **2016150840, 29.06.2015**

(24) Effective date for property rights:
29.06.2015

Registration date:
15.08.2019

Priority:
(30) Convention priority:
30.06.2014 US 14/320,111

(43) Application published: **25.06.2018** Bull. № **18**

(45) Date of publication: **15.08.2019** Bull. № **23**

(85) Commencement of national phase: **23.12.2016**

(86) PCT application:
US 2015/038220 (29.06.2015)

(87) PCT publication:
WO 2016/003849 (07.01.2016)

Mail address:
129090, Moskva, ul. B.Spasskaya, 25, stroenie 3, OOO "Yuridicheskaya firma Gorodisskij i Partnery"

(72) Inventor(s):
MITAL Vidzhaj (US),
RUBIN Derril E. (US)

(73) Proprietor(s):
MAJKROSOFT TEKNOLODZHI LAJSENSING, ElElSi (US)

RU 2697625 C2

RU 2697625 C2

(54) USE OF COMPOSITE TRANSFORMATION CHAINS FOR MULTIPLE DEVICES

(57) Abstract:

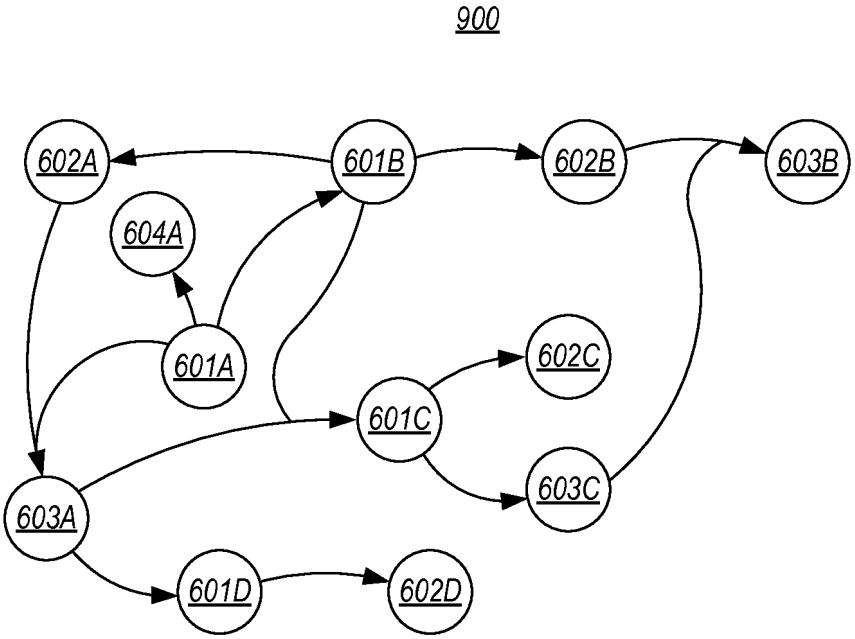
FIELD: calculation.

SUBSTANCE: invention relates to computer engineering. Disclosed is a computer-implemented method for providing a composite application on a plurality of devices, wherein the method comprises a step of identifying relationships between a first class of conversions and a second class of conversions by, at least, obtaining determination of the first class of transformation chains and determination of the second class of transformation chains, which mutually define relationships between the first class of transformation chains and the second class of conversions chains; a step of identifying a first device which is associated with a first instance of a first class of conversions; a step of identifying a second device which is associated with a second instance of a second class of conversions; a step of determining that the first instance and the second instance are to be merged; in response to determining that the first instance and the second instance are to be merged, combining the first instance and the second instance; and a step of enabling data flow between the first and second instances in accordance with the identified relationships between the first transform class and the second transform class

and in accordance with the state of dependencies between the first and second instances.

EFFECT: technical result is providing composite application on multiple devices.

29 cl, 37 dwg



ФИГ. 9

RU 2697625 C2

RU 2697625 C2

Уровень техники

[0001] Вычислительные технологии коренным образом изменили способ, которым мы работаем, играем и общаемся. По мере совершенствования вычислительных технологий, также совершенствуются множество устройств, которые осуществляют такие вычислительные системы или отображают контент из вычислительной системы. Например, вычислительная система может принимать форму серверной стойки, настольного компьютера, переносного компьютера, планшетного компьютера, смартфона, игровой приставки, часов, холодильника, интеллектуального дома и т.п.

[0002] Вместе с многоплановостью вычислительных систем, аналогично диверсифицированы типы устройств, которые могут использоваться для того, чтобы осуществлять рендеринг вычислительной выходной и входной информации в компьютере. Например, устройства вывода могут включать в себя дисплеи, такие как проекторы, телевизионные мониторы, трехмерные дисплеи, переносные компьютеры, планшетные компьютеры, телефоны и т.п. Устройства вывода могут включать в себя вывод для звука, такой как динамики. Устройства вывода также могут включать в себя актуаторы, световые индикаторы, клапаны и т.п. Устройства ввода могут включать в себя клавиатуры, указательные устройства (к примеру, мышь), сенсорные экраны, микрофоны, видеокамеры, цифровые фотокамеры, детекторы трехмерной позиции, мониторы по стандарту глобальной системы позиционирования, светочувствительные датчики, акселерометры, термометры, компасы и т.п.

[0003] Вычислительные системы и ассоциированные устройства ввода и вывода стали довольно разнообразными и зачастую мобильными. Зачастую, в любом данном местоположении, может присутствовать большое число и широкий спектр устройств. Например, в среднестатистическом конференц-зале, который полностью заполняется, могут быть предусмотрены диапроекторы, телевизионные экраны, переносные компьютеры, планшетные компьютеры, смартфоны, микрофоны, камеры, подсветка и т.п. Традиционная парадигма заключается в том, что каждое устройство выполняет собственное приложение или отображает контент из одного приложения. Когда приложения фактически взаимодействуют, они зачастую взаимодействуют в качестве отдельных приложений, взаимодействующих через интерфейс прикладного программирования.

[0004] Предмет изобретения, заявленный в данном документе, не ограничен вариантами осуществления, которые разрешают недостатки или которые работают только в таких окружениях, как описанные выше. Наоборот, это описание уровня техники предоставляется только для того, чтобы проиллюстрировать одну примерную область техники, в которой могут осуществляться на практике некоторые варианты осуществления, описанные в данном документе.

Сущность изобретения

[0005] По меньшей мере, некоторые варианты осуществления, описанные в данном документе, относятся к составному приложению, которое работает на множестве устройств с использованием составляющих цепочек преобразований, ассоциированных с несколькими устройствами. Зависимости идентифицируются между различными классами цепочек преобразований. Устройства также идентифицируются вместе со своим ассоциированным экземпляром класса цепочек преобразований. После определения того, что экземпляры цепочек преобразований, ассоциированные с различными устройствами, должны объединяться, данным разрешается протекать между различными устройствами в соответствии с идентифицированными зависимостями между их соответствующими классами ассоциированных экземпляров цепочек

преобразований и в соответствии с состоянием зависимостей между экземплярами. Таким образом, формируется составная цепочка преобразований. В некоторых вариантах осуществления, по мере того, как устройства выходят из окружения, ассоциированные экземпляры цепочек преобразований могут удаляться из составной цепочки преобразований. Таким образом, вместо устройств, имеющих взаимодействующие приложения, устройства становятся взаимно контактирующими с идентичным составным приложением. По мере того, как новые устройства входят в окружение, либо по мере того, как устройства выходят из окружения, само составное приложение изменяется.

[0006] Эта сущность не имеет намерение идентифицировать ключевые или важнейшие признаки заявленного предмета изобретения, а также не имеет намерение использоваться в качестве помощи при определении объема заявленного предмета изобретения.

Краткое описание чертежей

[0007] Чтобы описывать способ, которым могут получаться вышеизложенные и другие преимущества и признаки, приводится более подробное описание различных вариантов осуществления в отношении прилагаемых чертежей. При условии того, что эти чертежи изображают только примерные варианты осуществления и, следовательно, не должны рассматриваться как ограничивающие объем изобретения, изобретение описывается и поясняется с помощью дополнительной специфичности и подробностей посредством использования прилагаемых чертежей, на которых:

[0008] Фиг. 1 абстрактно иллюстрирует вычислительную систему, в которой могут использоваться некоторые варианты осуществления, описанные в данном документе;

[0009] Фиг. 2 абстрактно иллюстрирует простую цепочку преобразований, в которой предусмотрена всего лишь одна связь, связывающая один источник данных и одну цель данных, и в которой преобразование, представленное посредством связи, автоматически выполняется с использованием значения в источнике данных в качестве ввода, чтобы формировать значение в цели данных;

[0010] Фиг. 3 абстрактно иллюстрирует другую простую примерную цепочку преобразований, в которой преобразование выполняется с использованием входных значений из трех источников данных, чтобы формировать выходные значения в двух целях данных;

[0011] Фиг. 4 иллюстрирует цепочку преобразований в форме комбинации цепочки преобразований по фиг. 2 и цепочки преобразований по фиг. 3;

[0012] Фиг. 5 иллюстрирует примерное окружение, в котором могут работать принципы, описанные в данном документе, и которое включает в себя несколько устройств, ассоциированных с составляющими цепочками преобразований составного приложения, и которое также включает в себя устройства ввода и устройства вывода;

[0013] Фиг. 6A-6D иллюстрируют примерные цепочки преобразований, которые могут быть ассоциированы с соответствующими устройствами на фиг. 5 (стрелки, через которые не протекают данные при отсутствии объединения с другой цепочкой преобразований, проиллюстрированы с помощью "X", а элементы зависимостей, которые не представляют собой узлы в самой цепочке преобразований, проиллюстрированы с границами с пунктирной линией);

[0014] Фиг. 7A иллюстрирует дополненную цепочку преобразований, представляющую объединение цепочек преобразований по фиг. 6A и 6B;

[0015] Фиг. 7B иллюстрирует дополненную цепочку преобразований, представляющую объединение цепочек преобразований по фиг. 6A и 6C;

[0016] Фиг. 7C иллюстрирует дополненную цепочку преобразований,

представляющую объединение цепочек преобразований по фиг. 6B и 6C;

[0017] Фиг. 7D иллюстрирует дополненную цепочку преобразований, представляющую объединение цепочек преобразований по фиг. 6A и 6D;

[0018] Фиг. 8A иллюстрирует дополненную цепочку преобразований, представляющую объединение цепочек преобразований по фиг. 6A, 6B и 6C;

[0019] Фиг. 8B иллюстрирует дополненную цепочку преобразований, представляющую объединение цепочек преобразований по фиг. 6A, 6B и 6D;

[0020] Фиг. 8C иллюстрирует дополненную цепочку преобразований, представляющую объединение цепочек преобразований по фиг. 6A, 6C и 6D;

[0021] Фиг. 9 иллюстрирует дополненную цепочку преобразований, представляющую объединение цепочек преобразований по фиг. 6A, 6B, 6C и 6D;

[0022] Фиг. 10 иллюстрирует блок-схему последовательности операций способа для подготовки к обеспечению составного приложения;

[0023] Фиг. 11 абстрактно иллюстрирует систему, которая может использоваться для того, чтобы осуществлять способ по фиг. 10, 12, 13 и 14, и которая включает в себя библиотеку определений классов цепочек преобразований и регистраций устройств, а также службу уровня представления;

[0024] Фиг. 12 иллюстрирует блок-схему последовательности операций способа для объединения двух экземпляров классов цепочек преобразований;

[0025] Фиг. 13 иллюстрирует блок-схему последовательности операций способа для отсоединения цепочки преобразований от большей цепочки преобразований;

[0026] Фиг. 14 иллюстрирует блок-схему последовательности операций способа для рендеринга изменений одного или более узлов представления цепочки преобразований на надлежащем устройстве;

[0027] Фиг. 15A-15J иллюстрируют различные пользовательские интерфейсы, которые могут встречаться в сценарии, упоминаемом в данном документе в качестве "сценария для устройств", в котором пользователь должен заказывать определенное число устройств; и

[0028] Фиг. 15K-15O иллюстрируют различные пользовательские интерфейсы, которые могут встречаться, чтобы изменять составное приложение, используемое в сценарии для устройств по фиг. 15A-15J.

Подробное описание изобретения

[0029] По меньшей мере, некоторые варианты осуществления, описанные в данном документе, относятся к составному приложению, которое работает на множестве устройств с использованием составляющих цепочек преобразований, ассоциированных с несколькими устройствами. Зависимости идентифицируются между различными классами цепочек преобразований. Устройства также идентифицируются вместе со своим ассоциированным экземпляром класса цепочек преобразований. После определения того, что экземпляры цепочек преобразований, ассоциированные с различными устройствами, должны объединяться, данным разрешается протекать между различными устройствами в соответствии с идентифицированными зависимостями между их соответствующими классами ассоциированных экземпляров цепочек преобразований и в соответствии с состоянием зависимостей между экземплярами. Таким образом, формируется составная цепочка преобразований. В некоторых вариантах осуществления, по мере того, как устройства выходят из окружения, ассоциированные экземпляры цепочек преобразований могут удаляться из составной цепочки преобразований. Таким образом, вместо устройств, имеющих взаимодействующие приложения, устройства становятся взаимно контактирующими

с идентичным составным приложением. По мере того, как новые устройства входят в окружение, либо по мере того, как устройства выходят из окружения, само составное приложение изменяется.

[0030] Некоторое вводное пояснение вычислительной системы приводится относительно фиг. 1. После этого описывается технология составного приложения для устройства относительно последующих чертежей.

[0031] Вычислительные системы сегодня все в большей степени принимают множество форм. Вычислительные системы могут представлять собой, например, карманные устройства, бытовые приборы, переносные компьютеры, настольные компьютеры, мейнфреймы, распределенные вычислительные системы или даже устройства, которые традиционно не считаются вычислительной системой. В этом описании и в формуле изобретения, термин "вычислительная система" задается в широком смысле как включающий в себя любое устройство или систему (либо их комбинацию), которая включает в себя, по меньшей мере, один физический и материальный процессор и физическое и материальное запоминающее устройство, допускающее наличие исполняемых компьютером инструкций, которые могут выполняться посредством процессора. Запоминающее устройство может принимать любую форму и может зависеть от характера и формы вычислительной системы. Вычислительная система может быть распределена по сетевому окружению и может иметь несколько составляющих вычислительных систем.

[0032] Как проиллюстрировано на фиг. 1, в своей самой базовой конфигурации вычислительная система 100 типично включает в себя, по меньшей мере, один аппаратный модуль 102 обработки и запоминающее устройство 104. Запоминающее устройство 104 может представлять собой физическое системное запоминающее устройство, которое может быть энергозависимым, энергонезависимым либо комбинацией вышеозначенного. Термин "запоминающее устройство" также может быть использован в данном документе для того, чтобы ссылаться на энергонезависимое устройство хранения большой емкости, такое как физические носители данных. Если вычислительная система является распределенной, возможности обработки, запоминающего устройства и/или хранения также могут быть распределенными. При использовании в данном документе, термин "выполняемый модуль" или "выполняемый компонент" может означать программные объекты, маршрутизации или способы, которые могут выполняться в вычислительной системе. Различные компоненты, модули, механизмы и службы, описанные в данном документе, могут быть реализованы как объекты или процессы, которые выполняются в вычислительной системе (к примеру, как отдельные подпроцессы).

[0033] В нижеследующем описании, варианты осуществления поясняются со ссылкой на этапы, которые выполняются посредством одной или более вычислительных систем. Если эти этапы реализованы в программном обеспечении, один или более процессоров ассоциированной вычислительной системы, которая выполняет этап, управляют работой вычислительной системы в ответ на выполнение считываемых компьютером инструкций. Например, такие исполняемые компьютером инструкции могут быть осуществлены на одном или более считываемых компьютером носителей, которые формируют компьютерный программный продукт. Пример такой операции заключает в себе обработку данных. Исполняемые компьютером инструкции (и обработанные данные) могут сохраняться в запоминающем устройстве 104 вычислительной системы 100. Вычислительная система 101 также может содержать каналы 108 связи, которые обеспечивают возможность вычислительной системе 100 обмениваться данными с

другими процессорами сообщений, например, по сети 110.

[0034] Вычислительная система 100 также может потенциально включать в себя компоненты рендеринга вывода, такие как дисплеи, динамики, световые индикаторы, актуаторы и т.п. Вычислительная система 100 также может включать в себя компоненты ввода, такие как клавиатура, указательное устройство (к примеру, мышь или сенсорная панель), устройства распознавания речи и возможно также физические датчики (например, термометры, глобальные системы позиционирования, световые детекторы, компасы, акселерометры и т.д.).

[0035] Варианты осуществления, описанные в данном документе, могут содержать или использовать компьютер специального назначения или общего назначения, включающий в себя компьютерные аппаратные средства, такие как, например, один или более процессоров и системное запоминающее устройство, как подробнее поясняется ниже. Варианты осуществления, описанные в данном документе, также включают в себя физические и другие считываемые компьютером носители для переноса или сохранения исполняемых компьютером инструкций и/или структур данных. Эти считываемые компьютером носители могу представлять собой любые доступные носители, к которым можно осуществлять доступ посредством компьютерной системы общего назначения или специального назначения. Считываемые компьютером носители, которые сохраняют исполняемые компьютером инструкции, представляют собой физические носители данных. Считываемые компьютером носители, которые переносят исполняемые компьютером инструкции, представляют собой среды передачи. Таким образом, в качестве примера, а не ограничения, варианты осуществления изобретения могут содержать, по меньшей мере, два совершенно различных типа считываемых компьютером носителей: компьютерные носители данных и среды передачи.

[0036] Компьютерные носители данных включают в себя RAM, ROM, EEPROM, CD-ROM или другое устройство хранения данных на оптических дисках, устройство хранения данных на магнитных дисках либо другие магнитные устройства хранения данных или любой другой носитель данных, который может использоваться для того, чтобы сохранять требуемое средство программного кода в форме исполняемых компьютером инструкций или структур данных и к которому может осуществляться доступ посредством компьютера общего назначения или специального назначения.

[0037] "Сеть" задается как одна или более линий связи для передачи данных, которые обеспечивают транспортировку электронных данных между компьютерными системами и/или модулями и/или другими электронными устройствами. Когда информация передается или предоставляется по сети или другим соединениям связи (проводным, беспроводным либо комбинации проводных и беспроводных) в компьютер, компьютер надлежащим образом рассматривает соединение в качестве среды передачи. Среды передачи могут включать в себя сеть и/или линии связи для передачи данных, которые могут использоваться для того, чтобы переносить требуемое средство программного кода в форме исполняемых компьютером инструкций или структур данных, и к которым может осуществляться доступ посредством компьютера общего назначения или специального назначения. Комбинации вышеперечисленного также следует включать в число считываемых компьютером носителей.

[0038] Дополнительно, после достижения различных компонентов компьютерной системы, средство программного кода в форме исполняемых компьютером инструкций или структур данных может передаваться автоматически из сред передачи в компьютерные носители данных (или наоборот). Например, исполняемые компьютером инструкции или структуры данных, принятые по сети или линии связи для передачи

данных, могут быть буферизованы в RAM в пределах сетевого интерфейсного модуля (например, NIC) и затем в конечном счете переданы в RAM компьютерной системы и/или в менее энергозависимые компьютерные носители данных в компьютерной системе. Таким образом, следует понимать, что компьютерные носители данных могут быть включены в компоненты компьютерной системы, которые также (или даже главным образом) используют среды передачи.

[0039] Считываемые компьютером инструкции содержат, например, инструкции и данные, которые при выполнении в процессоре инструктируют компьютеру общего назначения, компьютеру специального назначения или устройству обработки специального назначения выполнять определенную функцию или группу функций. Исполняемые компьютером инструкции, например, могут представлять собой двоичные файлы или даже инструкции, которые подвергаются некоторой трансляции (к примеру, компиляции) перед прямым выполнением посредством процессоров, такие как инструкции в промежуточном формате, к примеру, язык ассемблера или даже исходный код. Хотя предмет изобретения описан на языке, характерном для структурных признаков и/или технологических этапов, следует понимать, что предмет изобретения, заданный в прилагаемой формуле изобретения, не обязательно ограничен описанными признаками или этапами, описанными выше. Наоборот, описанные признаки и этапы раскрываются как примерные формы реализации формулы изобретения.

[0040] Специалисты в данной области техники должны принимать во внимание, что изобретение может осуществляться на практике в сетевых вычислительных окружениях с множеством типов конфигураций компьютерных систем, включающих в себя персональные компьютеры, настольные компьютеры, переносные компьютеры, процессоры сообщений, карманные устройства, многопроцессорные системы, микропроцессорные или программируемые бытовые электронные приборы, сетевые PC, миникомпьютеры, мэйнфреймы, мобильные телефоны, PDA, устройства поискового вызова, маршрутизаторы, коммутаторы и т.п. Изобретение также может осуществляться на практике в распределенных системных окружениях, в которых локальные и удаленные компьютерные системы, которые соединены (посредством проводных линий связи для передачи данных, беспроводных линий связи для передачи данных или посредством комбинации проводных и беспроводных линий связи для передачи данных) через сеть, выполняют задачи. В распределенном системном окружении, программные модули могут размещаться как в локальных, так и в удаленных запоминающих устройствах.

[0041] Принципы, описанные в данном документе, работают с использованием цепочки преобразований. Цепочка преобразований представляет собой взаимно соединенный набор узлов, каждый из которых может представлять источники данных или цели данных. Предусмотрены связи между узлами, причем каждая связь представляет преобразование. Для любой данной связи, ассоциированное преобразование принимает копии значений одного или более источников данных, расположенных на входном конце связи, и формирует результирующие значения, предоставляемые в одной или более целей данных, расположенных на выходном конце связи. Для любого данного преобразования, когда значение в одном или более источников данных на входном конце изменяется, преобразование автоматически оценивается заново, что потенциально приводит к изменениям значения(й) одной или более целей данных на выходном конце преобразования.

[0042] В одном варианте осуществления, независимо от того, насколько сложной является цепочка преобразований, преобразования могут состоять из декларативных операторов, выражающих уравнения, правила, ограничения, моделирования или любой

другой тип преобразования, которое может принимать одно или более значений в качестве ввода и предоставлять результирующие одно или более значений в качестве вывода. Пример цепочки преобразований представляет собой программу для обработки электронных таблиц, в которой любая из ячеек может представлять собой источник данных или цель данных. Уравнение (т.е. преобразование) может быть ассоциировано с любой ячейкой, чтобы инструктировать этой ячейке представлять собой цель данных, в которой размещены результаты уравнения.

[0043] Только в качестве примера, фиг. 2 иллюстрирует простую цепочку 200 преобразований, в которой предусмотрена всего лишь одна связь 220. В системе обозначений чертежей, используемой в этом описании, связь проиллюстрирована в качестве стрелки, при этом входной конец представляется как хвост стрелки, а выходной конец представляется как острие стрелки. В случаях, в которых предусмотрено несколько источников данных на входном конце связи, стрелка должна представляться с несколькими хвостами. Копии значений источника(ов) данных в хвосте(ах) стрелки представляют ввод в преобразование. В случаях, в которых предусмотрено несколько целей данных, затрагиваемых посредством результирующего значения(й) преобразования, стрелка должна представляться с несколькими остриями. Значения цели(ей) данных в острие(ях) стрелки представляют вывод из преобразования.

[0044] Например, фиг. 2 иллюстрирует простую цепочку 200 преобразований, которая включает в себя источник 201 данных, цель 202 данных и одну связь 220. Связь 220 представляет преобразование, выполняемое для копии значения 211 в источнике 201 данных, чтобы формировать значение 212 в цели 202 данных. Если значение 211 изменяется, преобразование, представленное посредством связи 220, оценивается заново, что потенциально приводит к изменению значения 212 в цели 202 данных.

[0045] Фиг. 3 иллюстрирует другую простую примерную цепочку 300 преобразований, которая включает в себя три источника 301, 302 и 303 данных; две цели 304 и 305 данных и одну связь 320. Связь 320 представляет преобразование, выполняемое для копий значений в источниках 301, 302 и 303 данных, чтобы формировать значения в целях 304 и 305 данных. Если любое из значений в источниках 301, 302 или 303 данных изменяется, связь 320 типа "преобразование" оценивается заново, что потенциально приводит к изменению значений в любой одной или более из целей 304 и 305 данных.

[0046] Фиг. 4 иллюстрирует другую примерную цепочку 400 преобразований и иллюстрирует такой принцип, что цепочки преобразований могут основываться друг на друге, при этом источник данных в одну связь может представлять собой цель данных в другой связи, чтобы создавать еще более сложные цепочки преобразований. Например, цепочка 400 преобразований включает в себя экземпляр 401 цепочки 200 преобразований и экземпляр 402 цепочки 300 преобразований. В этом случае, цель 202 данных связи 220 также представляет собой источник 301 данных связи 320. Если значение для источника 201 данных изменяется, преобразование, представленное посредством связи 220, оценивается заново, что потенциально приводит к изменению значения в цели 202 данных, которая аналогично представляет собой источник 301 данных для следующей связи 320. Аналогично, изменение значения источника 301 данных должно приводить к оценке заново связи 320 типа "преобразование", что потенциально приводит к изменению значений в любой одной или более из целей 304 и 305 данных. Цели 304 и 305 данных аналогично могут представлять источники данных для еще других связей. Соответственно, в сложных цепочках преобразований, изменение значения может вызывать изменения распространяемых значений через несколько узлов в цепочке преобразований через надлежащую автоматизированную переоценку преобразований

в цепочке преобразований.

[0047] Хотя примерная цепочка 400 преобразований включает в себя всего две связи, цепочки преобразований могут быть довольно сложными и включать счетные узлы и ассоциированные связи, соединяющие эти счетные узлы. Принципы, описанные в данном документе, могут работать независимо от сложности цепочек преобразований.

[0048] Фиг. 5 иллюстрирует окружение 500, в котором предусмотрено шесть устройств 501-506. Эллипсы 507 представляют гибкость в числе устройств, которые присутствуют в окружении 500. Фактически, могут быть устройства, выходящие и входящие в окружение 500 достаточно динамически. Хотя необязательно, каждое из устройств 507 может иметь такую структуру, как описано для вычислительной системы 100 по фиг. 1.

[0049] Окружение 500 необязательно должно представлять собой физическое окружение, в котором все устройства расположены в идентичной близости, хотя это зачастую может иметь место. Окружение 500 вместо этого может рассматриваться в качестве любого окружения, в котором предусмотрен набор устройств, через которые один или более пользователей могут предоставлять ввод, и несколько устройств, через которые вывод может предоставляться нескольким пользователям.

[0050] Некоторые устройства (например, устройства 501-504) помогают в формировании того, что представляет собой больше чем просто взаимодействующие устройства со взаимодействующими приложениями. Вместо этого, устройства 501-504 ассоциированы с компонентом составного приложения. По мере того, как любые из устройств 501-504 выходят из окружения 500, составное приложение становится меньшим, в силу этого приводя к измененной функциональности приложения. С другой стороны, по мере того как устройства (к примеру, устройства 501-504), которые имеют компоненты приложения, входят в окружение 500, составное приложение фактически становится более крупным, в силу этого фактически изменяя функциональность и саму структуру составного приложения. В соответствии с принципами, описанными в данном документе, цепочка преобразований одного устройства может объединяться с цепочкой преобразований других устройств, что приводит к большей цепочке преобразований, которая может эффективнее использовать дополненный набор устройств.

[0051] Устройство 505 представляет устройство вывода, которое может использоваться в окружении 500, но не обязательно добавляет цепочку преобразований в большую цепочку преобразований составного приложения. Например, устройство 505 может представлять собой дисплей с большим экраном. Устройство 506 представляет устройство ввода, которое может использоваться в окружении 500, но не обязательно добавляет цепочку преобразований в большую цепочку преобразований составного приложения. Например, устройство 506 может представлять собой микрофон. Присутствие устройств 501-504 приводит к добавлению частей соответствующих ассоциированных цепочек преобразований в большую цепочку преобразований составного приложения. Тем не менее, устройства 501-504 также могут обладать возможностью ввода и возможностью вывода, которые могут использоваться посредством составного приложения в целом. Окружение 500 необязательно может включать в себя внешнюю систему 510, которая подробнее описывается ниже.

[0052] Фиг. 6A-6D иллюстрируют примерные экземпляры или классы 600A-600D цепочек преобразований. Экземпляры должны иметь структуру, идентичную структуре классов, и в силу этого проиллюстрированные формы могут считаться представляющими классы преобразований, а также экземпляры преобразований. Тем не менее, экземпляры должны иметь конкретное состояние экземпляра, ассоциированное с каждым из одного

или более узлов цепочки преобразований. Соответственно, элементы 600A-600D могут упоминаться в качестве классов цепочек преобразований или экземпляров цепочек преобразований. Термин "цепочка преобразований" используется для того, чтобы, в общем, означать как классы цепочек преобразований, так и их ассоциированные экземпляры цепочек преобразований. В качестве примера, экземпляры 600A-600D цепочек преобразований могут быть ассоциированы с соответствующими устройствами 501-504.

[0053] Примерные цепочки 600A-600D преобразований являются относительно простыми во избежание затруднения понимания более широких принципов, описанных в данном документе, в силу чрезмерно сложного примера. Тем не менее, принципы, описанные в данном документе, применяются независимо от того, насколько сложной является цепочка преобразований, и независимо от числа цепочек преобразований и ассоциированных устройств, которые находятся в окружении и формируют составное приложение.

[0054] В системе обозначений по фиг. 6A-6D, узлы, которые принадлежат классу 600N преобразований (где N варьируется от A до D), представлены с использованием суффикса N. Например, на фиг. 6A, цепочка 600A преобразований включает в себя узлы 601A, 602A, 603A и 604A. Оставшиеся элементы 601B, 601C и 601D не завершаются суффиксом A и в силу этого не представляют собой узлы в цепочке 600A преобразований. Вместо этого, элементы 601B, 601C и 601D представляют зависимости с другими цепочками преобразований.

[0055] На всех фиг. 6A-6D, 7A-7D, 8A-8C и 9, чтобы подчеркнуть те элементы, которые представляют собой элементы зависимостей, а не узлы в самой цепочке преобразований, элементы зависимостей представлены с границами с пунктирной линей. Данные не протекают из узла в элемент зависимостей до тех пор, пока цепочка преобразований не будет объединена с другой цепочкой преобразований, которая включает в себя узел, представленный посредством элемента зависимостей. Тот факт, что данные не могут протекать вдоль конкретного преобразования, представлен на чертежах посредством связи, помеченной с помощью "X".

[0056] Например, элемент 601B в цепочке 600A преобразований представляет зависимость с узлом 601B в цепочке 600B преобразований. Элемент 601B зависимостей ограничен пунктирными линиями, и все связи, идущие в/из этого элемента 601B зависимостей, помечаются с помощью "X", поскольку на этой стадии, цепочка 600A преобразований не объединена с цепочкой 600B преобразований. Элемент 601C в цепочке 600A преобразований представляет зависимость с узлом 601C в цепочке 600C преобразований. Элемент 601D в цепочке 600A преобразований представляет зависимость с узлом 601D в классе 600D цепочек преобразований.

[0057] Отдельно, экземпляр 600A цепочки преобразований может выступать в качестве приложения. Например, значение из источника 601A данных может использоваться для того, чтобы формировать преобразованный результат в качестве значения цели 604A данных. Кроме того, значения из источников 601A и 602A данных могут преобразовываться, чтобы приводить к значению цели 603A данных. Если экземпляр 600A цепочки преобразований является отдельным, преобразования, идущие в/из элементов 601B, 601C и 601D, не оцениваются.

[0058] Цепочка 600B преобразований включает в себя три узла 601B, 602B и 603B. Тем не менее, цепочка 600B преобразований также включает в себя элементы 601A, 602A, 601C и 603C зависимостей, которые ссылаются на узел в другой цепочке преобразований. С другой стороны, экземпляр 600B цепочки преобразований может

работать независимо в качестве отдельного приложения. Например, значение из источника 601B данных может предоставляться посредством преобразования, чтобы формировать результирующее значение для цели 602B данных. Значение из источника 602B данных может предоставляться посредством преобразования, чтобы формировать результирующее значение для цели 603B данных.

[0059] Хотя экземпляры 600A и 600B цепочек преобразований могут работать независимо, фиг. 7A иллюстрирует объединенную цепочку 700A преобразований, которая включает в себя цепочку 600A преобразований, объединенную с цепочкой 600B преобразований. Элементы зависимостей в каждой из цепочек преобразований теперь заменены фактическим узлом, на который ссылаются. Например, элемент 601B зависимостей по фиг. 6A теперь представляет собой узел 601B, и элемент 601A зависимостей по фиг. 6B теперь представляет собой узел 601A. Все узлы, которые имеют суффикс A или B, представляют собой узлы в цепочке 700A преобразований, и только те узлы, которые имеют суффиксы C или D, представляют собой элементы зависимостей. Например, узлы 601A, 602A, 603A, 604A, 601B, 602B и 603B представляют собой узлы в дополненной цепочке 700A преобразований, а функциональность составного приложения становится в определенной степени лучше суммы функциональности отдельных цепочек 600A и 600B преобразований по отдельности.

[0060] Цепочка 600C преобразований включает в себя три узла 601C, 602C и 603C. Тем не менее, цепочка 600C преобразований также включает в себя элементы 603A, 601B и 603B зависимостей, которые ссылаются на узел в другой цепочке преобразований. С другой стороны, экземпляр 600C цепочки преобразований может работать независимо в качестве отдельного приложения. Например, значение из источника 601C данных может предоставляться посредством преобразования, чтобы формировать результирующее значение для цели 602C данных. Аналогично, значение из источника 601C данных также может предоставляться посредством преобразования, чтобы формировать результирующее значение для цели 603C данных.

[0061] Хотя экземпляры 600A и 600C цепочек преобразований могут работать независимо, фиг. 7B иллюстрирует объединенную цепочку 700B преобразований, которая включает в себя цепочку 600A преобразований, объединенную с цепочкой 600C преобразований. Элементы зависимостей в каждой из цепочек преобразований теперь заменены фактическим узлом, на который ссылаются, до такой степени, что элемент зависимостей ссылается на узел в любой из цепочек 600A или 600C преобразований. Теперь все узлы, которые имеют суффикс A или C, представляют собой узлы в цепочке преобразований, и только те узлы, которые имеют суффиксы B или D, представляют собой элементы зависимостей. Например, узлы 601A, 602A, 603A, 604A, 601C, 602C и 603C представляют собой узлы в дополненной цепочке 700B преобразований. Функциональность составного приложения становится лучше суммы функциональностей отдельных экземпляров 600A и 600C цепочек преобразований.

[0062] Фиг. 7C иллюстрирует объединенную цепочку 700C преобразований, которая включает в себя класс 600B цепочек преобразований, объединенный с классом 600C цепочек преобразований. Элементы зависимостей в каждой из цепочек преобразований заменены фактическим узлом, на который ссылаются, до такой степени, что элемент зависимостей ссылается на узел в любой из цепочек 600B или 600C преобразований. Теперь все узлы, которые имеют суффикс B или C, представляют собой узлы в цепочке преобразований, и только те узлы, которые имеют суффиксы A или D, представляют собой элементы зависимостей. Например, узлы 601B, 602B, 603B, 601C, 602C и 603C представляют собой узлы в дополненной цепочке 700C преобразований, и

функциональность составного приложения становится лучше суммы функциональностей отдельных экземпляров 600B и 600C цепочек преобразований.

[0063] Фиг. 8A иллюстрирует объединенную цепочку 800A преобразований, которая включает в себя цепочки 600A, 600B и 600C преобразований, также объединенные. Элементы зависимостей в каждой из цепочек преобразований заменены фактическим узлом, на который ссылаются, до такой степени, что элемент зависимостей ссылается на узел в любой из цепочек 600A, 600B или 600C преобразований. Следует отметить, что все проиллюстрированные узлы фактически представляют собой узлы в цепочке преобразований, за исключением элемента 601D зависимостей. Функциональность составного приложения становится лучше суммы функциональности отдельных цепочек 600A, 600B и 600C преобразований.

[0064] Цепочка 600D преобразований включает в себя два узла 601D и 602D. Тем не менее, цепочка 600D преобразований также включает в себя один элемент 603A зависимостей, ссылающийся на узел в другом классе 600A цепочек преобразований. С другой стороны, экземпляры класса 600D цепочек преобразований могут работать независимо в качестве отдельного приложения. Например, значение из источника 601D данных может предоставляться посредством преобразования, чтобы формировать результирующее значение для цели 602D данных.

[0065] Хотя экземпляры 600A и 600D цепочек преобразований могут работать независимо, фиг. 7D иллюстрирует объединенную цепочку 700D преобразований, которая включает в себя цепочку 600A преобразований, объединенную с цепочкой 600D преобразований. Элементы зависимостей в каждой из цепочек преобразований теперь заменены фактическим узлом, на который ссылаются, до такой степени, что элемент зависимостей ссылается на узел в любой из цепочек 600A или 600D преобразований. Теперь все узлы, которые имеют суффикс A или D, представляют собой узлы в цепочке преобразований, и только те узлы, которые имеют суффиксы B или C, представляют собой элементы зависимостей. Например, узлы 601A, 602A, 603A, 604A, 601D и 602D представляют собой узлы в дополненной цепочке 700D преобразований, и функциональность составного приложения становится в определенной степени лучше суммы функциональности отдельной цепочки 600A и 600D преобразований.

[0066] Следует отметить, что фиг. 7A-7D иллюстрируют все возможные перестановки, заключающие в себе две и только две из цепочек 600A, 600B, 600C и 600D преобразований. Цепочки 600B и 600D преобразований не объединяются непосредственно в комбинации из двух цепочек преобразований, поскольку ни одна цепочка преобразований не имеет элемент зависимостей, ссылающийся на узел в другой цепочке преобразований. Кроме того, преобразования 600C и 600D не объединяются непосредственно в комбинации из двух цепочек преобразований, поскольку ни одна из них не имеет ссылку зависимостей на другую.

[0067] Фиг. 8A иллюстрирует одну из трех возможных комбинаций из трех и только трех цепочек 600A, 600B, 600C и 600D преобразований. В частности, фиг. 8A иллюстрирует дополненную цепочку 800A преобразований, которая комбинирует цепочки 600A, 600B и 600C преобразований. Фиг. 8B иллюстрирует дополненную цепочку 800B преобразований, которая комбинирует цепочки 600A, 600B и 600D преобразований (в которой все узлы являются частью цепочки преобразований, за исключением элементов 601C и 603C зависимостей). Фиг. 8C иллюстрирует дополненную цепочку 800C преобразований, которая комбинирует цепочки 600A, 600C и 600D преобразований (в которой все узлы являются частью цепочки преобразований, за

исключением элементов 601B и 603B зависимостей). Следует отметить, что не проиллюстрирована комбинация цепочек 600B, 600C и 600D преобразований, поскольку цепочка 600D преобразований не включает в себя ссылки зависимостей на цепочку 600B преобразований (или наоборот) либо на цепочку 600C преобразований (или наоборот). Фиг. 9 иллюстрирует комбинированную цепочку 900 преобразований, которая включает в себя все комбинированные цепочки 600A, 600B, 600C и 600D преобразований.

[0068] Соответственно, с учетом цепочек 600A, 600B, 600C и 600D преобразований, ассоциированных с соответствующими устройствами 501-504 в окружении, предусмотрено 8 возможных составных приложений, которые могут формироваться (согласно цепочкам преобразований по фиг. 7A-7D, фиг. 8A-8C и фиг. 9). Таким образом, по мере того как цепочки преобразований различных устройств объединяются с и отсоединяются от окружения, непосредственно сама цепочка преобразований изменяется, и структура составного приложения за счет этого изменяется. Например, изменение значения источника 601A данных может оказывать существенно отличающееся влияние на цепочку преобразований, поскольку эффекты этого изменения автоматически распространяются посредством одного или более преобразований, в зависимости от того, находится или нет этот источник 601A данных только в одной цепочке 600A преобразований, в цепочке 700A преобразований, в цепочке 700B преобразований, в цепочке 700D преобразований, в цепочке 800A преобразований, в цепочке 800B преобразований, в цепочке 800C преобразований или в цепочке 900 преобразований.

[0069] В качестве примера, допустим, что устройство 501 сначала входит в окружение, что приводит к использованию цепочки 600A преобразований. Устройство 502 затем входит в окружение, что приводит к объединению цепочки 600B преобразований с цепочкой 600A преобразований, что приводит к цепочке 700A преобразований. Таким образом, рабочая цепочка преобразований изменяется с цепочки 600A преобразований на цепочку 700A преобразований. Теперь допустим, что третье устройство 503 входит в окружение, что приводит к объединению цепочки 600C преобразований с цепочкой 700A преобразований, в силу этого приводя к цепочке 800A преобразований. Таким образом, рабочая цепочка преобразований изменяется с цепочки 700A преобразований на цепочку 800A преобразований. Теперь предположим, что устройство 500B покидает окружение. Цепочка 700B преобразований затем должна становиться рабочей. Теперь допустим, что устройство 504 входит в окружение, что приводит к объединению цепочки 600BC преобразования с цепочкой 700B преобразований, что приводит к тому, что цепочка 800C преобразований становится рабочей цепочкой преобразований. Теперь предположим, что устройство 500C выходит, что приводит к рабочей цепочке 700D преобразований. Теперь предположим, что устройство 501 выходит, что приводит к рабочей цепочке 600D преобразований. В завершение, устройство 504 выходит, не оставляя рабочей цепочки преобразований в окружении. В этом сценарии, работающее приложение изменяет структуру (и за счет этого изменяет функциональность) семь раз следующим образом: 1) начало с цепочки 600A преобразований, 2) затем переключение на цепочку 700A преобразований, 3) затем переключение на цепочку 800A преобразований, 4) затем переключение на цепочку 700B преобразований, 5) затем переключение на цепочку 800C преобразований, 6) затем переключение на цепочку 700D преобразований, 7) затем завершение в цепочке 600D преобразований.

[0070] Фиг. 10 иллюстрирует блок-схему последовательности операций способа 1000 для подготовки к обеспечению составного приложения, представленного посредством

нескольких объединенных экземпляров цепочек преобразований. Способ 1000 может осуществляться посредством любого из устройств 501-507, которые находятся в окружении 500. Альтернативно или помимо этого, способ 1000 может осуществляться внешне для устройств 501-507, к примеру, посредством внешней системы 510 по фиг. 5. Например, внешняя система 510 может представлять собой услугу, предоставляемую в облачном вычислительном окружении. Независимо от того, где осуществляется способ 1000, фиг. 11 иллюстрирует систему 1100, которая может осуществлять способ 1000. Соответственно, независимо от того, включена или нет система 1100 в одно или более устройств 501-507, либо того, представляет собой система 1100 или нет внешнюю систему 510 по фиг. 5, либо от комбинаций вышеозначенного, далее описывается способ 1000 с частой ссылкой на систему 1100 по фиг. 11.

[0071] Способ 1000 включает в себя два этапа 1001 и 1002, которые не имеют временных зависимостей. Например, способ включает в себя идентификацию зависимостей между различными классами цепочек преобразований (этап 1001). Эти зависимости по существу представляют собой авторизации, созданные автором класса, который формирует класс, чтобы обеспечивать возможность экземплярам различных классов цепочек преобразований взаимодействовать (согласно дополнительному подтверждению на основе экземпляров).

[0072] Например, система 1100 включает в себя логический компонент 1101 и библиотеку 1110 определений классов преобразований, которая проиллюстрирована как включающая в себя шесть определений 1111-1116 классов цепочек преобразований. Тем не менее, эллипсы 1117 представляют то, что библиотека 1110 определения класса цепочек преобразований может включать в себя любое число определений классов цепочек преобразований. В качестве примера, допустим, что определения 1111, 1112, 1113 и 1114 классов цепочек преобразований, соответственно, задают классы 600A, 600B, 600C и 600D цепочек преобразований по фиг. 6A-6D. Поскольку каждый класс цепочек преобразований задает элементы зависимостей, система 1100 может использовать библиотеку 1110 для того, чтобы идентифицировать зависимости между различными классами цепочек преобразований.

[0073] Например, посредством наблюдения элемента 601B зависимостей цепочки 600A преобразований и элемента 601A зависимостей в цепочке 600B преобразований, логический компонент 1101 может идентифицировать зависимость между узлами 601A и 601B, которая должна существовать, если классы 600A цепочек и 600B преобразований были объединены, либо если экземпляры класса были объединены. Логический компонент 1101 может логически выводить то, что авторы цепочек 600A и 600B преобразований согласны авторизовать объединение (при условии, что пользователи также авторизуют на уровне экземпляра во время выполнения) в этих элементах зависимостей.

[0074] Способ 1000 также включает в себя идентификацию (этап 1002) устройств, которые ассоциированы с экземплярами любого из классов цепочек преобразований. Устройства могут сами идентифицироваться в системе 1100, возможно, с помощью идентификатора устройства, а также потенциально любых классов цепочек преобразований, которые устройство хочет использовать. Альтернативно, система 1100 может иметь конкретный класс преобразований по умолчанию, ассоциированный с каждым устройством, и за счет этого, возможно, принимать только идентификатор устройства. Могут быть предусмотрены некоторые классы цепочек преобразований, которые устройство лицензировано использовать, и другие классы цепочек преобразований, которые устройство не лицензировано использовать.

[0075] Например, система 1100 также включает в себя регистрации 1120 устройств, которые включают в себя регистрацию устройств как ассоциированных с различными классами цепочек преобразований. Например, регистрации 1120 устройств включают в себя пять регистраций 1121-1125 устройств, хотя эллипсы 1126 представляют то, что регистрации 1120 устройств могут включать в себя любое число регистраций устройств. Кроме того, регистрации 1120 устройств могут сохраняться в течение длительного срока и/или могут, возможно, сохраняться только в течение короткого времени. Независимо, система 1100 идентифицирует несколько устройств и находит ассоциированные цепочки преобразований для этих устройств. Только в качестве примера, возможно, устройства 501-504 по фиг. 5 ассоциированы с регистрациями 1121-1124. Посредством такого ассоциирования, система 1100 имеет возможность обнаруживать то, что экземпляры 600A, 600B, 600C, 600D цепочек преобразований (соответственно, заданные посредством определений 1111, 1112, 1113 и 1114 классов цепочек преобразований), соответственно, ассоциированы с этими устройствами 501-504.

[0076] В некоторых вариантах осуществления, сама система 1100 выполняет представление экземпляра конкретного класса цепочек преобразований от имени соответствующего устройства. Например, после идентификации того, что класс 600A цепочек преобразований ассоциирован с устройством 501, система 1100 может работать с представлением экземпляра 600A цепочки преобразований для устройства 501. Соответственно, по мере того как вводы принимаются из устройства, система 1100 идентифицирует изменение с помощью конкретного узла в цепочке преобразований и распространяет цепочки по всей цепочке преобразований. Это упоминается в качестве "практического подхода". Этот практический подход имеет некоторые преимущества в том, что обработка может разгружаться на устройства или системы, или облака, которые имеют большую способность к обработке, чем устройство, которое предоставляет непосредственно ввод.

[0077] В другом подходе, упоминаемом в данном документе в качестве "автоматического подхода", устройство ассоциировано с классом преобразований посредством отдельного выполнения экземпляра класса. Когда зависимость встречается с другой цепочкой преобразований, ассоциированной с другим устройством, данные могут протекать непосредственно (например, через сети между равноправными узлами) в это устройство, если это другое устройство также участвует в автоматическом подходе. Альтернативно, устройство может обеспечивать протекание данных косвенно в другое устройство (например, через внешнюю систему или облако). Если внешняя система 510 выполняет представление экземпляра класса цепочек преобразований для этого другого устройства, устройство вместо этого может обеспечивать протекание данных во внешнюю систему 510 для дополнительного распространения в остаток цепочки преобразований.

[0078] Фиг. 12 иллюстрирует блок-схему последовательности операций способа 1200 для объединения двух экземпляров классов преобразований. Во-первых, классы цепочек преобразований подтверждаются как имеющие тип, который может объединяться (этап 1201). Например, если классы цепочек преобразований не ссылаются друг на друга, то, возможно, отсутствует авторизация на основе классов для того, чтобы объединять экземпляры соответствующего класса. Например, класс 600D цепочек преобразований не может объединяться с классом 600B цепочек преобразований или классом 600C цепочек преобразований, поскольку класс 600D цепочек преобразований не ссылается на элементы в классе 600B или 600C цепочек преобразований. Тем не менее, класс 600D

цепочек преобразований может объединяться с классом 600A цепочек преобразований, поскольку они взаимно содержат ссылки зависимостей друг на друга.

[0079] Тем не менее, в этом примере, хотя класс 600D цепочек преобразований не может непосредственно объединяться с классами 600B и 600C цепочек преобразований, класс 600D цепочек преобразований может объединяться с цепочками 700A, 700B и 800A преобразований, даже если эти цепочки 700A, 700B и 800A преобразований включают в себя одну или обе из цепочек 600B и 600C преобразований.

[0080] Тем не менее, в альтернативном варианте осуществления, автор класса цепочек преобразований может указывать дополнительные ограничения на объединение других классов цепочек преобразований. Например, автор может указывать такое общее ограничение, что объединение конкретного экземпляра класса преобразований не разрешается, если этот экземпляр класса преобразований уже представляет собой составную цепочку преобразований, и/или если составная цепочка преобразований имеет конкретную составляющую цепочку преобразований. Например, при объединении двух экземпляров цепочек преобразований, система 1100 может отслеживать идентификационные данные классов преобразований, которые использованы для того, чтобы составлять составную цепочку преобразований вплоть до этой точки. Этот список может использоваться для того, чтобы подтверждать то, удовлетворены или нет условия для авторизации на основе классов.

[0081] Автор также может выражать ограничения со степенью детализации в одну зависимость. Например, в элементе 601B зависимостей класса 600A цепочек преобразований, автор может выражать то, что объединение авторизуется в этом элементе зависимостей, только если цепочка преобразований, к которой он объединяется, не включает в себя идентифицированный класс цепочек преобразований, созданный конкурентом. Автор также может управлять данными, которые вытекают из цепочки преобразований в другую объединенную цепочку преобразований, посредством записи ограничений или условий в преобразование, которые соединяют саму зависимость (например, между узлами 601A и элементом 601B зависимостей).

[0082] Тем не менее, даже если классы цепочек преобразований могут взаимодействовать, это не означает то, что пользователь хочет объединение своего конкретного экземпляра этого класса цепочек преобразований с другими экземплярами других классов цепочек преобразований. В конце концов, непосредственно данные (например, состояние экземпляра) могут быть конфиденциальными для пользователя. Соответственно, способ 1200 также включает в себя определение того, что экземпляры различных классов цепочек преобразований должны объединяться (этап 1202).

[0083] Критерии объединения для авторизации двух экземпляров различных классов цепочек преобразований для объединения могут включать в себя одно или более из следующего: присутствует или нет пользователь в списке участников собрания, отношения (например, семья, друг из социальной сети и т.п.) пользователей различных устройств, возможности связи (например, ближняя связь) между устройствами, близость соответствующих устройств (например, в идентичном конференц-зале), запрос пользователей и т.п. Например, критерии объединения могут включать в себя некоторые бизнес-критерии, к примеру, то, что ассоциированные пользователи экземпляров находятся в одной группе. В качестве другого примера, одно устройство может представлять собой киоск в торговом зале или отеле, при этом клиент использует киоск, и продавец или портье может автоматически использовать свое устройство для того, чтобы объединять цепочку преобразований с цепочкой преобразований киоска, чтобы за счет этого взаимодействовать с клиентом с использованием составного приложения.

Условия могут применяться к критериям объединения. Например, устройство коридорного может иметь возможность объединять приложение клиента, если портье нет рядом (что, возможно, обнаружено посредством отсутствия активного использования портье спариваемого приложения для объединения с приложением клиентов или нахождения вне сети).

[0084] Если только экземпляр класса цепочек преобразований, ассоциированного с устройством, не должен начинаться в состоянии по умолчанию, заданном посредством этого класса, при практическом подходе устройство затем может выгружать данные экземпляров (например, значения в каждом из узлов цепочки преобразований, ассоциированной с устройством) в систему 1100 (этап 1203). При автоматическом подходе, возможно, данные экземпляров только в граничных узлах выгружаются в систему 1100.

[0085] Экземпляр цепочки преобразований затем объединяется с остатком цепочки преобразований (этап 1204). Например, это может осуществляться посредством выполнения поиска в исходном экземпляре цепочки преобразований элементы зависимостей, которые соответствуют узлам в целевом экземпляре цепочки преобразований. После того, как такой элемент зависимостей найден, этот элемент зависимостей заменен фактическим идентифицированным узлом в целевом экземпляре цепочки преобразований, за счет этого создавая фактическую связь между исходными и целевыми экземплярами цепочек преобразований. Это повторяется для всех таких найденных элементов зависимостей в исходном экземпляре цепочки преобразований, который идентифицирует узел в целевом экземпляре цепочки преобразований. Если найдены элементы зависимостей в исходном экземпляре цепочки преобразований, которые не соответствуют узлу в целевом экземпляре цепочки преобразований, то эти элементы зависимостей остаются элементами цепочки зависимостей в объединенном экземпляре цепочки преобразований.

[0086] Следует отметить, что в примерах по фиг. 6A-6D, ни один из классов 600A-600D цепочек преобразований не включает в себя элементы зависимостей, которые ссылаются на идентичный класс. Тем не менее, принципы, описанные в данном документе, могут работать, даже если экземпляр конкретного класса цепочек преобразований может иметь элемент зависимостей, ссылающийся на другой узел в другом экземпляре идентичного конкретного класса цепочек преобразований. Кроме того, если разрешено посредством конкретного объединяющегося узла, несколько узлов из одного или более различных экземпляров цепочек преобразований могут объединяться с этим конкретным узлом.

[0087] После объединения, цепочка преобразований дополняется, и данные протекают в дополненной цепочке преобразований (этап 1205), фактически как если сначала цепочка преобразований первоначально была создана дополненным способом. Например, как только экземпляр класса 600A цепочек преобразований объединяется с экземпляром класса 600B цепочек преобразований, результат представляет собой один экземпляр класса 700A цепочек преобразований. После того, как экземпляр класса 600A цепочек преобразований объединяется с экземпляром класса 600C цепочек преобразований, результат представляет собой один экземпляр класса 700B цепочек преобразований. После того, как экземпляр класса 600B цепочек преобразований объединяется с экземпляром класса 600C цепочек преобразований, результат представляет собой один экземпляр класса 700C цепочек преобразований. После того, как экземпляр класса 600A цепочек преобразований объединяется с экземпляром класса 600D цепочек преобразований, результат представляет собой один экземпляр класса

700D цепочек преобразований.

[0088] Экземпляр класса 800A цепочек преобразований может формироваться либо посредством объединения экземпляров классов 700A и 600C цепочек преобразований, либо посредством объединения экземпляров класса 600A и 700C цепочек преобразований. Экземпляр класса 800B цепочек преобразований может формироваться посредством объединения экземпляров классов 700A и 600D цепочек преобразований. Экземпляр класса 800C цепочек преобразований может формироваться посредством объединения экземпляров классов 700B и 600D цепочек преобразований. Эта операция объединения может выполняться много раз, чтобы за счет этого создавать очень большую цепочку преобразований в ситуациях, в которых доступно множество устройств для совместной работы в данном окружении.

[0089] Соответственно, после объединения (этап 1204), данные могут протекать свободно (этап 1205) в дополненной цепочке преобразований даже через то, что представляет собой границы между составляющими цепочками преобразований. Например, в цепочке 700A преобразований по фиг. 7A, данные могут протекать свободно (без пользователя интерфейса прикладного программирования, и без вызовов функций) из узла 601A в узел 601B (через надлежащее преобразование(я)) точно так же, как данные могут вытекать из узла 601A в узел 604A (через надлежащее преобразование(я)).

[0090] Соответственно, система может использоваться для того, чтобы объединять преобразования, ассоциированные с широким спектром устройств, относительно автоматизированным и удобным способом, чтобы за счет этого дополнять цепочку преобразований составного приложения, которое совместно используется на множестве устройств. Это обеспечивает широкий спектр сценариев.

[0091] Например, рассмотрим людей, приходящих на собрание в конкретном конференц-зале. При входе первого пользователя в конференц-зал с переносным компьютером и смартфоном, экземпляр цепочки преобразований, ассоциированный со смартфоном, объединяется с экземпляром цепочки преобразований переносного компьютера, так что единое приложение эффективно выполняется на смартфоне и переносном компьютере первого пользователя. Второй пользователь входит в конференц-зал практически сразу же. Второй пользователь также приглашен на собрание, и в силу этого цепочка преобразований смартфона пользователя объединяется с существующей составной цепочкой преобразований, чтобы еще дополнять цепочку преобразований. Телевизионный экран затем автоматически включается, чтобы выводить некоторые данные, ассоциированные с собранием. Третий пользователь входит с планшетным компьютером и также является приглашенным на собрание. Таким образом, цепочка преобразований, ассоциированная с планшетным компьютером, объединяется с составной цепочкой преобразований, чтобы еще дополнять цепочку преобразований. Таким образом, по мере того, как дополнительные устройства объединяются с окружением, подходящем для объединения цепочек преобразований, приложение, работающее на всех устройствах, фактически дополняется, за счет этого изменяя функциональность самого приложения. Множество пользователей могут совместно работать в окружении, так что множество пользователей наблюдают каждое, по меньшей мере, из одного или некоторых устройств.

[0092] Фиг. 13 иллюстрирует блок-схему последовательности операций способа 1300 для разъединения цепочек преобразований. Как предусмотрен способ для объединения цепочек преобразований, когда устройства входят в окружение, так и предусмотрен способ для разъединения цепочек преобразований, когда устройства выходят из

окружения. Во-первых, система определяет то, что экземпляр класса цепочек преобразований должен разъединяться (этап 1301). В ответ, поток данных прекращается в/из того экземпляра (этап 1302). Критерии разъединения, используемые для определения того, что экземпляр должен отсоединяться, могут включать в себя любые критерии, но в качестве примера, могут включать в себя одно или более из близости устройства, ассоциированного с экземпляром, к другим устройствам в окружении, состояния завершения собрания, возможности связи между устройством, ассоциированным с экземпляром, и другими устройствами в окружении, запроса пользователя и т.д. Таким образом, цепочки преобразований могут изменяться динамически по мере того, как устройства входят и выходят из окружения.

[0093] В одном варианте осуществления, система 1100 включает в себя службу 1150 уровня представления. Один или более узлов экземпляра цепочки преобразований, который выполняется, могут управлять рендерингом на одном или более окружающих устройств. Аналогично, одно или более устройств могут предоставлять ввод в один или более узлов экземпляра цепочки преобразований. Например, допустим, что экземпляр цепочки 900 преобразований выполняется в окружении 500 по фиг. 5, и что устройства 501-504 ассоциированы с экземплярами соответствующих классов 600A-600D цепочек преобразований, в силу этого приводя к экземпляру дополненного класса 900 преобразований. Ссылаясь на фиг. 9, допустим, что узлы 601A и 601C представляют собой входные узлы, а узлы 604A, 603B, 602C и 602D представляют собой выходные узлы. Служба 1150 уровня представления может выбирать то, какие устройства могут предоставлять ввод в узлы 601A и 601C, и если да, какие преобразования должны выполняться. В одном примере, устройства ассоциированы с соответствующими частями своих цепочек преобразований, так что входные узлы в этой соответствующей части предоставляются посредством соответствующего устройства. Таким образом, по умолчанию, во входной узел 601A, первоначально представляющий собой часть цепочки 600A преобразований, может предоставляться ввод из устройства 501 (после потенциально некоторого преобразования(й)). Кроме того, в случае по умолчанию, во входной узел 601C, первоначально представляющий собой часть цепочки 600C преобразований, может предоставляться ввод из устройства 503 (после потенциально некоторого преобразования(й)). Служба 1150 уровня представления также может выбирать то, какие устройства должны осуществлять рендеринг вывод из узлов 604A, 603B, 602C и 602D, и какие (если имеются) должны выполняться преобразования.

[0094] Фиг. 14 иллюстрирует блок-схему последовательности операций способа 1400 для рендеринга вывода цепочки преобразований в окружении с несколькими устройствами. Способ 1400 может осуществляться посредством системы 1100 по фиг. 11 или возможно посредством службы 1150 уровня представления по фиг. 11. Логический компонент 1101 определяет (этап 1401) то, что изменяется конкретный узел (узел представления) экземпляра класса цепочек преобразований. Например, в примере, узлы 604A, 603B, 602C и 602D представляют собой выходные узлы или "узлы представления".

[0095] Логический компонент также идентифицирует характеристики (этап 1402) нескольких возможных устройств, на которых может осуществляться рендеринг, по меньшей мере, версии этого измененного вывода. Например, логический компонент 1101 может обращаться к реестру 1120 устройств и/или запрашивать устройства непосредственно, с тем чтобы определять возможность рендеринга каждого устройства. Характеристики каждого устройства также могут включать в себя сопоставление каждого пользователя относительно устройства.

[0096] Система затем выбирает (этап 1403) по меньшей мере одно из возможных

устройств, на которых можно осуществлять рендеринг по меньшей мере версии рендеринга, на основе идентифицированных характеристик. Система осуществляет выбор, чтобы максимизировать удобство и простоту использования вывода. Например, если вывод предназначен для всех пользователей, и все пользователи физически присутствуют, рендеринг (исполнение) вывода может осуществляться на большем экране. Если вывод является интерактивным и требует ввода из конкретного устройства, это устройство может выбираться для вывода.

[0097] Система определяет то, должны или нет какие-либо преобразования применяться (этап 1404 принятия решения) к рендерингу до фактического вывода в устройство. Преобразования учитывают применимость выбранного устройства, причем преобразования могут отличаться в зависимости от выбранного устройства для рендеринга. Примеры преобразований включают в себя изменение типа рендеринга (например, текст в речь, речь в текст, видео в текст, текст в видео и т.д.). Преобразования также могут включать в себя кинематизацию вывода. Например, видео может создаваться из ввода, в котором фон предоставляется и возможно изменяется надлежащим образом, и движущиеся элементы перемещаются в/из видео.

[0098] Если преобразования должны выполняться ("Да" на этапе 1404 принятия решения), то эти преобразования выполняются (этап 1405). В завершение, рендеринг обеспечивается на выбранном устройстве (этап 1406). Соответственно, составная цепочка преобразований может принимать ввод из любого устройства в любой форме и преобразовывать ввод, если требуется, в форму, которая распознается посредством входного узла составной цепочки преобразований. Таким образом, этот ввод распространяется по всей цепочке преобразований. Если значения одного или более узлов представления изменяются, надлежащее устройство может выбираться для рендеринга вывода. Соответственно, наиболее подходящее устройство для рендеринга может выбираться без учета того, какое устройство ассоциировано с частью цепочки преобразований, которая содержит узел представления. Например, рендеринг изменений значения узла 604A представления не должен осуществляться исключительно или вовсе не должен осуществляться на устройстве 501, которое соответствует экземпляру 600A цепочки преобразований. В некоторых вариантах осуществления, рендеринг может заключать в себе преобразование информации в некоторую физическую активацию, включающую в себя некоторый тип перемещения. Например, рендеринг может включать в себя открытие или закрытие двери, открытие или закрытие клапана, отпирание или запирание двери, включение или выключение телевизора и т.д.

[0099] Ниже описывается подробный сценарий (далее называемый "сценарием для устройств") относительно фиг. 15A-15M (совместно упоминаемых как "фиг. 15"). Этот сценарий предоставляется в качестве всего одного из бесконечного множества сценариев, которые обеспечиваются посредством более широких принципов, описанных в данном документе. В этом сценарии, пользователь заказывает различные вычислительные устройства. Сценарий заключает в себе четыре устройства 1501, 1502, 1503 и 1504, хотя сценарий начинается на фиг. 15A только с двух устройств 1501 и 1502, которые являются видимыми. Каждое из четырех устройств 1501-1504 участвует в предоставлении ввода в и приеме вывода из составного приложения.

[00100] Фиг. 15A вводит два устройства 1501 и 1502, которые представляют собой переносные компьютеры. Предположим, что эти два устройства 1501 и 1502 первоначально являются единственными устройствами в операционном окружении, и в силу этого их соответствующие цепочки преобразований объединены. Цепочка преобразований, ассоциированная с устройством 1501, предоставляет возможность

навигации и выбора изготовителя. Цепочка преобразований, ассоциированная с устройством 1502, предоставляет возможность навигации по списку устройств, предлагаемых посредством изготовителя, и также обеспечивает возможность пользователю изменять число любых из устройств в списке.

[00101] Более конкретно, устройство 1501 обеспечивает возможность пользователю горизонтально прокручивать определенное число изготовителей и выбирать изготовителя. На фиг. 15A устройство 1501 показывает только трех таких изготовителей, помеченных как "поставщик 7", "поставщик 8" и "поставщик 9", причем пользователь выбирает "поставщика 8". С другой стороны, устройство 1502 показывает горизонтально прокручиваемый список устройств, которые предоставляются посредством поставщика, выбранного в устройстве 1501. На фиг. 15A, поскольку поставщик 8 выбирается на устройстве 1501, устройство 1502 иллюстрирует список устройств, предоставленный посредством поставщика 8. Каждый элемент в списке устройств включает в себя изображение 1505 устройства, имя 1506 устройства, стоимость 1507 единицы устройства и число 1508 заказываемых устройств. Число для каждого устройства может управляться с использованием соответствующего элемента управления прокруткой.

[00102] Фиг. 15B показывает состояние сценария, который проистекает из состояния по фиг. 15A, в котором пользователь прокручивает список изготовителей влево и также выбирает "поставщика 4". Соответственно, вывод из устройства 1502 автоматически изменяется, чтобы отображать список устройств, изготовленных посредством поставщика 4. Для этого, данные протекают автоматически (без использования интерфейса прикладного программирования) из цепочки преобразований, ассоциированной с устройством 1501, в цепочку преобразований, ассоциированную с устройством 1502.

[00103] Фиг. 15C показывает состояние сценария, который проистекает из состояния по фиг. 15B, в котором пользователь прокручивает список изготовителей влево и также выбирает "поставщика 2". Соответственно, вывод из устройства 1502 автоматически изменяется, чтобы отображать список устройств, изготовленных посредством поставщика 2. Для этого, данные снова протекают автоматически из цепочки преобразований, ассоциированной с устройством 1501, в цепочку преобразований, ассоциированную с устройством 1502.

[00104] Фиг. 15D показывает состояние сценария, который проистекает из состояния по фиг. 15C, в котором пользователь использует элементы 1511 и 1512 управления прокруткой, отображаемые в устройстве 1502, чтобы изменять заказанное число для двух устройств, предлагаемых посредством поставщика 2. Фиг. 15E показывает состояние сценария, который проистекает из состояния по фиг. 15D, в котором пользователь использует устройство 1502, чтобы прокручивать вправо список устройств, предлагаемый посредством поставщика 2, и дополнительно использует элементы 1513 и 1514 управления прокруткой, отображаемые на устройстве 1502, чтобы изменять заказанное число для двух дополнительных устройств, предлагаемых посредством поставщика 2. Соответственно, на этой стадии, пользователь вводит заказанные числа для четырех устройств, предлагаемых посредством поставщика 2.

[00105] Фиг. 15F показывает состояние сценария, который проистекает из состояния по фиг. 15E, в котором третье устройство 1503 вводится в окружение. Например, третье устройство 1503 может быть ассоциировано с цепочкой преобразований, которая отображает максимальную стоимость единицы и совокупную стоимость для всего заказа. После того, как устройство 1503 вводится в окружение, эта часть цепочки преобразований объединяется с существующей составной цепочкой преобразований,

за счет этого изменяя функциональность составного приложения таким образом, чтобы теперь выводить такие данные стоимости в устройство 1503. После того, как экземпляр цепочки преобразований объединяется, данные, представляющие числа и стоимости единицы, протекают в эту часть цепочки преобразований, приводя к немедленному заполнению устройства 1503. Следует отметить, что потоки также возникают в противоположном направлении, поскольку устройство 1503 указывает максимальную цену за единицу, и в силу этого список устройств, показанный в устройстве 1502, ограничивается любым устройством ниже максимальной цены за единицу. В этом случае, все ранее перечисленные устройства опускаются ниже максимальной цены за единицу, и в силу этого отсутствует изменение визуализаций на устройстве 1502.

[00106] Фиг. 15G показывает состояние сценария, который проистекает из состояния по фиг. 15F, в котором пользователь использует элемент 1515 управления прокруткой в устройстве 1502, чтобы вводить число для еще одного другого устройства, предлагаемого посредством поставщика 2. Данные числа и данные стоимости единицы протекают в часть цепочки преобразований, соответствующей устройству 1503, что приводит к изменению совокупной стоимости, отображаемой на устройстве 1503.

[00107] Фиг. 15H показывает состояние сценария, который проистекает из состояния по фиг. 15G, в котором пользователь регулирует вниз максимальную цену за единицу до 987$ с использованием элемента 1516 управления прокруткой на устройстве 1503. Это максимальное изменение цены за единицу протекает из части цепочки преобразований, ассоциированной с устройством 1503, в часть цепочки преобразований, ассоциированную с устройством 1502, что приводит к тому, что несколько устройств, предлагаемых посредством поставщика 2, исключаются из списка. Если пользователь возвращает более высокую сумму, эти исключенные элементы должны автоматически добавляться обратно в список, возможно возвращая устройство 1502 в состояние, проиллюстрированное на фиг. 15G.

[00108] Фиг. 15I показывает состояние сценария, который проистекает из состояния по фиг. 15G, в котором четвертое устройство 1504 вводится в окружение. Например, четвертое устройство 1504 может быть ассоциировано с цепочкой преобразований, которая отображает максимальный вес устройства. После того, как устройство 1504 вводится в окружение, эта часть цепочки преобразований объединяется с существующей составной цепочкой преобразований, за счет этого изменяя функциональность составного приложения таким образом, чтобы теперь выводить такие данные стоимости. После того, как экземпляр цепочки преобразований объединяется, данные, представляющие максимальный вес, протекают из части цепочки преобразований, ассоциированной с устройством 1504, в часть цепочки преобразований, ассоциированную с устройством 1502. Устройство 1502 отвечает посредством отображения предупреждений 1518A, 1518B, 1518C и 1518D о перевесе, ассоциированных с каждым устройством, которое тяжелее максимального веса. В настоящее время, на фиг. 15I, вес составляет всего 606 граммов, и в силу этого всем устройствам отображается такое предупреждение.

[00109] Фиг. 15J показывает состояние сценария, который проистекает из состояния по фиг. 15I, в котором пользователь использует элемент 1516 управления прокруткой в устройстве 1504, чтобы увеличивать максимальный вес до более чем 2 килограммов. Изменение максимального веса протекает из части цепочки преобразований, ассоциированной с устройством 1504, в часть цепочки преобразований, соответствующую устройству 1502, что приводит к удалению предупреждений 1518C и 1518D о перевесе (предупреждения 1518A и 1518B о перевесе остаются).

[00110] Это завершает сценарий с точки зрения пользователя. Фиг. 15K-15O иллюстрируют то, что во время авторской разработки, автор может декларативно изменять преобразования, чтобы за счет этого изменять функциональность составного приложения. На фиг. 15K, элемент 1520 пользовательского интерфейса проиллюстрирован как включающий в себя декларативное преобразование, которое отфильтровывает те устройства, которые имеют предлагаемую розничную цену изготовителя, которая меньше значения, указываемого посредством ползунка 1516. Фиг. 15L иллюстрирует то, что это ограничение теперь удаляется, и в силу этого указанные устройства теперь не отфильтровываются посредством значения ползунка 1516.

[00111] На фиг. 15M, элемент 1521 пользовательского интерфейса проиллюстрирован как задающий цвет выделения яркостью, используемый для предлагаемой розничной цены изготовителя. Фиг. 15N иллюстрирует то, что цвет выделения яркостью теперь зависит от того, составляет или нет предлагаемая розничная цена изготовителя выше значения, указываемого посредством элемента управления-ползунка 1516. Если да, цвет выделения яркостью изменяется, например, на зеленый. Фиг. 15O иллюстрирует то, что по мере того, как элемент управления-ползунок 1516 продвигается в сторону увеличения в этом случае, поведение составного приложения изменяется с состояния, в котором устройства удаляются из списка, если их предлагаемая розничная цена изготовителя превышает значение элемента управления-ползунка 1516, на состояние, в котором предлагаемая розничная цена изготовителя (по сравнению со значением ползунка 1516) регулирует выделение яркостью предлагаемой розничной цены изготовителя.

[00112] Соответственно, описано формулирование и использование составного приложения, которое распределяется по множеству устройств. Кроме того, составное приложение фактически может существенно изменять форму в зависимости от доступных устройств. Таким образом, составное приложение может быть индивидуально адаптировано с возможностью эффективно использовать ресурсы, предоставленные посредством различных устройств в окружении.

[00113] Настоящее изобретение может быть осуществлено в других характерных формах без отступления от своей сущности или неотъемлемых характеристик. Описанные варианты осуществления должны рассматриваться во всех отношениях только как иллюстративные, а не ограничивающие. Следовательно, объем изобретения указан посредством прилагаемой формулы изобретения, а не посредством вышеприведенного описания. Все изменения, которые подпадают под смысл и рамки равнозначности формулы изобретения, должны охватываться ее объемом.

(57) Формула изобретения

1. Реализуемый компьютером способ для обеспечения составного приложения на множестве устройств, причем способ содержит:

этап, на котором идентифицируют зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований посредством, по меньшей мере, получения определения первого класса цепочек преобразований и определения второго класса цепочек преобразований, которые взаимно задают зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований;

этап, на котором идентифицируют первое устройство, которое ассоциировано с первым экземпляром первого класса цепочек преобразований;

этап, на котором идентифицируют второе устройство, которое ассоциировано со

вторым экземпляром второго класса цепочек преобразований;

этап, на котором определяют то, что первый экземпляр и второй экземпляр должны объединяться;

в ответ на определение того, что первый экземпляр и второй экземпляр должны объединяться, объединение первого экземпляра и второго экземпляра; и

этап, на котором обеспечивают протекание данных между первым и вторым экземплярами в соответствии с идентифицированными зависимостями между первым классом преобразований и вторым классом преобразований и в соответствии с состоянием зависимостей между первым и вторым экземплярами.

2. Способ по п. 1, дополнительно содержащий:

этап, на котором идентифицируют зависимости между первым классом цепочек преобразований и третьим классом цепочек преобразований;

этап, на котором идентифицируют третье устройство, которое работает с третьим экземпляром третьего класса цепочек преобразований;

этап, на котором определяют, что первый экземпляр и третий экземпляр должны быть объединены; и

в ответ на определение того, что первый экземпляр и третий экземпляр должны быть объединены, этап, на котором обеспечивают протекание данных между первым и третьим экземплярами в соответствии с идентифицированными зависимостями между первым классом преобразований и третьим классом преобразований и в соответствии с состоянием зависимостей между первым и третьим экземплярами.

3. Способ по п. 2, дополнительно содержащий:

этап, на котором идентифицируют зависимости между вторым классом цепочек преобразований и третьим классом цепочек преобразований;

этап, на котором определяют, что второй экземпляр и третий экземпляр должны быть объединены; и

в ответ на определение того, что второй экземпляр и третий экземпляр должны быть объединены, этап, на котором обеспечивают протекание данных между вторым и третьим экземплярами в соответствии с идентифицированными зависимостями между вторым классом преобразований и третьим классом преобразований и в соответствии с состоянием зависимостей между вторым и третьим экземплярами.

4. Способ по п. 1, дополнительно содержащий:

этап, на котором работают с представлением первого экземпляра первого класса цепочек преобразований от имени первого устройства и преобразуют первый экземпляр в ответ на ввод из первого устройства ввода.

5. Способ по п. 4, в котором первый экземпляр первого класса цепочек преобразований содержит один или более узлов, которые управляют визуализациями на первом устройстве.

6. Способ по п. 5, в котором второй экземпляр второго класса цепочек преобразований содержит один или более узлов, которые управляют визуализациями на втором устройстве.

7. Способ по п.5, дополнительно содержащий:

этап, на котором преобразуют вывод одного или более узлов перед обеспечением визуализации первому устройству.

8. Способ по п.4, дополнительно содержащий:

этап, на котором работают с представлением второго экземпляра второго класса цепочек преобразований от имени второго устройства и преобразуют второй экземпляр в ответ на ввод из второго устройства ввода.

9. Способ по п.8, в котором второй экземпляр второго класса цепочек преобразований содержит один или более узлов, которые управляют визуализациями на втором устройстве.

10. Способ по п.9, дополнительно содержащий:

этап, на котором преобразуют вывод одного или более узлов второго экземпляра перед обеспечением визуализации второму устройству.

11. Способ по п.4, в котором второе устройство ассоциировано со вторым экземпляром второго класса цепочек преобразования посредством выполнения вторым устройством второго экземпляра второго класса цепочек преобразования на втором устройстве.

12. Способ по п. 1, дополнительно содержащий:

этап, на котором определяют то, что второй экземпляр должен отсоединяться от первого экземпляра; и

в ответ на определение того, что второй экземпляр должен отсоединяться от первого экземпляра, этап, на котором прекращают протекание данных между первым и вторым экземплярами.

13. Способ по п. 12, в котором этап определения, что первый экземпляр и второй экземпляр должны разъединяться, выполняется в соответствии с одним или более критериями отсоединения, включающими в себя, по меньшей мере, близость первого и второго устройств.

14. Способ по п. 12, в котором этап определения, что первый экземпляр и второй экземпляр должны разъединяться, выполняется в соответствии с состоянием завершения собрания.

15. Способ по п. 12, в котором этап определения, что первый экземпляр и второй экземпляр должны разъединяться, выполняется в соответствии с возможностью связи между первым устройством и вторым устройством.

16. Способ по п. 1, в котором этап определения, что первый экземпляр и второй экземпляр должны объединяться, выполняется в соответствии с одним или более критериями объединения, включающими в себя, по меньшей мере, близость первого и второго устройств.

17. Способ по п. 1, в котором этап определения, что первый экземпляр и второй экземпляр должны объединяться, выполняется в соответствии с одним или более критериями объединения, включающими в себя, по меньшей мере, список участников собрания.

18. Способ по п. 1, в котором этап определения, что первый экземпляр и второй экземпляр должны объединяться, выполняется в соответствии с отношением между пользователем первого устройства и пользователем второго устройства.

19. Способ по п. 1, в котором этап определения, что первый экземпляр и второй экземпляр должны объединяться, выполняется в соответствии с возможностью связи между первым устройством и вторым устройством.

20. Способ по п. 1, при этом способ осуществляется посредством внешней системы, которая не включает в себя первое устройство или второе устройство.

21. Способ по п.1, причем способ выполняется первым устройством.

22. Способ по п.1, причем способ выполняется вторым устройством.

23. Считываемый компьютером носитель, имеющий исполняемые компьютером инструкции, которые являются исполняемыми одним или более процессорами вычислительной системы, чтобы инструктировать вычислительной системе осуществить способ для обеспечения составного приложения на множестве устройств, причем способ

содержит:

этап, на котором идентифицируют зависимости между первым классом цепочек преобразований, который включает в себя первый экземпляр класса цепочек преобразований, и вторым классом цепочек преобразований, который включает в себя второй экземпляр класса цепочек преобразований, посредством, по меньшей мере, получения определения первого класса цепочек преобразований и определения второго класса цепочек преобразований, которые взаимно задают зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований;

этап, на котором определяют то, что первый экземпляр цепочки преобразований и второй экземпляр цепочки преобразований должны объединяться;

в ответ на определение того, что первый экземпляр цепочки преобразований и второй экземпляр цепочки преобразований должны объединяться, объединение первого экземпляра цепочки преобразований и второго экземпляра цепочки преобразований; и

этап, на котором обеспечивают протекание данных между первым и вторым экземплярами цепочки преобразований в соответствии с идентифицированными зависимостями и в соответствии с состоянием зависимостей между первым и вторым экземплярами цепочки преобразований.

24. Вычислительная система для обеспечения составного приложения на множестве устройств, содержащая:

один или более процессоров; и

одно или более аппаратных считываемых компьютером запоминающих устройств, имеющих исполняемые компьютером инструкции, которые являются исполняемыми одним или более процессорами, чтобы инструктировать вычислительной системе осуществить:

идентификацию зависимостей между первым классом цепочек преобразований, который включает в себя первый экземпляр класса цепочек преобразований, и вторым классом цепочек преобразований, который включает в себя второй экземпляр класса цепочек преобразований, посредством, по меньшей мере, получения определения первого класса цепочек преобразований и определения второго класса цепочек преобразований, которые взаимно задают зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований;

определение того, что первый экземпляр цепочки преобразований и второй экземпляр цепочки преобразований должны объединяться;

в ответ на определение, что первый экземпляр цепочки преобразований и второй экземпляр цепочки преобразований должны объединяться, объединение первого экземпляра цепочки преобразований и второго экземпляра цепочки преобразований; и

обеспечение протекания данных между первым и вторым экземплярами цепочки преобразований в соответствии с идентифицированными зависимостями и в соответствии с состоянием зависимостей между первым и вторым экземплярами цепочки преобразований.

25. Вычислительная система по п. 24, в которой исполняемые компьютером инструкции дополнительно инструктируют вычислительной системе осуществить:

идентификацию первого устройства, которое ассоциировано с первым экземпляром класса цепочек преобразований;

идентификацию второго устройства, которое ассоциировано со вторым экземпляром класса цепочек преобразований, при этом второе устройство ассоциировано со вторым

экземпляром второго класса цепочек преобразования посредством выполнения вторым устройством второго экземпляра второго класса цепочек преобразования на втором устройстве;

работу с представлением первого экземпляра класса цепочек преобразований от имени первого устройства и преобразовать этот первый экземпляр класса цепочек преобразований в ответ на ввод из первого устройства.

26. Реализуемый компьютером способ для обеспечения составного приложения на множестве устройств, причем способ содержит:

этап, на котором идентифицируют зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований;

этап, на котором идентифицируют первое устройство, которое ассоциировано с первым экземпляром первого класса цепочек преобразований;

этап, на котором идентифицируют второе устройство, которое ассоциировано со вторым экземпляром второго класса цепочек преобразований;

этап, на котором определяют то, что первый экземпляр и второй экземпляр должны объединяться;

в ответ на определение того, что первый экземпляр и второй экземпляр должны объединяться, объединение первого экземпляра и второго экземпляра;

этап, на котором обеспечивают протекание данных между первым и вторым экземплярами в соответствии с идентифицированными зависимостями между первым классом преобразований и вторым классом преобразований и в соответствии с состоянием зависимостей между первым и вторым экземплярами;

этап, на котором определяют то, что второй экземпляр должен отсоединяться от первого экземпляра; и

в ответ на определение, что второй экземпляр должен отсоединяться от первого экземпляра, этап, на котором прекращают протекание данных между первым и вторым экземплярами.

27. Реализуемый компьютером способ для обеспечения составного приложения на множестве устройств, причем способ содержит:

этап, на котором идентифицируют зависимости между первым классом цепочек преобразований и вторым классом цепочек преобразований;

этап, на котором идентифицируют первое устройство, которое ассоциировано с первым экземпляром первого класса цепочек преобразований;

этап, на котором идентифицируют второе устройство, которое ассоциировано со вторым экземпляром второго класса цепочек преобразований;

этап, на котором определяют то, что первый экземпляр и второй экземпляр должны объединяться в соответствии с одним или более критериями объединения, включающими в себя, по меньшей мере, близость первого и второго устройств;

в ответ на определение того, что первый экземпляр и второй экземпляр должны объединяться, объединение первого экземпляра и второго экземпляра; и

этап, на котором обеспечивают протекание данных между первым и вторым экземплярами в соответствии с идентифицированными зависимостями между первым классом преобразований и вторым классом преобразований и в соответствии с состоянием зависимостей между первым и вторым экземплярами.

28. Способ по п. 27, дополнительно содержащий:

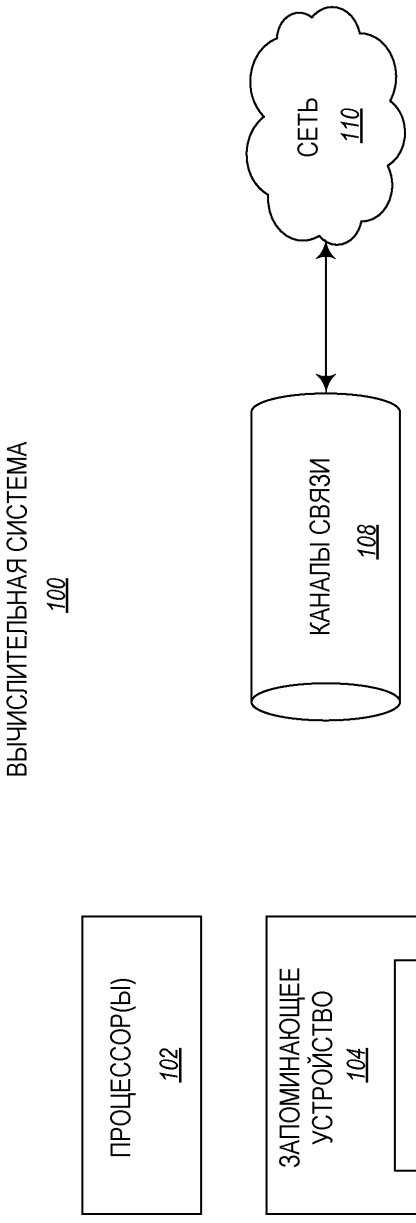
этап, на котором определяют то, что второй экземпляр должен отсоединяться от первого экземпляра; и

в ответ на определение того, что второй экземпляр должен отсоединяться от первого

экземпляра, этап, на котором прекращают протекание данных между первым и вторым экземплярами.

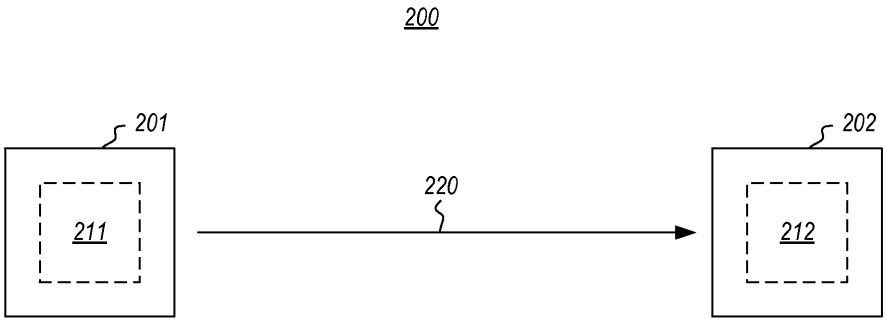
29. Способ по п. 28, в котором этап определения, что первый экземпляр и второй экземпляр должны разъединяться, выполняется в соответствии с одним или более критериями отсоединения, включающими в себя, по меньшей мере, близость первого и второго устройств.

1

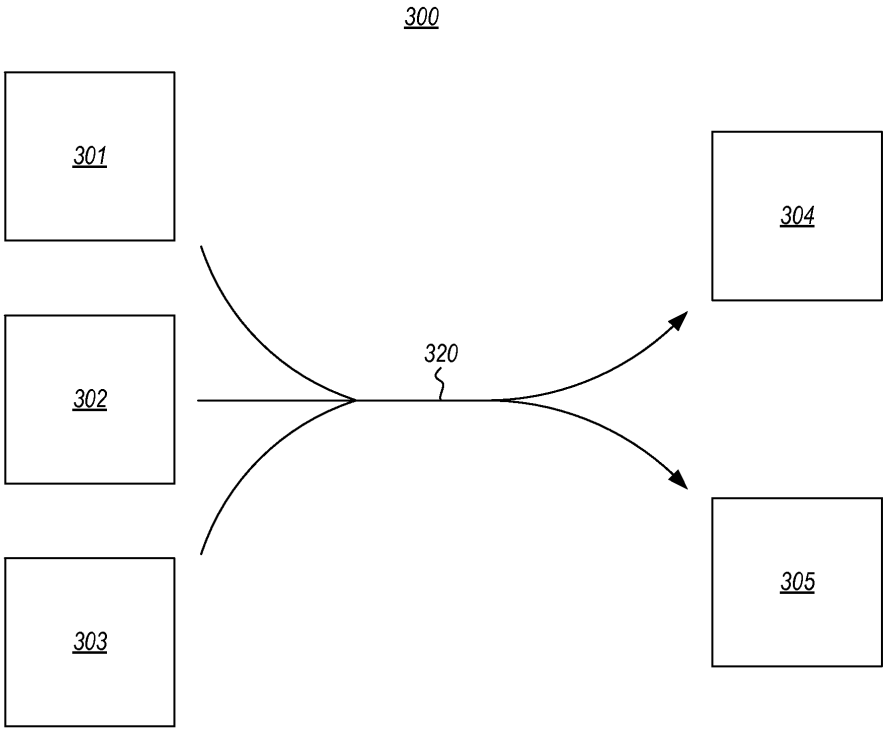

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА
100
ПРОЦЕССОР(Ы)
102
ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО
104
ЭНЕРГОЗАВИСИМОЕ
ЭНЕРГОНЕЗАВИСИМОЕ
КАНАЛЫ СВЯЗИ
108
СЕТЬ
110

ФИГ. 1

2

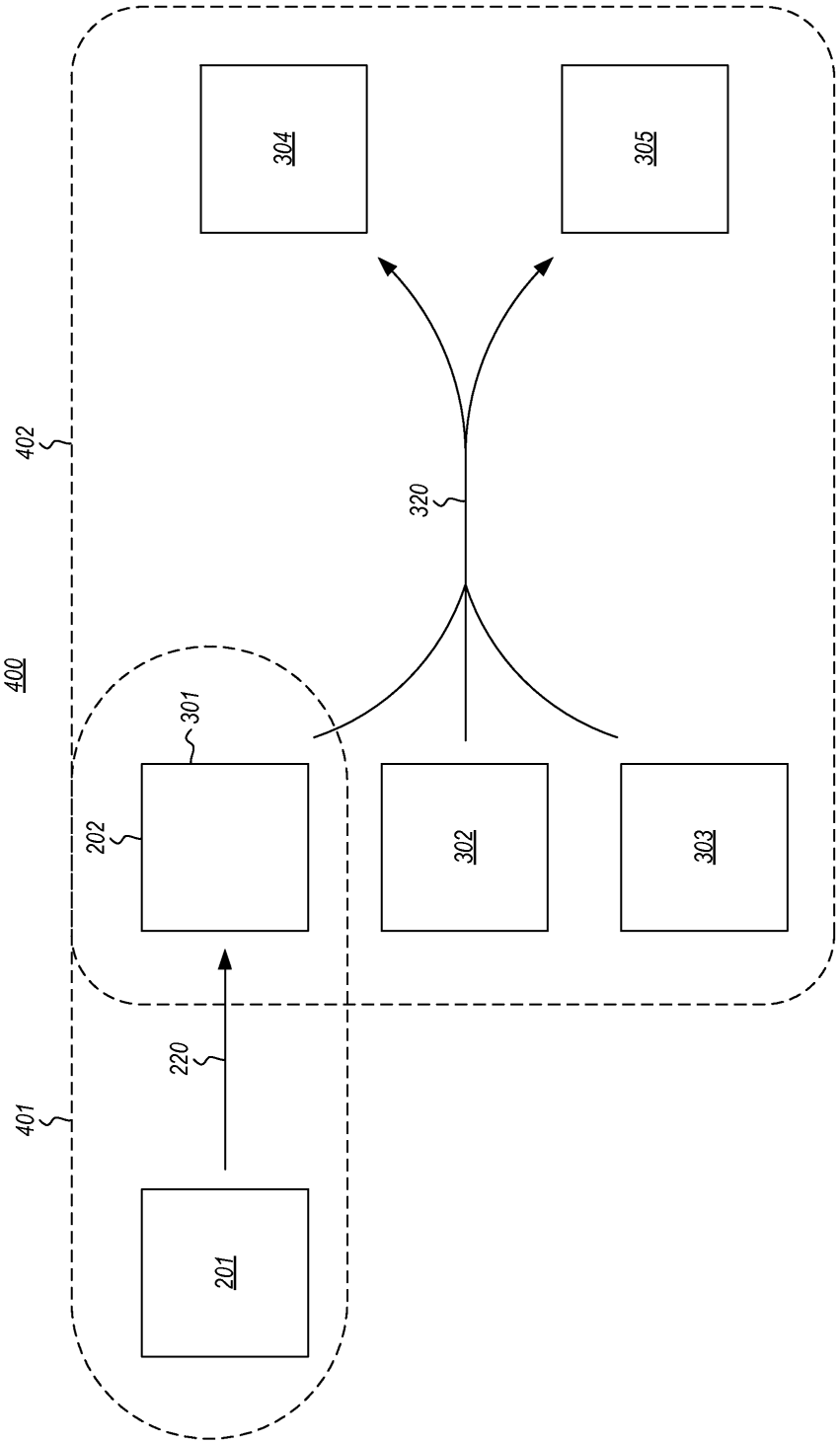


ФИГ. 2



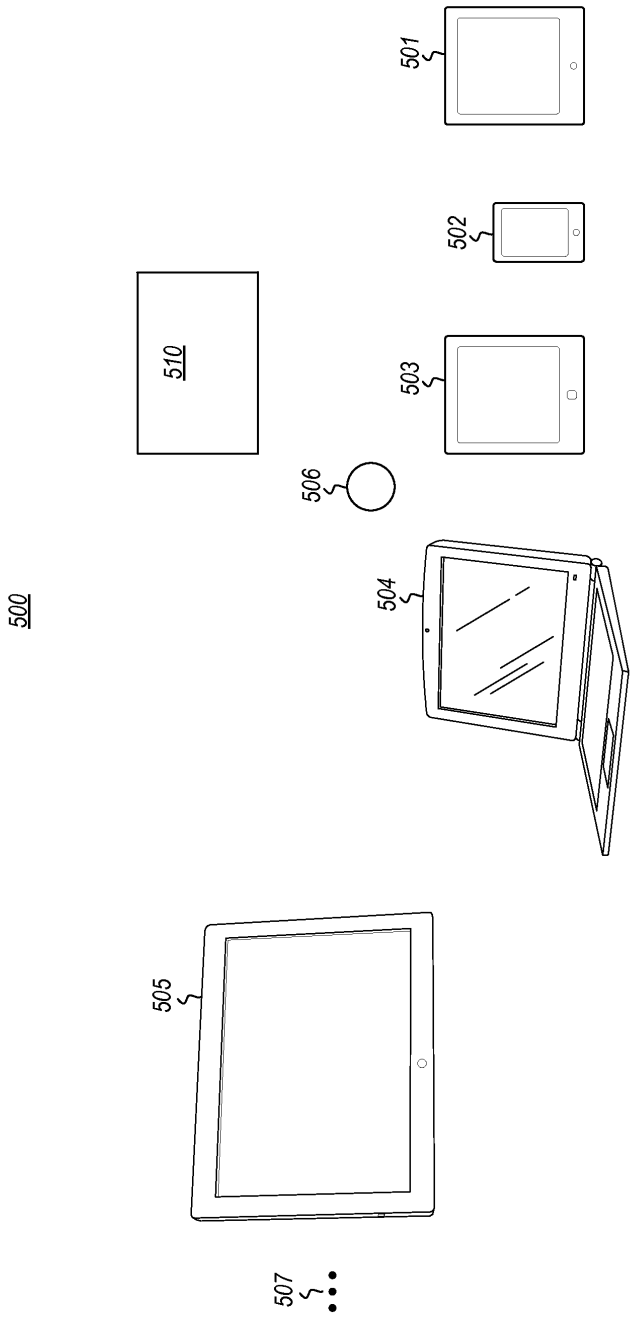
ФИГ. 3

3



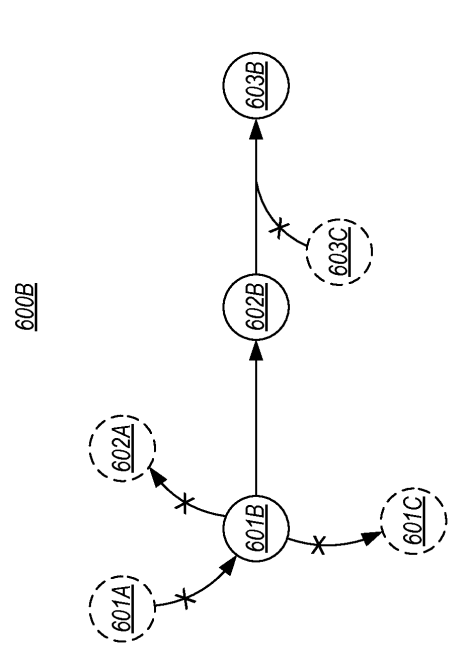
ФИГ. 4

4

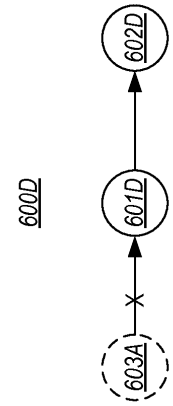


ФИГ. 5

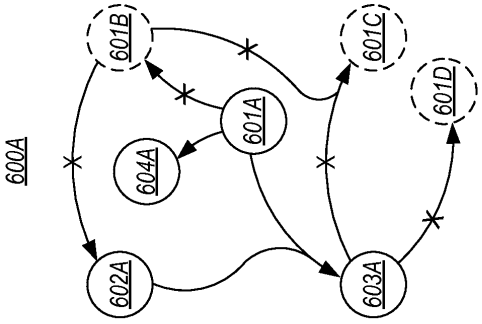
5



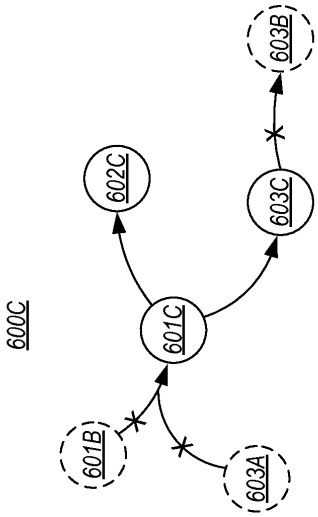
ФИГ. 6B



ФИГ. 6D

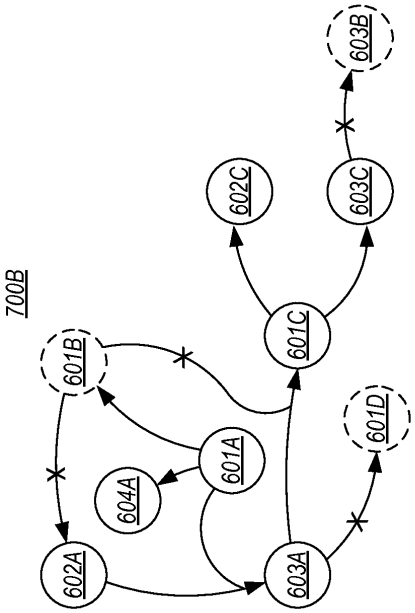


ФИГ. 6A

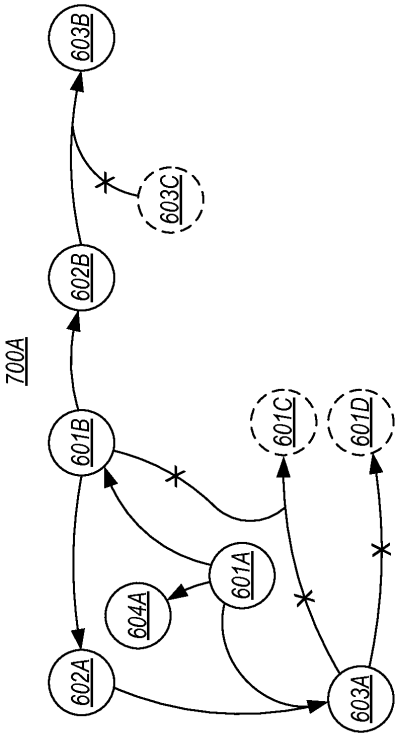


ФИГ. 6C

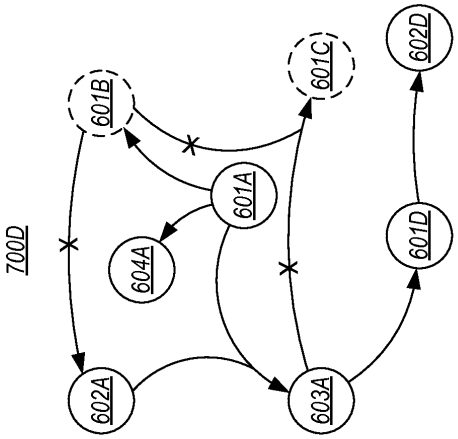
6



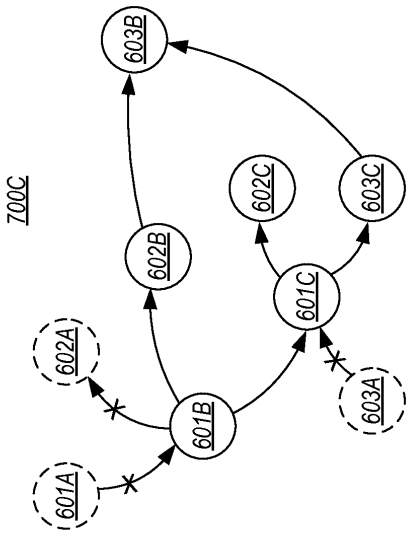
ФИГ. 7B



ФИГ. 7A

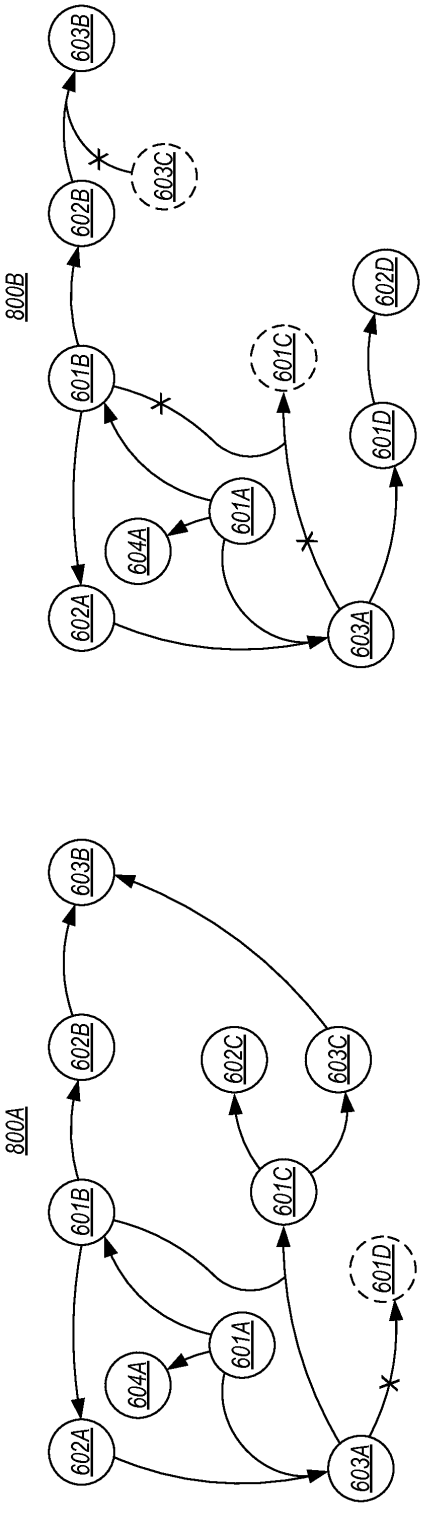


ФИГ. 7D



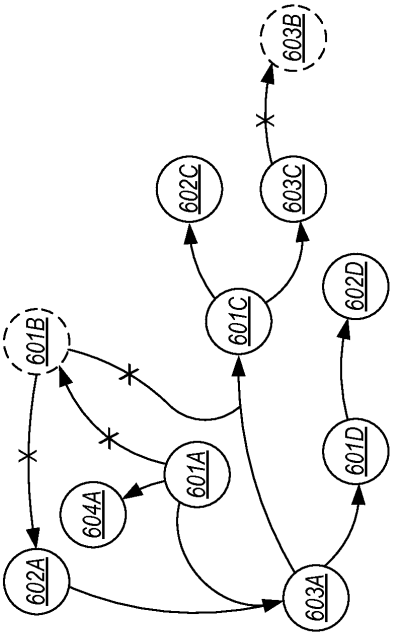
ФИГ. 7C

7

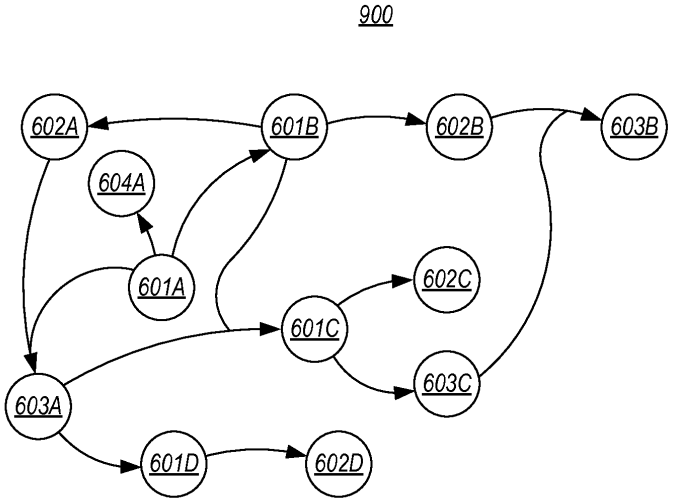


ФИГ. 8A

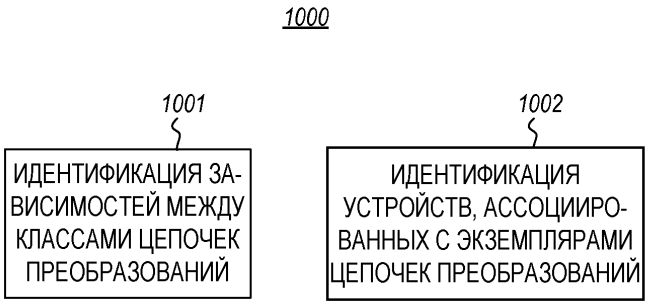
ФИГ. 8B



ФИГ. 8C

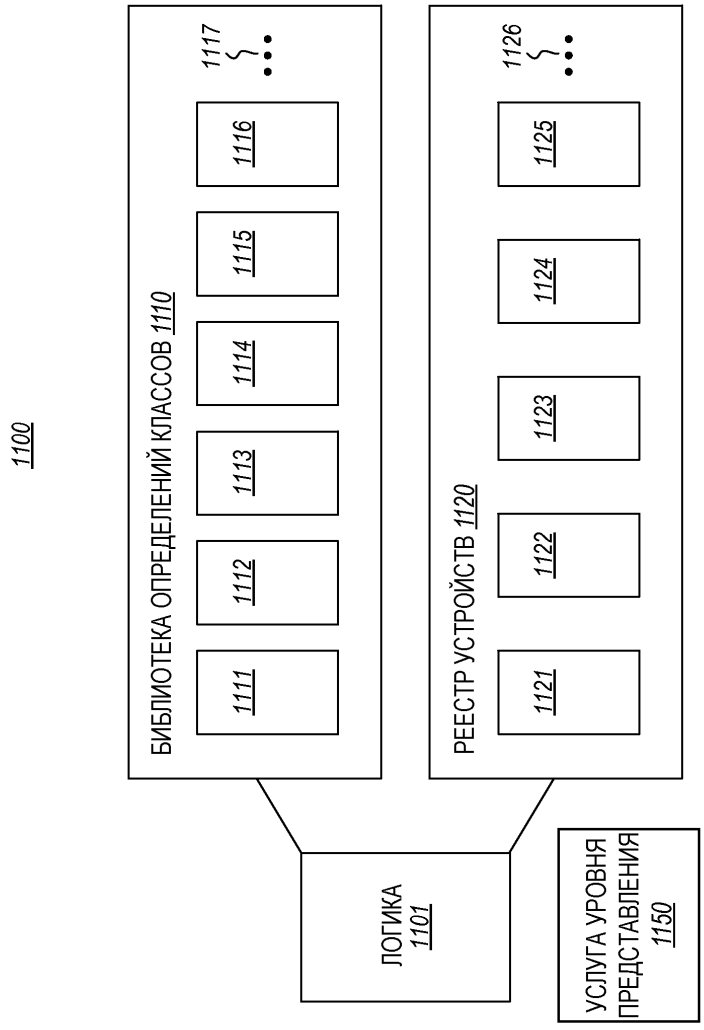

900
602A
601B
602B
603B
604A
601A
602C
601C
603C
603A
601D
602D

ФИГ. 9

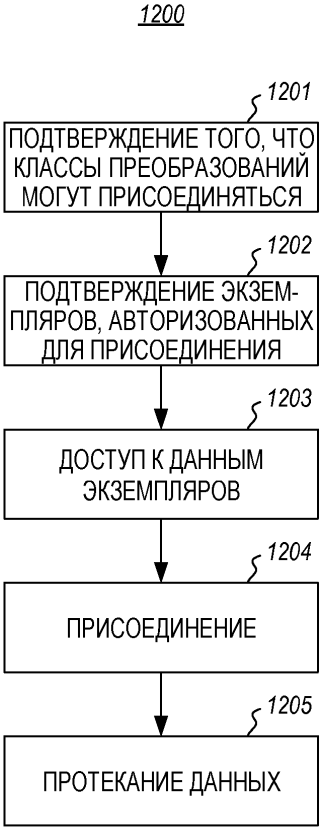

1000
1001
ИДЕНТИФИКАЦИЯ ЗАВИСИМОСТЕЙ МЕЖДУ КЛАССАМИ ЦЕПОЧЕК ПРЕОБРАЗОВАНИЙ
1002
ИДЕНТИФИКАЦИЯ УСТРОЙСТВ, АССОЦИИРОВАННЫХ С ЭКЗЕМПЛЯРАМИ ЦЕПОЧЕК ПРЕОБРАЗОВАНИЙ

ФИГ. 10

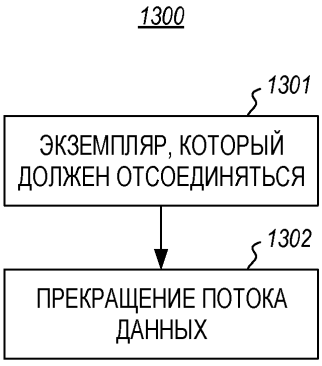
9


1100
БИБЛИОТЕКА ОПРЕДЕЛЕНИЙ КЛАССОВ 1110
1111
1112
1113
1114
1115
1116
1117
РЕЕСТР УСТРОЙСТВ 1120
1121
1122
1123
1124
1125
1126
ЛОГИКА 1101
УСЛУГА УРОВНЯ ПРЕДСТАВЛЕНИЯ 1150

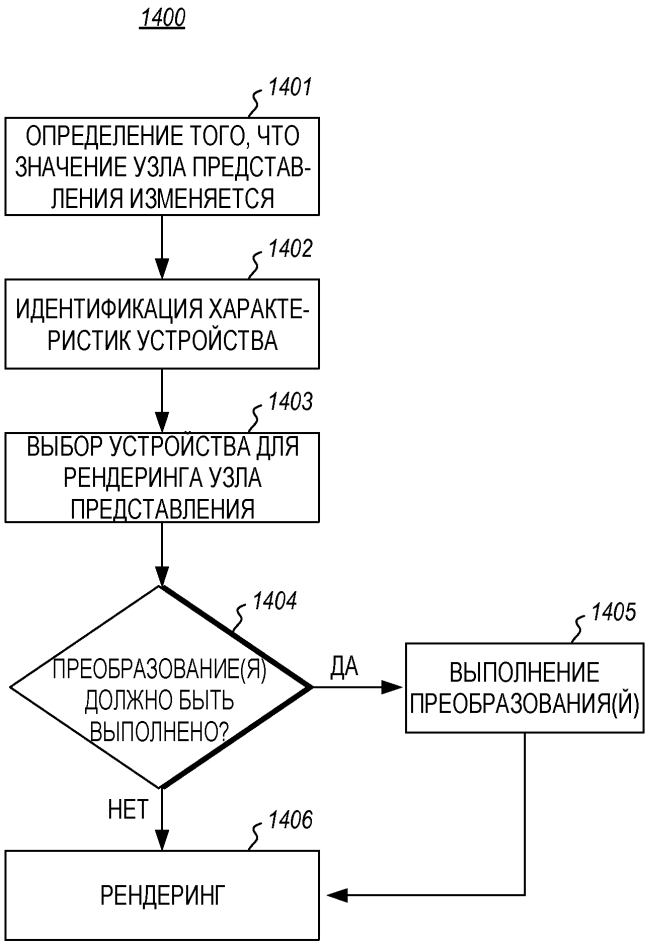
ФИГ. 11


1200
1201
ПОДТВЕРЖДЕНИЕ ТОГО, ЧТО КЛАССЫ ПРЕОБРАЗОВАНИЙ МОГУТ ПРИСОЕДИНЯТЬСЯ
1202
ПОДТВЕРЖДЕНИЕ ЭКЗЕМПЛЯРОВ, АВТОРИЗОВАННЫХ ДЛЯ ПРИСОЕДИНЕНИЯ
1203
ДОСТУП К ДАННЫМ ЭКЗЕМПЛЯРОВ
1204
ПРИСОЕДИНЕНИЕ
1205
ПРОТЕКАНИЕ ДАННЫХ

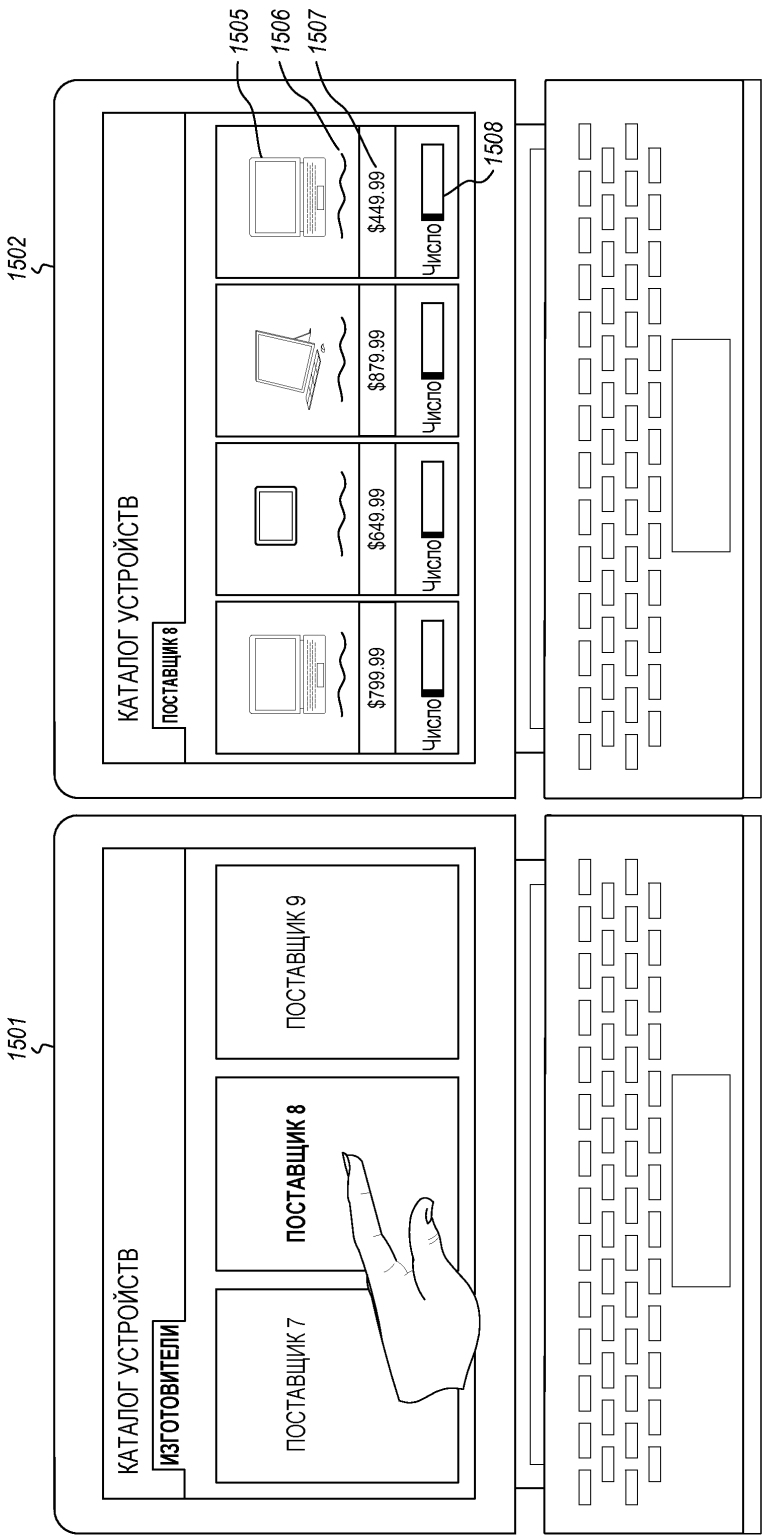
ФИГ. 12


1300
1301
ЭКЗЕМПЛЯР, КОТОРЫЙ ДОЛЖЕН ОТСОЕДИНЯТЬСЯ
1302
ПРЕКРАЩЕНИЕ ПОТОКА ДАННЫХ

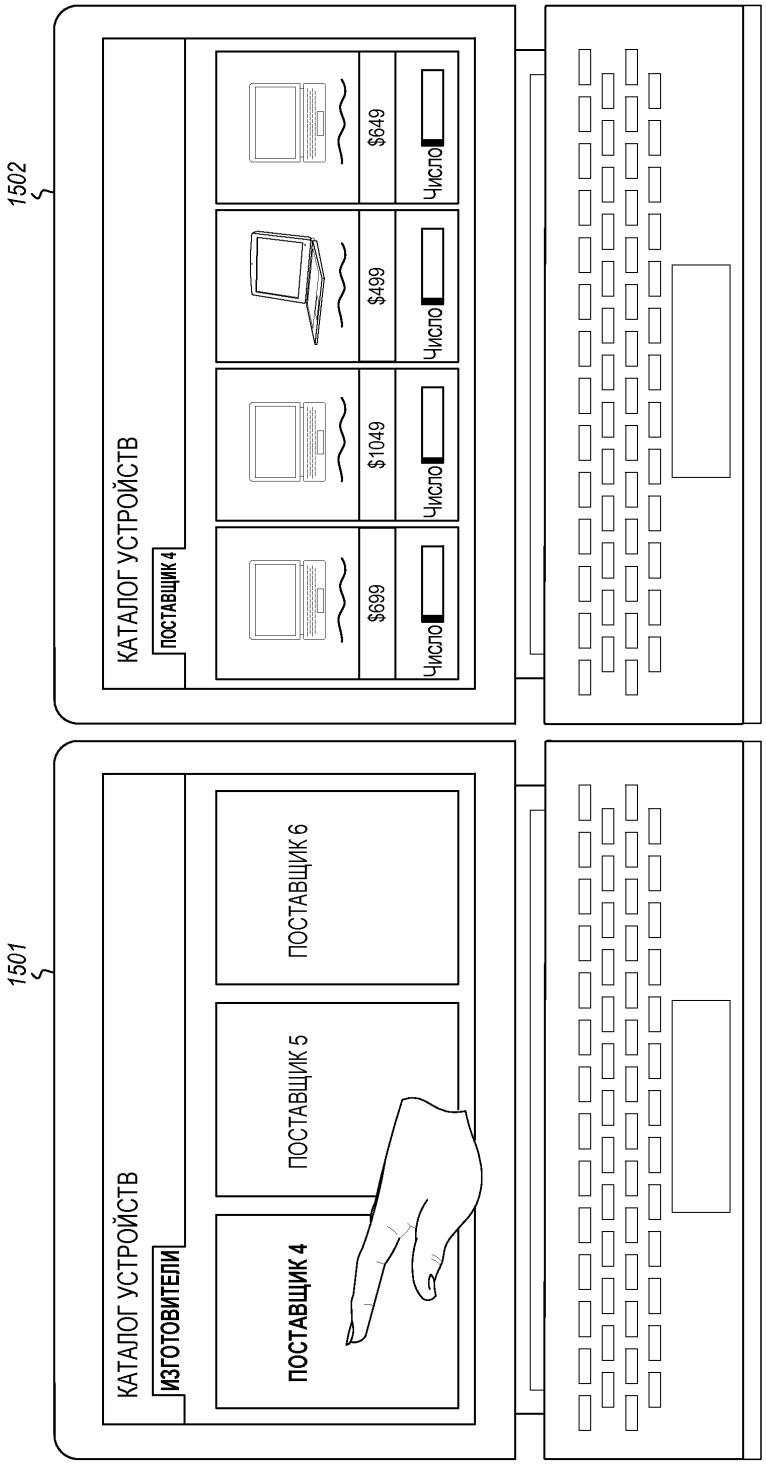
ФИГ. 13


1400
1401
ОПРЕДЕЛЕНИЕ ТОГО, ЧТО ЗНАЧЕНИЕ УЗЛА ПРЕДСТАВЛЕНИЯ ИЗМЕНЯЕТСЯ
1402
ИДЕНТИФИКАЦИЯ ХАРАКТЕРИСТИК УСТРОЙСТВА
1403
ВЫБОР УСТРОЙСТВА ДЛЯ РЕНДЕРИНГА УЗЛА ПРЕДСТАВЛЕНИЯ
1404
ПРЕОБРАЗОВАНИЕ(Я) ДОЛЖНО БЫТЬ ВЫПОЛНЕНО?
ДА
1405
ВЫПОЛНЕНИЕ ПРЕОБРАЗОВАНИЯ(Й)
НЕТ
1406
РЕНДЕРИНГ

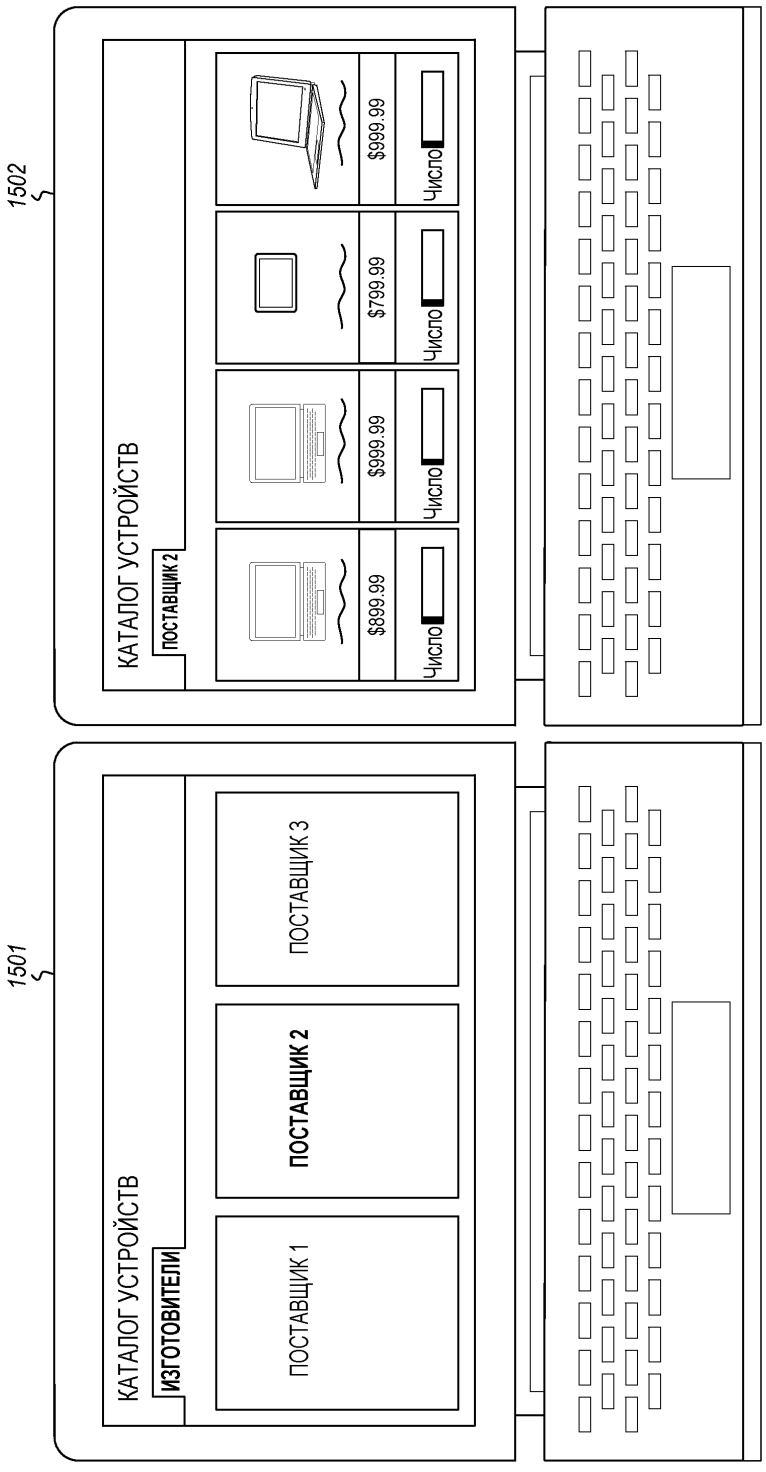
ФИГ. 14


1501
КАТАЛОГ УСТРОЙСТВ
ИЗГОТОВИТЕЛИ
ПОСТАВЩИК 7
ПОСТАВЩИК 8
ПОСТАВЩИК 9
1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 8
1505
1506
1507
1508
$799.99
$649.99
$879.99
$449.99
Число
Число
Число
Число

ФИГ. 15А

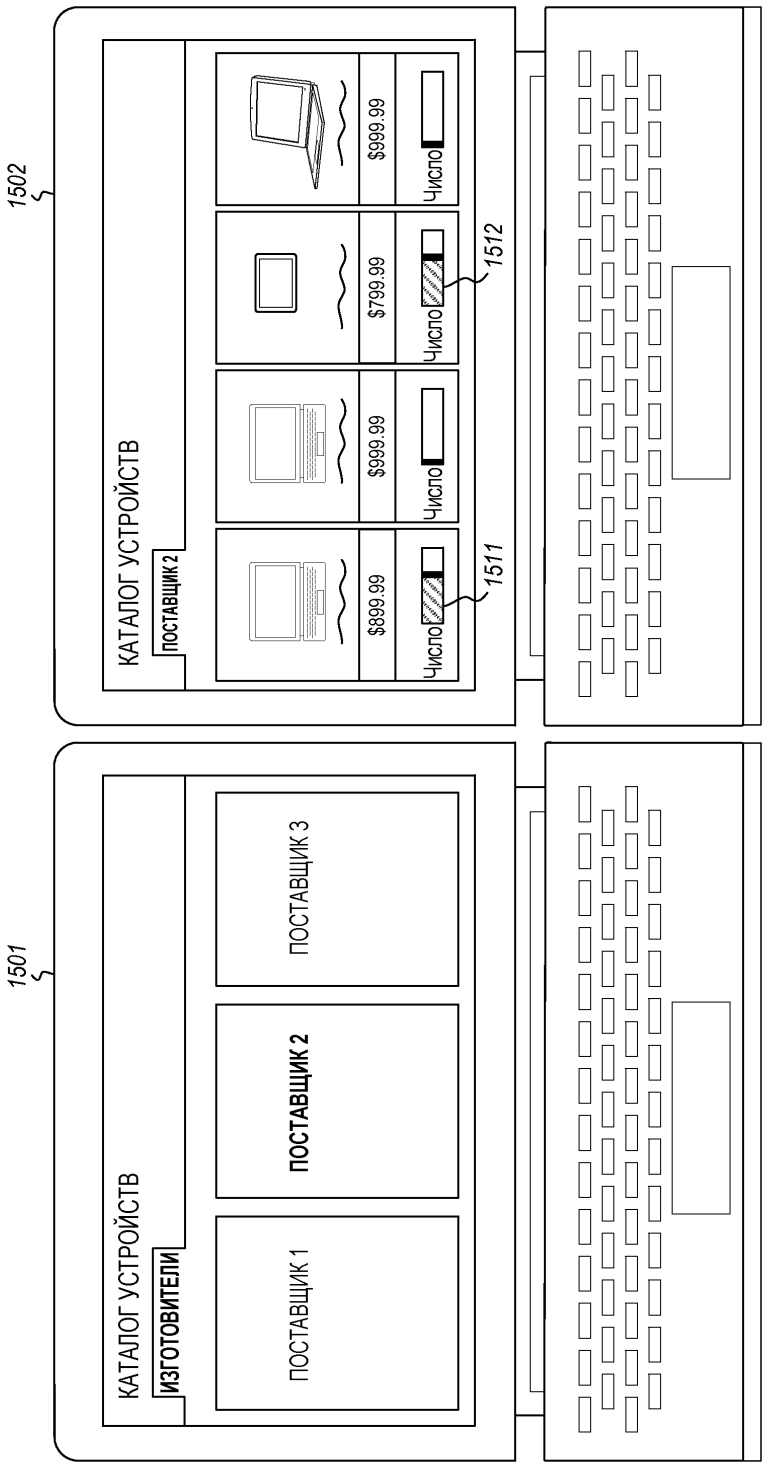

1501
КАТАЛОГ УСТРОЙСТВ
ИЗГОТОВИТЕЛИ
ПОСТАВЩИК 4
ПОСТАВЩИК 5
ПОСТАВЩИК 6
1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 4
$699
$1049
$499
$649
Число
Число
Число
Число

ФИГ. 15В

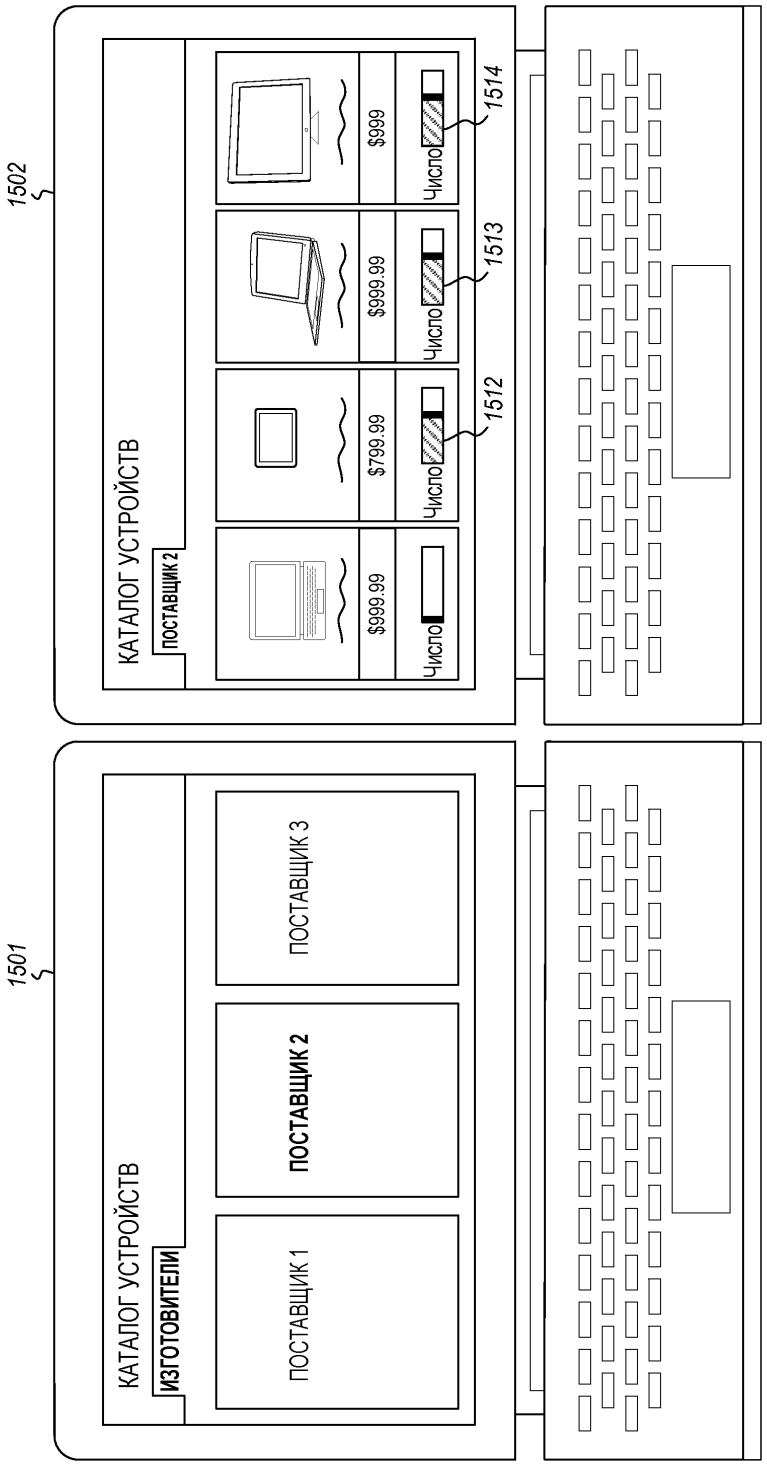

1501
КАТАЛОГ УСТРОЙСТВ
ИЗГОТОВИТЕЛИ
ПОСТАВЩИК 1
ПОСТАВЩИК 2
ПОСТАВЩИК 3
1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$899.99
$999.99
$799.99
$999.99
Число
Число
Число
Число

ФИГ. 15C

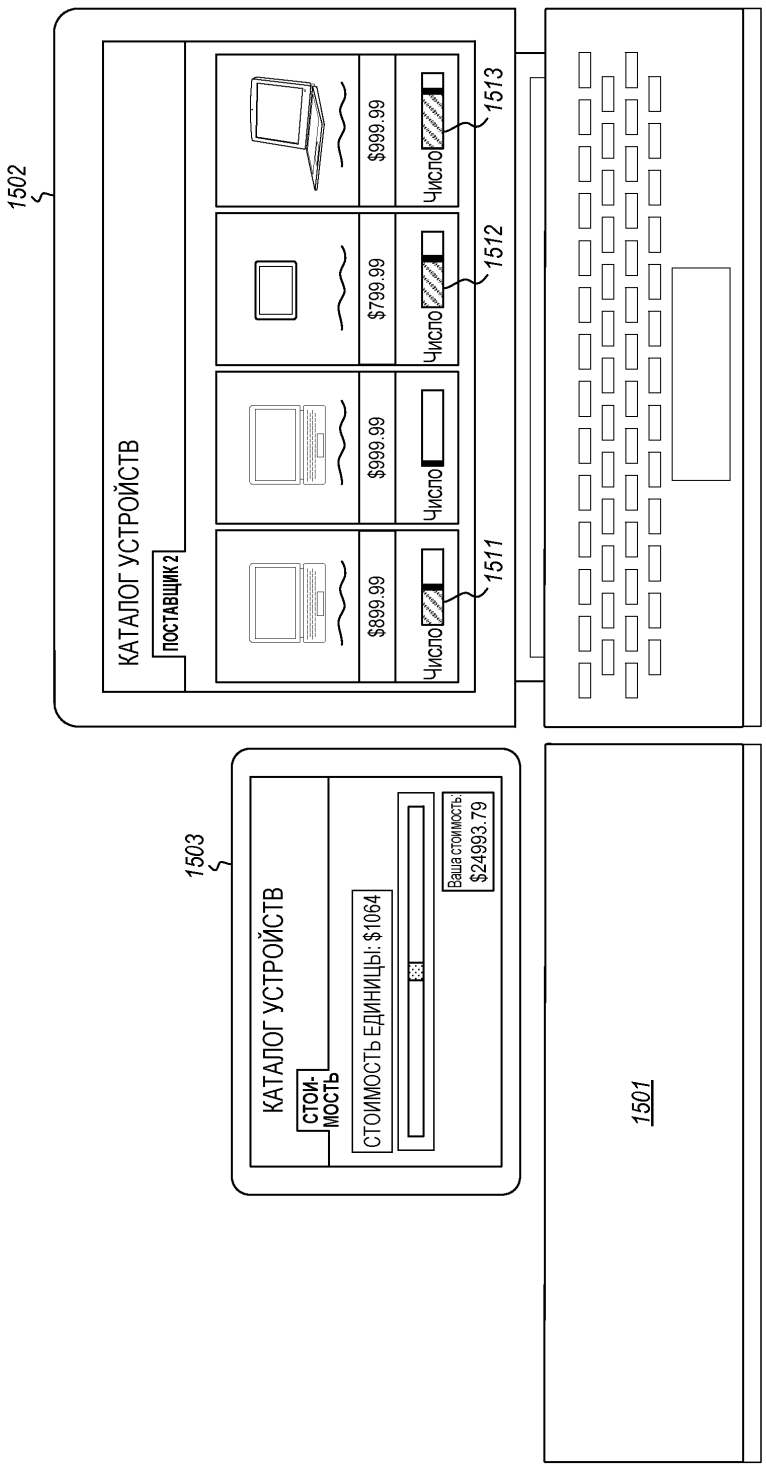
15


1501
КАТАЛОГ УСТРОЙСТВ
ИЗГОТОВИТЕЛИ
ПОСТАВЩИК 1
ПОСТАВЩИК 2
ПОСТАВЩИК 3
1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$899.99
$999.99
$799.99
$999.99
Число
Число
Число
Число
1511
1512

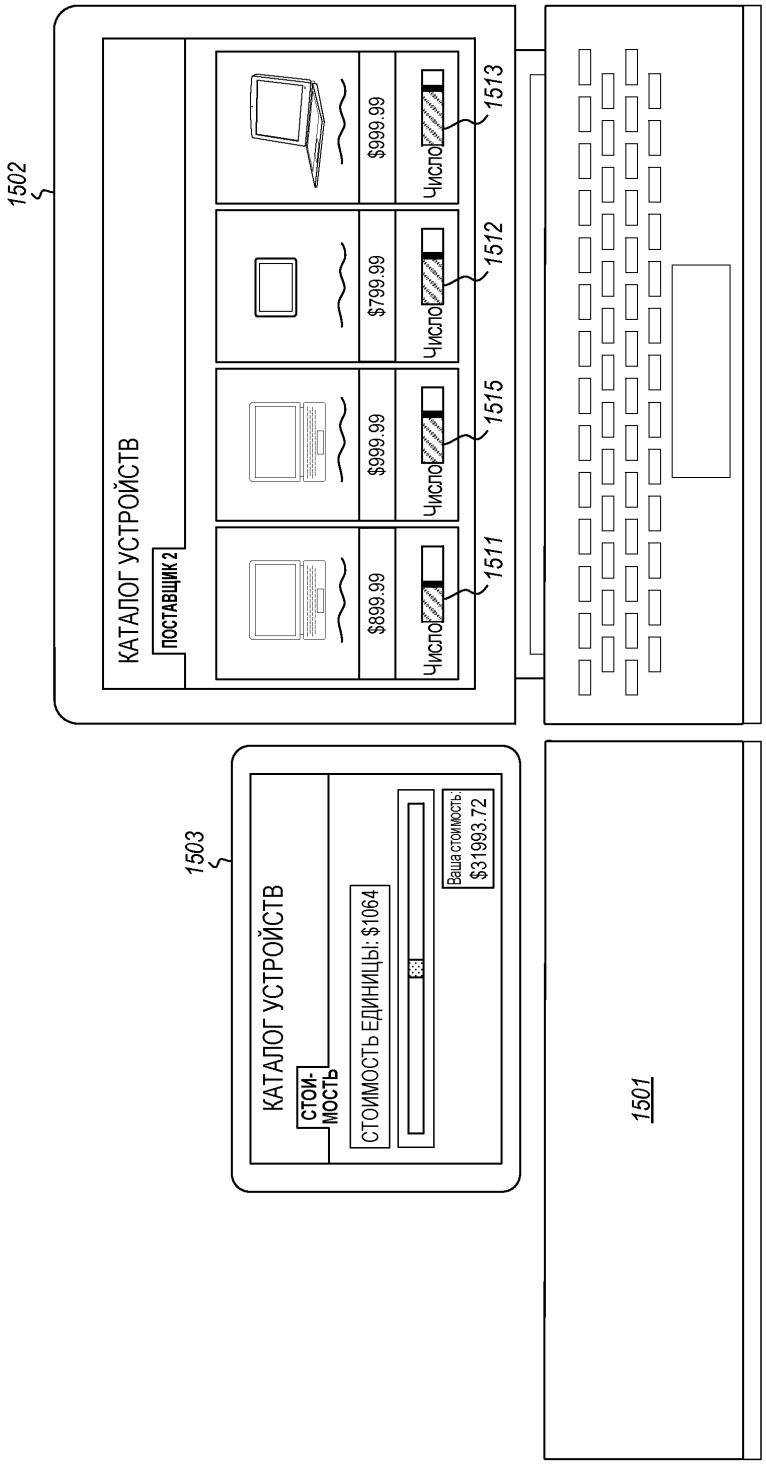
ФИГ. 15D


1501
КАТАЛОГ УСТРОЙСТВ
ИЗГОТОВИТЕЛИ
ПОСТАВЩИК 1
ПОСТАВЩИК 2
ПОСТАВЩИК 3
1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$999.99
Число
$799.99
Число
$999.99
Число
$999
Число
1512
1513
1514

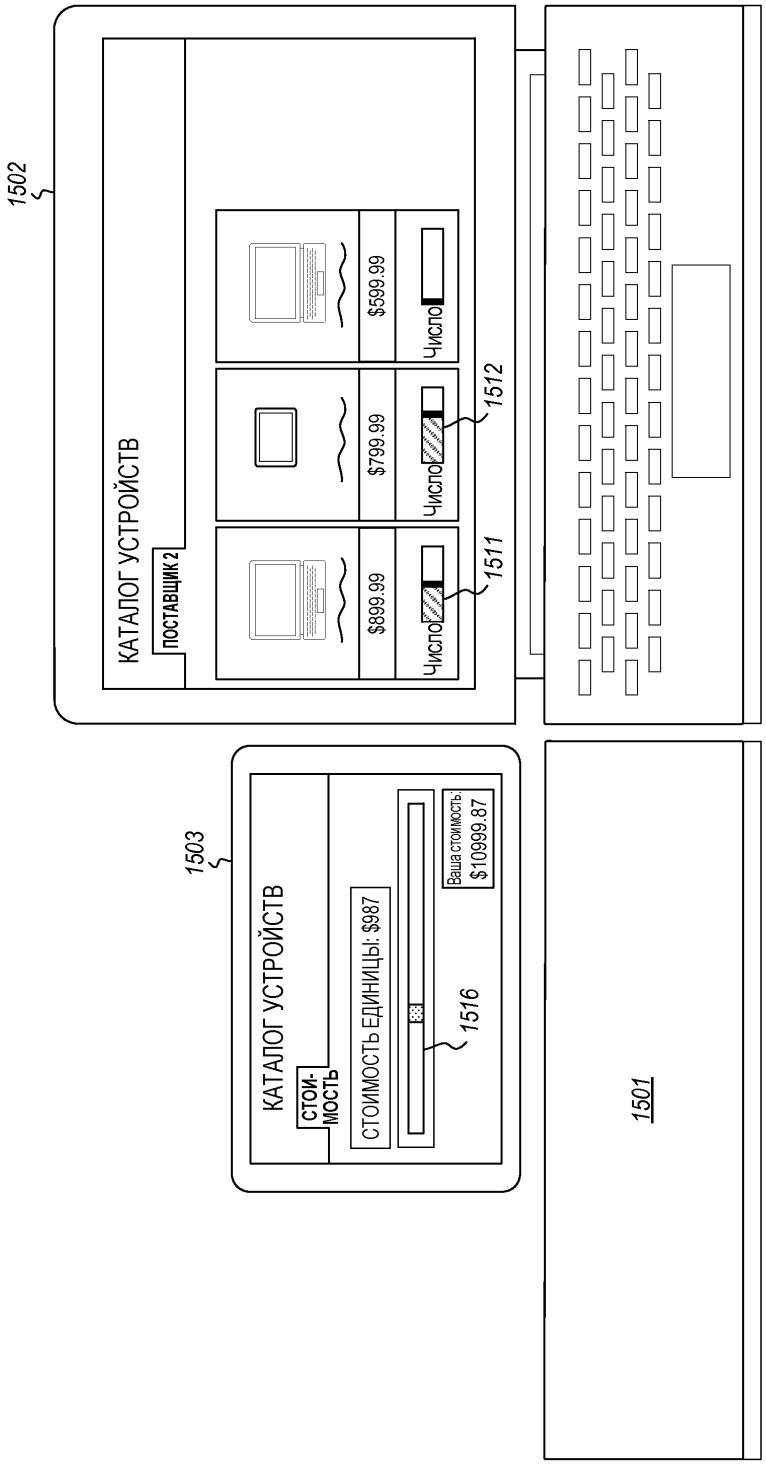
ФИГ. 15E


1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$899.99
$999.99
$799.99
$999.99
Число
Число
Число
Число
1511
1512
1513
1503
КАТАЛОГ УСТРОЙСТВ
СТОИ-МОСТЬ
СТОИМОСТЬ ЕДИНИЦЫ: $1064
Ваша стоимость: $24993.79
1501

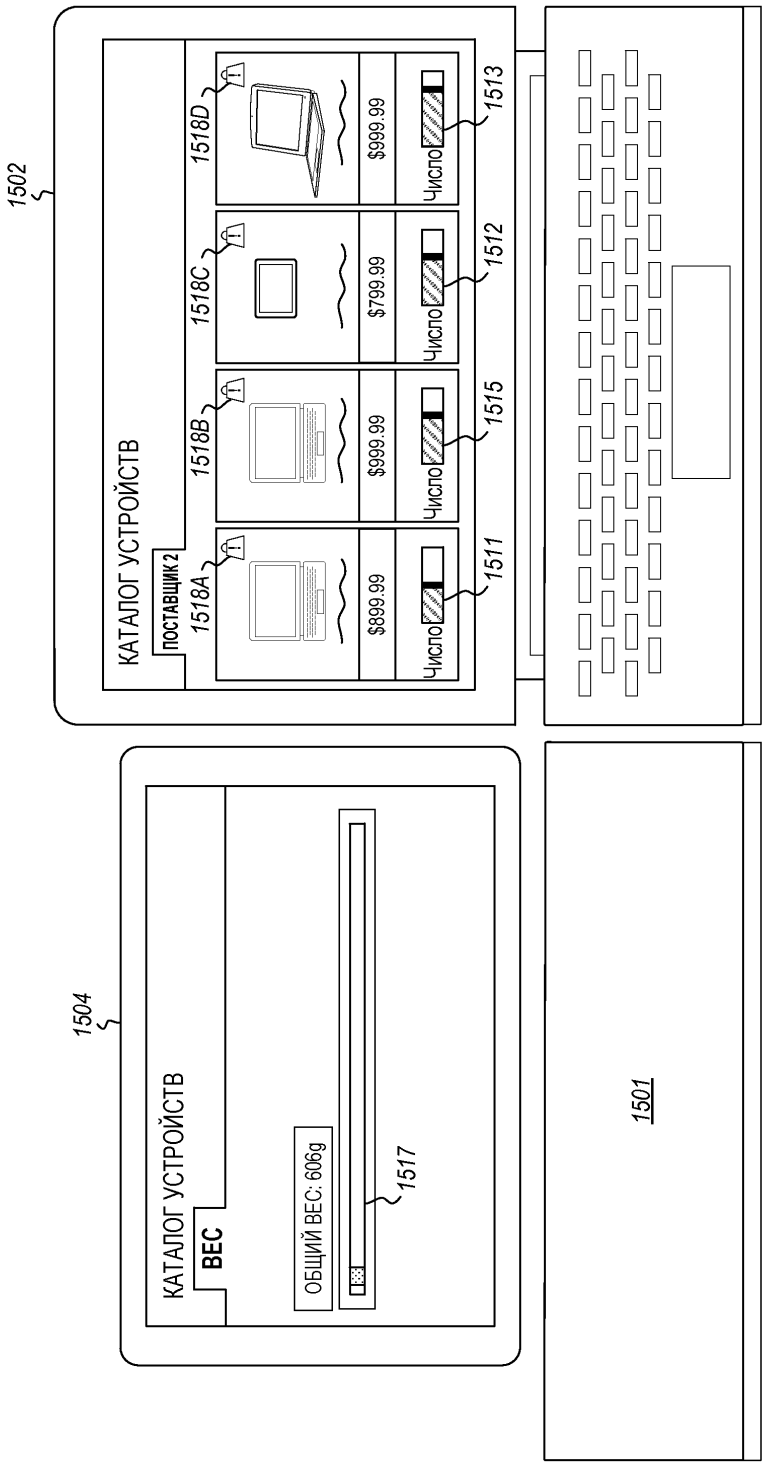
ФИГ. 15F


1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$899.99
$999.99
$799.99
$999.99
Число
Число
Число
Число
1511
1515
1512
1513
1503
КАТАЛОГ УСТРОЙСТВ
СТОИМОСТЬ
СТОИМОСТЬ ЕДИНИЦЫ: $1064
Ваша стоимость: $31993.72
1501

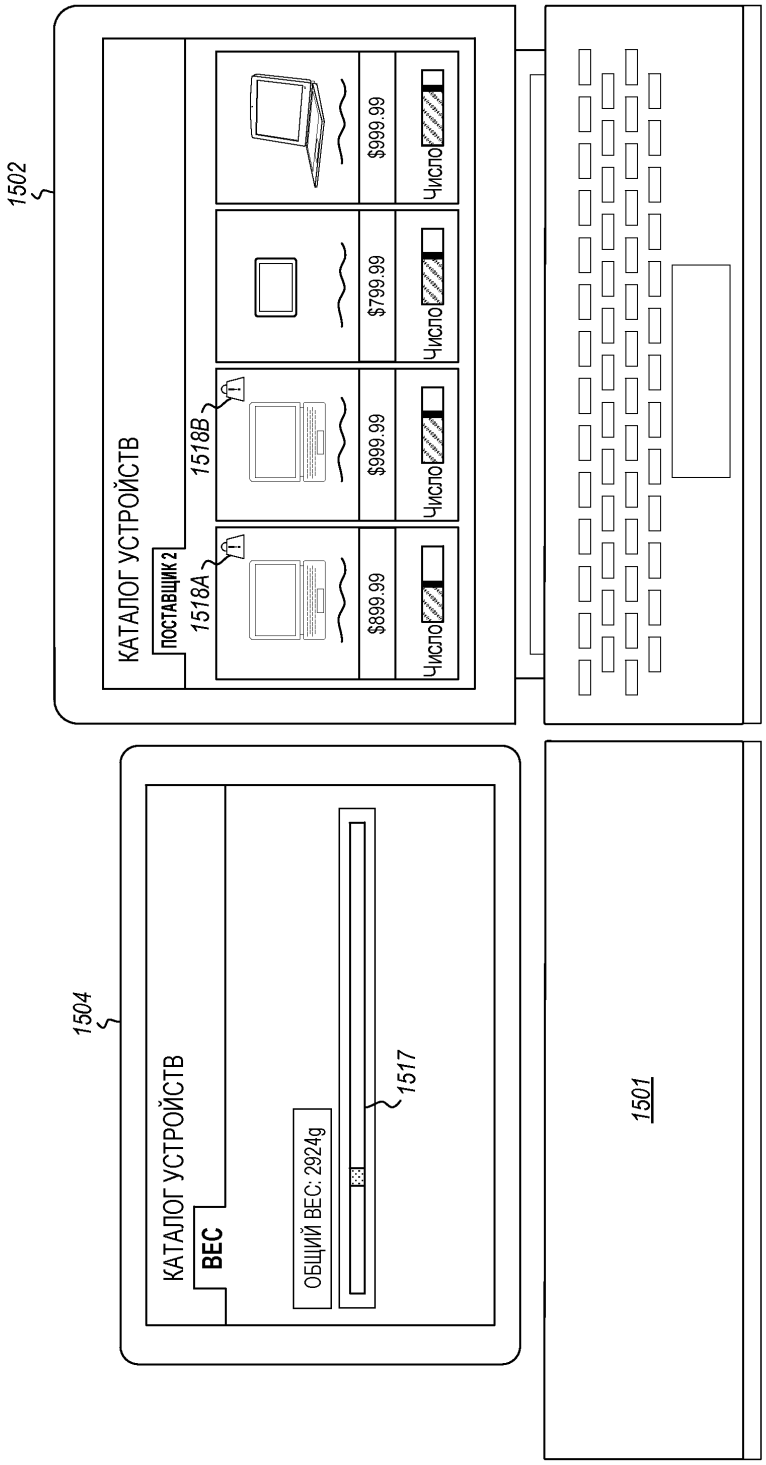
ФИГ. 15G


1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$899.99
$799.99
$599.99
Число
Число
Число
1511
1512
1503
КАТАЛОГ УСТРОЙСТВ
СТОИМОСТЬ
СТОИМОСТЬ ЕДИНИЦЫ: $987
Ваша стоимость: $10999.87
1516
1501

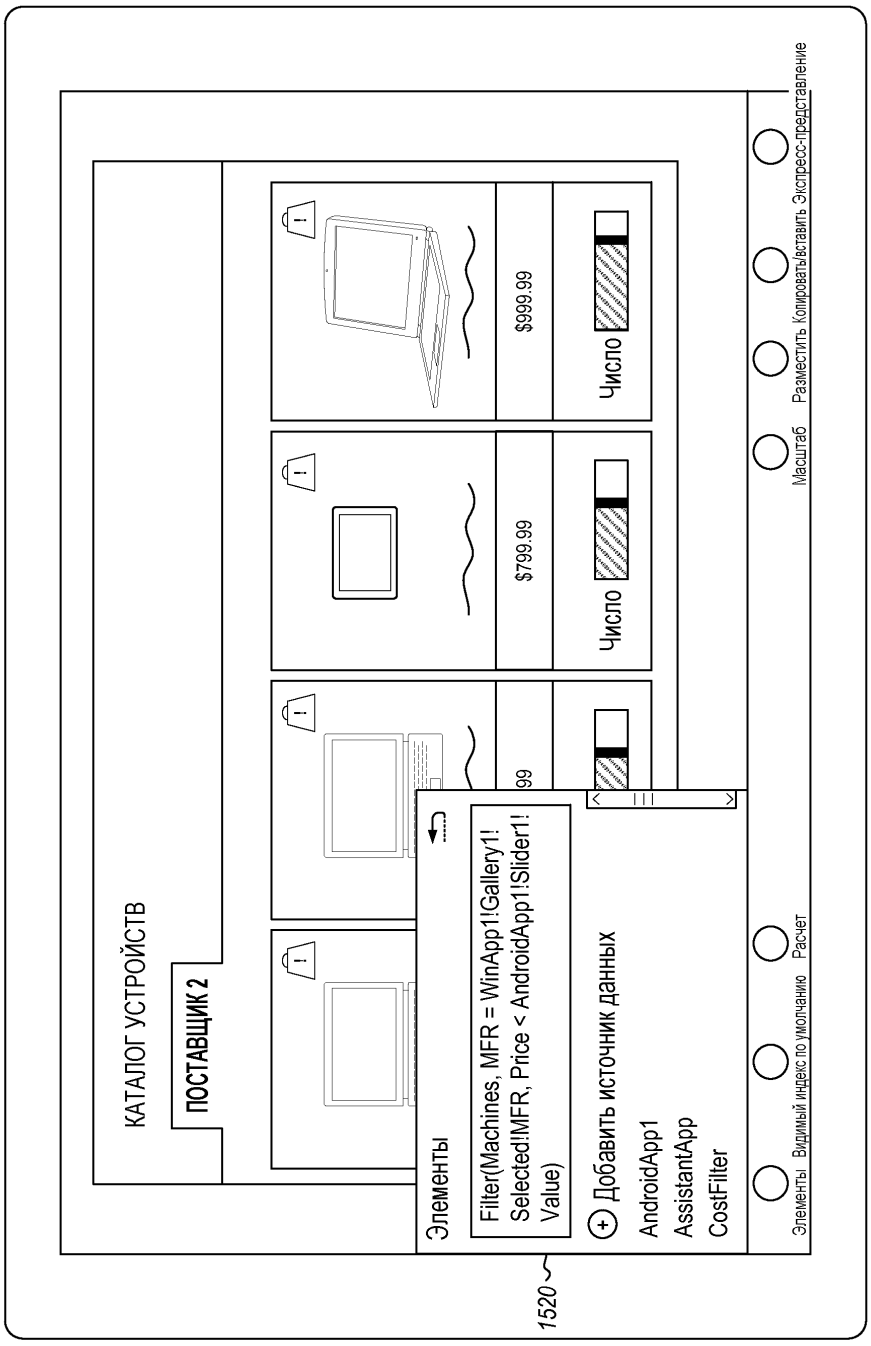
ФИГ. 15H


1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК2
1518A
1518B
1518C
1518D
$899.99
$999.99
$799.99
$999.99
Число
1511
1515
1512
1513
1504
КАТАЛОГ УСТРОЙСТВ
ВЕС
ОБЩИЙ ВЕС: 606g
1517
1501

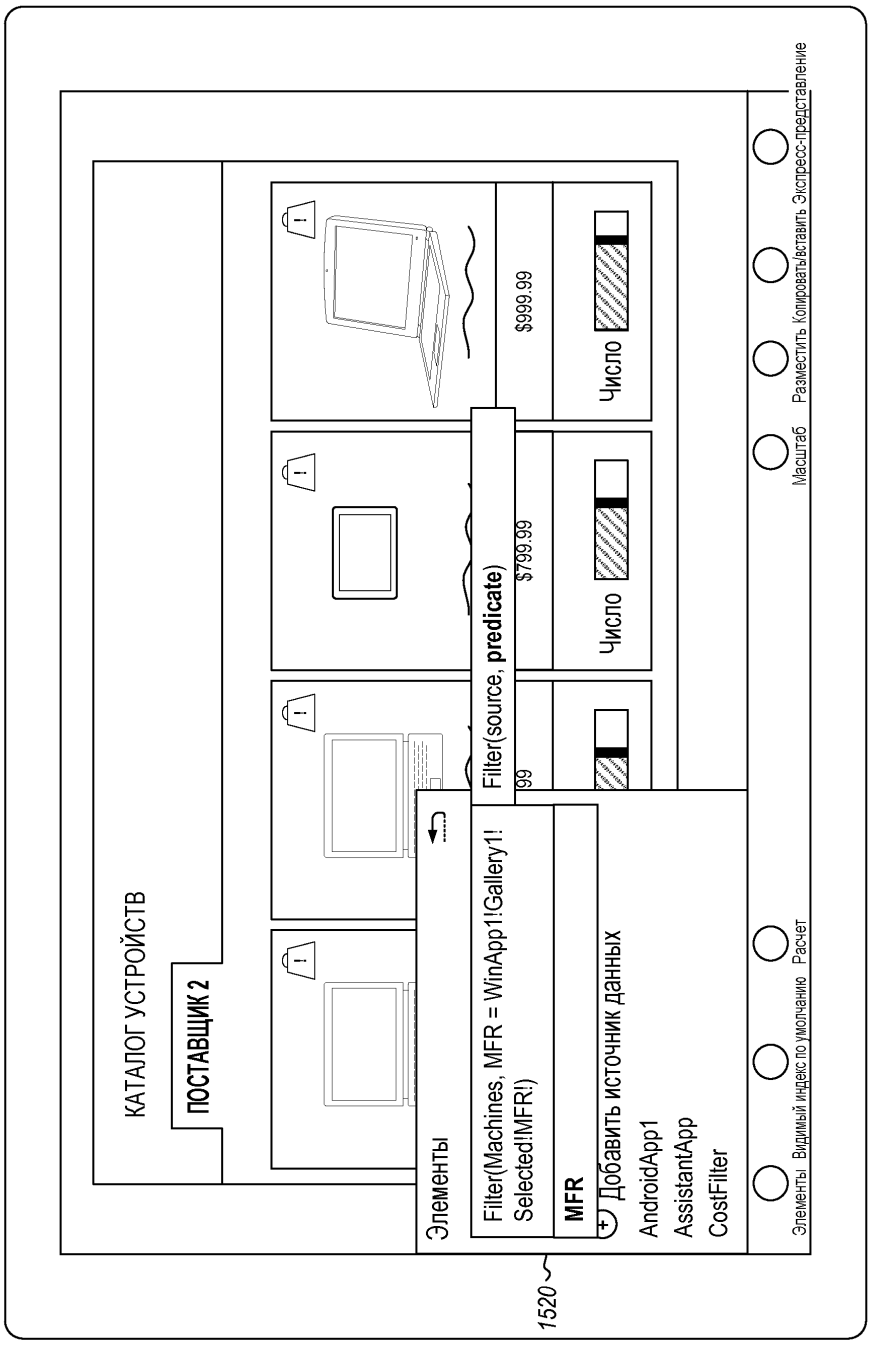
ФИГ. 15I


1502
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК2
1518A
1518B
$899.99
$999.99
$799.99
$999.99
Число
Число
Число
Число
1504
КАТАЛОГ УСТРОЙСТВ
ВЕС
ОБЩИЙ ВЕС: 2924g
1517
1501

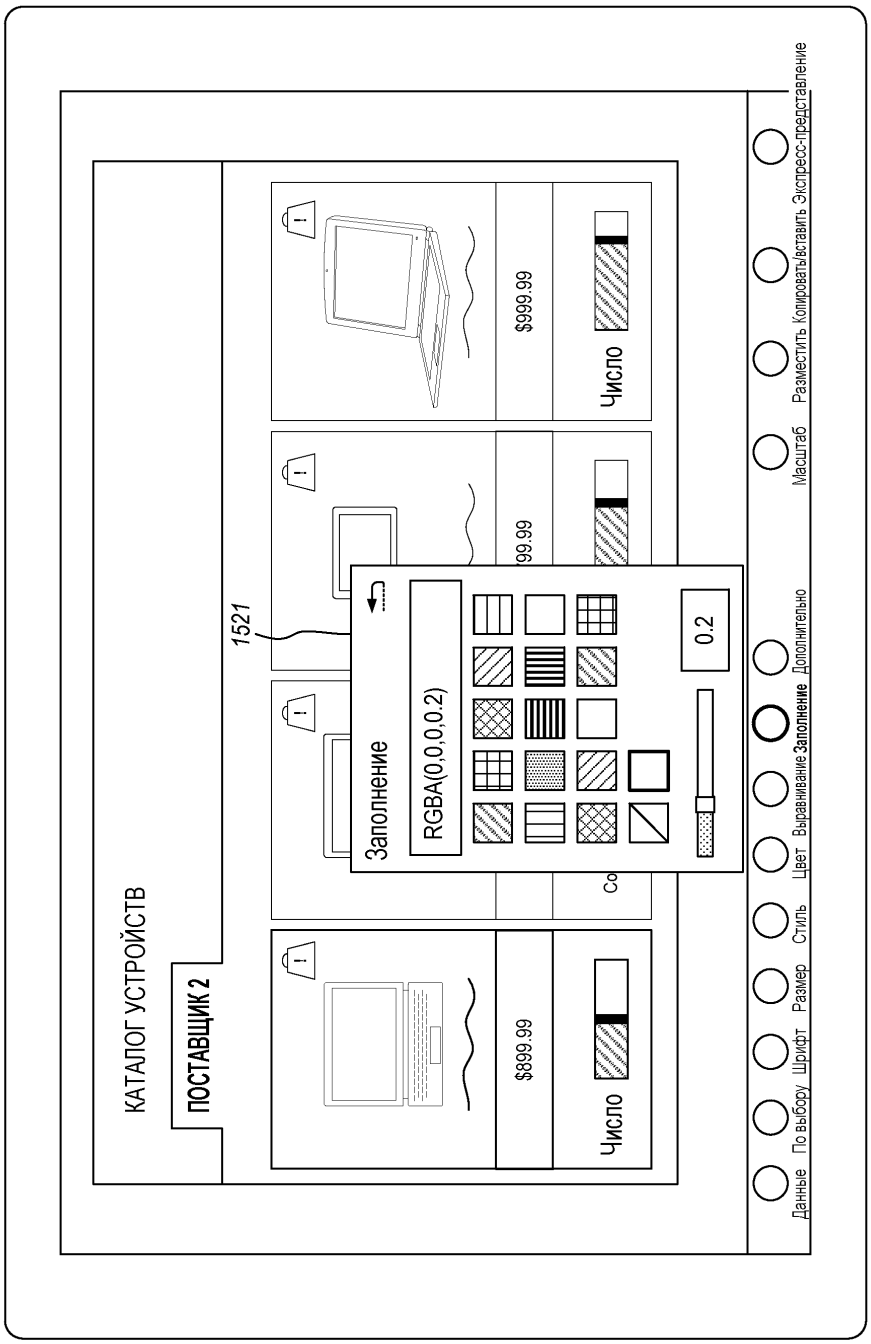
ФИГ. 15J


КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$999.99
Число
$799.99
Число
99
Элементы
Filter(Machines, MFR = WinApp1!Gallery1!
Selected!MFR, Price < AndroidApp1!Slider1!
Value)
Добавить источник данных
AndroidApp1
AssistantApp
CostFilter
1520
Элементы
Видимый индекс по умолчанию
Расчет
Масштаб
Разместить
Копировать/вставить
Экспресс-представление

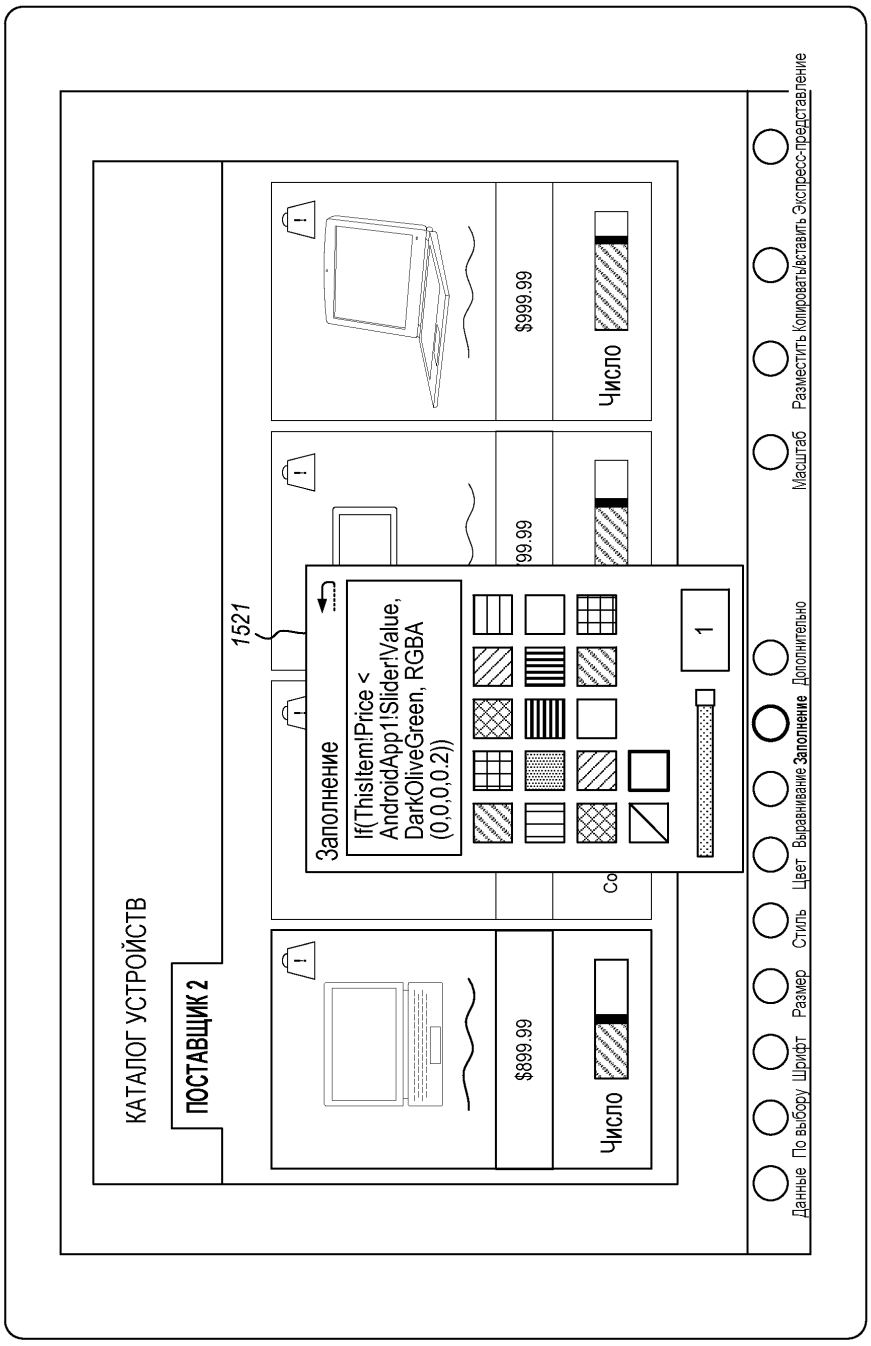
ФИГ. 15K


КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
Элементы
Filter(Machines, MFR = WinApp1!Gallery1!Selected!MFR!)
MFR
Добавить источник данных
AndroidApp1
AssistantApp
CostFilter
Filter(source, predicate)
$999.99
$799.99
99
Число
Число
1520
Элементы
Видимый индекс по умолчанию
Расчет
Масштаб
Разместить
Копировать/вставить
Экспресс-представление

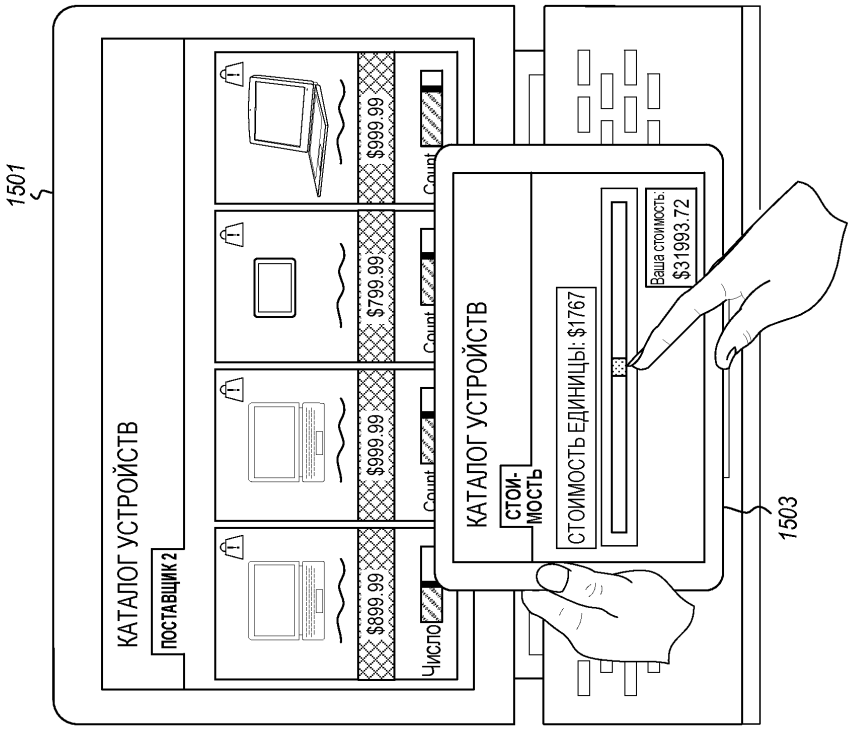
ФИГ. 15L


КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$899.99
Число
$999.99
Число
1521
Заполнение
RGBA(0,0,0,0.2)
0.2
Данные
По выбору
Шрифт
Размер
Стиль
Цвет
Выравнивание
Заполнение
Дополнительно
Масштаб
Разместить
Копировать
Вставить
Экспресс-представление

ФИГ. 15М


КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
1521
Заполнение
If(ThisItem!Price < AndroidApp1!Slider!Value, DarkOliveGreen, RGBA (0,0,0,0.2))
$999.99
Число
$899.99
Число
Данные
По выбору
Шрифт
Размер
Стиль
Цвет
Выравнивание
Заполнение
Дополнительно
Масштаб
Разместить
Копировать/вставить
Экспресс-представление

ФИГ. 15N


1501
КАТАЛОГ УСТРОЙСТВ
ПОСТАВЩИК 2
$899.99
$999.99
$799.99
$999.99
Число
Count
КАТАЛОГ УСТРОЙСТВ
СТОИМОСТЬ
СТОИМОСТЬ ЕДИНИЦЫ: $1767
Ваша стоимость: $31993.72
1503

ФИГ. 15О