



(51) МПК
G06F 21/22 (2006.01)
G06F 15/16 (2006.01)
H04L 29/06 (2006.01)

**ФЕДЕРАЛЬНАЯ СЛУЖБА
 ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
 ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: 2005109315/09, 31.03.2005

(24) Дата начала отсчета срока действия патента:
 31.03.2005

(30) Конвенционный приоритет:
 01.04.2004 US 10/816,319

(43) Дата публикации заявки: 10.10.2006

(45) Опубликовано: 20.01.2010 Бюл. № 2

(56) Список документов, цитированных в отчете о
 поиске: US 2003/0085923 A1, 08.05.2003. US
 2004/0054735 A1, 18.03.2004. RU 99109968 A,
 02.10.1997. US 6301557 B1, 09.10.2001. WO
 03/096187 A2, 20.11.2003.

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

(72) Автор(ы):

**ДЖАМП Дэниел Б. (US),
 МАДДИН Мэтью Д. (US),
 НИЗАР Шахида П. (US)**

(73) Патентообладатель(и):

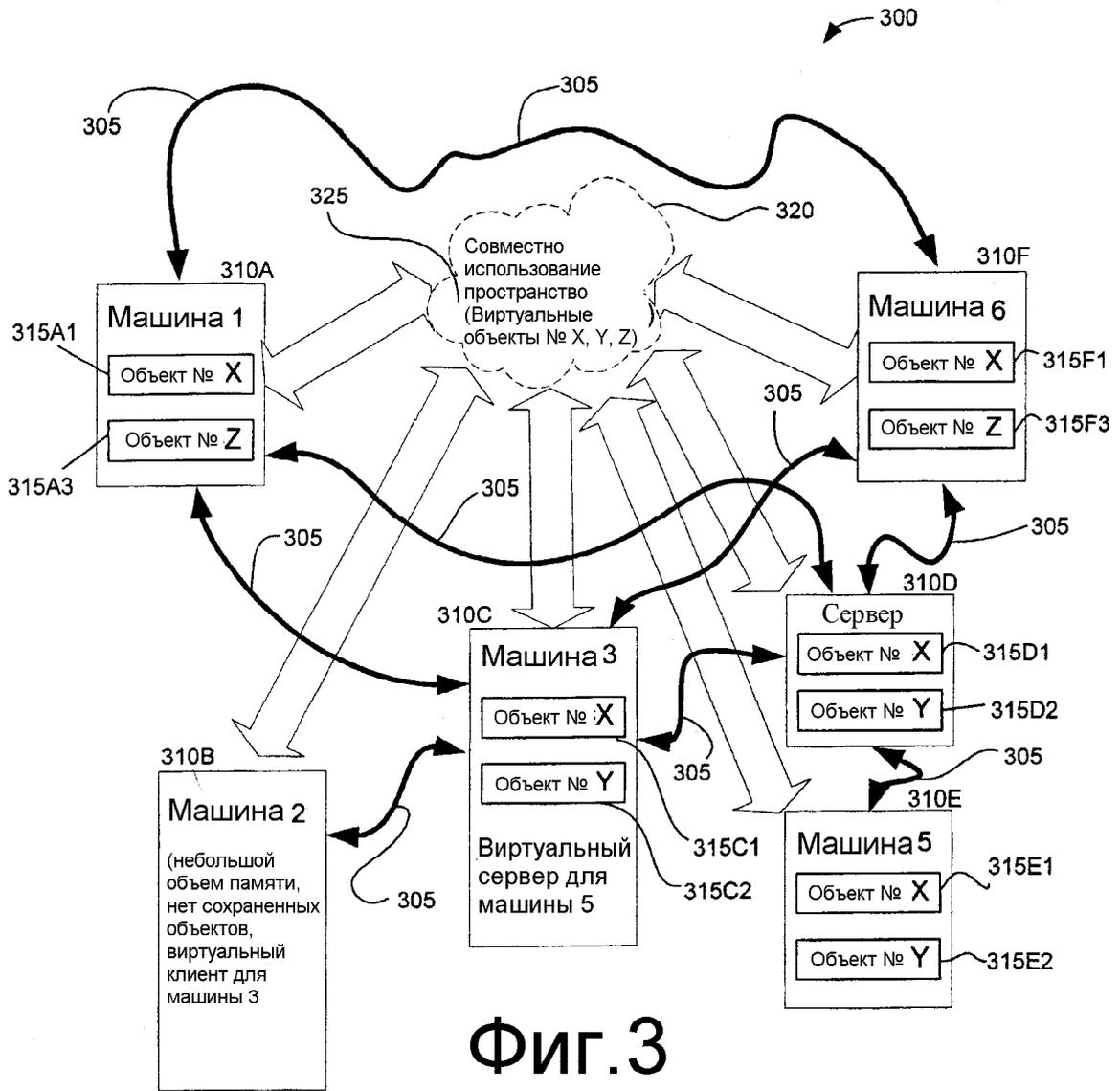
МАЙКРОСОФТ КОРПОРЕЙШН (US)

(54) СИСТЕМА И СПОСОБ ДЛЯ СОВМЕСТНОГО ИСПОЛЬЗОВАНИЯ ОБЪЕКТОВ МЕЖДУ КОМПЬЮТЕРАМИ ПО СЕТИ

(57) Реферат:

Настоящее изобретение относится к способу и системе совместного использования объектов, которые могут находиться в разных машинах. К этим объектам можно осуществить доступ и совместно использовать с помощью компьютерной сети, например, Internet. Объекты могут представлять собой компьютерные объекты программирования, которые могут включать в себя интерфейсы прикладного программирования (API), библиотеки объектов программирования, определения объектов компьютерных

программ и другую подобную информацию для приложений, основывающихся на компьютерной сети. Для способа и системы не требуется компьютерный сервер, так как изобретение может работать скорее как одноранговая или многоточечная компьютерная сеть. Способ и система могут работать как с одноранговыми сетями, так и с сетями "клиент-сервер", не требуя идентификации компьютеров в сети как серверов или компьютеров несерверного типа (клиентовские). 3 н. и 17 з.п. ф-лы, 9 ил.



ФИГ.3



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(51) Int. Cl.
G06F 21/22 (2006.01)
G06F 15/16 (2006.01)
H04L 29/06 (2006.01)

(12) ABSTRACT OF INVENTION

(21), (22) Application: **2005109315/09, 31.03.2005**
(24) Effective date for property rights:
31.03.2005
(30) Priority:
01.04.2004 US 10/816,319
(43) Application published: **10.10.2006**
(45) Date of publication: **20.01.2010 Bull. 2**

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

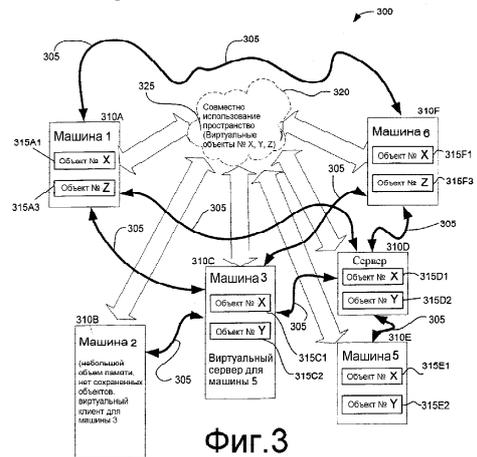
(72) Inventor(s):
**DZhAMP Dehniel B. (US),
MADDIN Meht'ju D. (US),
NIZAR Shakhida P. (US)**
(73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

(54) SYSTEM AND METHOD FOR SHARING OBJECTS BETWEEN COMPUTERS OVER NETWORK

(57) Abstract:
FIELD: physics; computer engineering.
SUBSTANCE: present invention relates to a method and system for sharing objects which can be located in different machines. These objects can be accessed and shared over a computer network, for example the Internet. The objects can be computer programming objects which can include application programming interfaces (API), programming object libraries, determination of computer program objects and other similar information for applications based on a computer network. The method and the system do not require a server computer since the invention can work as a peer-to-peer or a multipoint computer network.
EFFECT: method and system can work with peer-to-peer networks and "client-server" networks

without requirement for identification of computers on the network as servers or non-server computers (clients).

20 cl, 9 dwg



RU 2 379 755 C2

RU 2 379 755 C2

Область техники, к которой относится изобретение

Настоящее изобретение относится к совместному использованию информации между компьютерами по сети, и, более точно, оно относится к одноранговой или многоточечной сети, которая допускает совместное использование компьютерных объектов между компьютерами.

Предшествующий уровень техники

Известные среды совместного использования компьютерных объектов обычно используют архитектуру "клиент-сервер", согласно которой компьютеры-клиенты осуществляют доступ к центральному серверу для запуска программ или извлечения данных. Архитектура "клиент-сервер" является общей формой распределительной вычислительной системы, в которой программное обеспечение поделено между задачами сервера и задачами клиента. Клиент может посылать запросы на сервер согласно некоторому протоколу, запрашивая информацию или действие, и сервер может отреагировать на этот запрос.

Модель "клиент-сервер" аналогична потребителю (клиенту), который отправляет заказ (запрос) согласно форме заказа поставщику (серверу), который посылает товары и счет-фактуру (ответ). Форма заказа и счет-фактура являются частью "протокола", используемого в данном случае для связи.

Обратимся теперь к Фиг.1, которая иллюстрирует известную архитектуру "клиент-сервер", в которой сервер 105 обеспечивает первому, второму и третьему компьютерам-клиентам 110А, 110В и 110С доступ к сетевым объектам 115, расположенным на сервере 105. Согласно этой иллюстративной архитектуре, если первому компьютеру 110А требуется послать видеофайл на второй компьютер 110В этой сети, первый компьютер 110А должен передать файл на сервер 105. Сервер 105, в свою очередь, должен передать видеофайл на второй компьютер 110В. В этом сценарии передачи файла сервер 105 замедляет передачу информации между первым и вторым компьютерами 110А и 110В.

Согласно другому иллюстративному сценарию, когда один из компьютеров 110 желает изменить сетевой объект 115, расположенный на сервере 105, компьютер 110 должен ожидать посылки сервером 105 обратного сообщения на компьютер 110, подтверждающего, что изменение было выполнено. Этот период ожидания может быть назван задержкой. Задержка может причинять неудобства, когда требуются немедленные изменения сетевых объектов 115. Например, задержка может быть очень нежелательна для приложений совместной работы в реальном времени, таких как программы интерактивного общения и презентационные приложения, где необходимы "живые" изменения или распространение информации, которое может включать звуковые и компьютерные данные.

Другая проблема, связанная с известной архитектурой "клиент-сервер", а также с приложением программирования одноранговой связи, состоит в том, что если написаны программы, использующие сетевой объект, то должен быть задан тип компьютерной архитектуры, в которой имеет место связь, относящаяся к этому сетевому объекту. Другими словами, необходимо идентифицировать, будет ли сетевой объект совместно использоваться в рамках архитектуры "клиент-сервер" или в рамках одноранговой сети (не показано на Фиг.1). Это означает, что известная технология должна идентифицировать, какие протоколы связи будут использованы для доступа к сетевому объекту и манипуляции им.

Еще одна проблема, связанная с предшествующим уровнем техники, состоит в том, что сетевой объект может совместно использоваться только в пределах одного типа

компьютерной архитектуры, но не между различными типами сетевых архитектур, поскольку тип протокола связи для конкретной сети должен быть определен и на него должна быть сделана ориентация при написании любого компьютерного кода. Это означает, что программа не может быть написана с известными компьютерными программами, где сетевой объект совместно используется как по одноранговой сети, так и по сети “клиент-сервер”. Известная компьютерная программа должна задавать только один, но не оба типа компьютерных архитектур.

Таким образом, существует необходимость в технике в способе и системе, которые могут совместно использовать сетевые объекты по различным типам компьютерных архитектур, таких как и архитектуры “клиент-сервер”, и архитектуры однорангового типа, без задания протокола связи, необходимого для осуществления связи по различным архитектурам. Далее существует необходимость в технике в значительном уменьшении или устранении задержки, обычно связанной с совместным использованием сетевых объектов между компьютерами, с целью улучшения связи между компьютерами в режиме реального времени. Наконец, существует необходимость в технике, где могут быть написаны компьютерные программы с использованием стандартных объектно-ориентированных методик, независимо от типа компьютерной архитектуры, которая будет вмещать любые сетевые объекты.

Сущность изобретения

Настоящее изобретение содержит способ и систему для совместного использования объектов, которые могут находиться в различных машинах и которые доступны по компьютерной сети, такой как Интернет. Объекты могут содержать компьютерные объекты программирования, которые могут включать в себя, но не в ограничительном смысле интерфейсы прикладного программирования (API), библиотеки объектов программирования, определения объектов компьютерных программ и другую подобную информацию для приложений, основывающихся на компьютерной сети. Система и способ могут позволить написать компьютерные программы, выполняющие запросы к сетевым объектам, находящимся на других компьютерах, подсоединенных к компьютерной сети.

Одно из преимуществ настоящего изобретения состоит в том, что оно не нуждается в компьютерном сервере, так как изобретение может работать в большей степени подобно одноранговой компьютерной сети. Другое преимущество настоящего изобретения состоит в том, что оно может интерпретировать одноранговые компьютеры одинаковым образом, даже если одноранговый компьютер является компьютерным сервером. Другими словами, настоящее изобретение может работать как с одноранговыми сетями, так и с сетями “клиент-сервер”, не требуя идентификации компьютеров как серверов или как компьютеров несерверного типа (клиентов).

Применения этого изобретения могут включать в себя совместную работу в реальном времени (RTC), которая может позволить компьютерам, соединенным через сеть, осуществлять связь и совместно использовать информацию в реальном времени, почти одновременно по сравнению с известными архитектурами “клиент-сервер”. Иллюстративные приложения RTC могут включать в себя, но не в ограничительном смысле, следующие примеры: взаимный обмен информацией через программы интерактивного общения, которые могут находиться на разных компьютерах, совместное использование документов в пределах программ обработки текстов, которые могут находиться на разных компьютерах, совместное использование электронных таблиц в пределах программ работы с электронными таблицами, которые могут находиться на разных компьютерах, совместное использование

презентаций между презентационными приложениями, которые могут находиться на разных компьютерах, совместное использование данных между игровыми программами, которые могут находиться на разных компьютерах, и другие подобные приложения RTC, где выгодно совместно использовать данные между несколькими
5 разными компьютерами в реальном времени.

Чтобы обеспечивать это совместное использование информации среди компьютеров в сети, сетевые объекты или совместно используемые объекты, поддерживающие приложения RTC, обычно находятся на каждом компьютере. Тем не
10 менее, в некоторых ситуациях первый компьютер, который может нуждаться в использовании сетевого объекта, может не иметь копии этого сетевого объекта, и первому компьютеру может быть дан доступ к упомянутому сетевому объекту, который может находиться на втором компьютере. Это совместное использование сетевых объектов может иметь место, если первый компьютер имеет ограниченную
15 память, при этом первый компьютер может быть карманным компьютером, таким, как персональный цифровой ассистент (PDA).

Когда компьютерная программа выполняет вызов к сетевому объекту, могут быть запущены проверки безопасности для того, чтобы определить, разрешен ли
20 компьютеру, выполняющему эту программу, доступ к конкретному сетевому объекту. Например, проверки безопасности могут запускаться локально в пределах компьютера, выполняющего программу, чтобы определить, разрешено ли осуществлять доступ к сетевому объекту, который может также локально храниться в
25 компьютере. Аналогично, проверки безопасности могут быть также запущены по сети для других компьютеров, имеющих копию сетевого объекта для того, чтобы определить, разрешен ли доступ к сетевому объекту компьютеру, выполняющему программу.

Если компьютер, выполняющий вызов к сетевому объекту, имеет разрешение на
30 доступ к сетевому объекту, тогда компьютер может синхронизировать сетевой объект, находящийся в его памяти, со всеми такими же объектами, которые могут находиться на других компьютерах в сети. Таким образом, каждый компьютер, который имеет доступ к конкретному сетевому объекту и копию конкретного сетевого объекта, может получить все из самых последних обновлений, которые могут быть сделаны в
35 отношении сетевого объекта, причем обновления могут иметь место в режиме реального времени.

В случае приложений компьютерного программирования, использующих сетевые объекты, как только создан класс объекта, где класс может включать в себя сетевые
40 сетевые объекты, то могут быть использованы стандартные объектно-ориентированные методики, для работы со всеми объектами класса, включая любые сетевые объекты. Система и способ согласно изобретению могут перехватить такие вызовы методов, выполняемые к сетевым объектам, и обрабатывать вызовы методов согласно сетевым атрибутам сетевых объектов.

Например, вызов к сетевому объекту, который может находиться на компьютере, выполняющем вызов, может не быть исполнен или завершен, пока не будет принята
45 квитанция подтверждения всеми другими компьютерами, которые имеют копию сетевого объекта и от которых требуется предоставить их квитанции подтверждения. Если вызов выполняется к сетевому объекту, имеющему полностью упорядоченный метод, этапы которого должны происходить в специфической последовательности, то от всех других компьютеров, имеющих копию этого сетевого объекта, может потребоваться согласие с последовательностью до того, как вызов будет исполнен или
50

запущен локально на компьютере, выполняющем вызов к сетевому объекту.

Чтобы управлять сетевыми объектами и отслеживать сетевые объекты, система согласно изобретению может использовать контексты. Каждый сетевой объект системы может иметь один ассоциированный контекст. И каждый сетевой объект
5 может быть частью одного контекста. Каждый контекст может содержать информацию, которая прослеживает одну или более из взаимосвязей между множеством системных сетевых объектов и взаимосвязи между компьютерами, которые могут иметь копии объектов и/или доступ к объектам.

10 Каждый контекст может идентифицировать, какие компьютеры имеют один или более конкретных сетевых объектов, и он может помочь управлять сетевыми объектами, являющимися частью конкретного контекста. Контекст может помочь убедиться в том, что каждый компьютер, являющийся частью контекста, имеет одну и ту же версию одного или более сетевых объектов, что могут составлять контекст.

15 Каждый контекст может иметь подписки безопасности, чтобы помочь определить, какие компьютеры могут осуществлять доступ к определенным сетевым объектам в списке контекста и какие компьютеры могут изменять объекты в контексте.

Контекст может иметь различные состояния. Неактивное состояние может
20 означать, что компьютеры с сетевыми объектами, являющимися частью конкретного контекста, могут не использовать эти сетевые объекты. Активное состояние может означать, что компьютеры с сетевыми объектами используют эти сетевые объекты некоторым способом, и связь, относящаяся к сетевым объектам, имеет место между компьютерами, имеющимися в контексте.

25 Система и способ могут также использовать полностью упорядоченные пакеты в течение различной связи между компьютерами в контексте так, чтобы поддерживались согласованность и целостность сетевого объекта.

Перечень фигур чертежей

30 Фиг.1 - блок-диаграмма известной компьютерной архитектуры "клиент-сервер".

Фиг.2 - блок-диаграмма сетевого персонального компьютера, обеспечивающего иллюстративную рабочую среду для настоящего изобретения.

Фиг.3 - функциональная блок-диаграмма, показывающая иллюстративную систему для совместного использования сетевых объектов согласно одному иллюстративному
35 варианту осуществления настоящего изобретения.

Фиг.4А - функциональная блок-диаграмма, показывающая иллюстративную высокоуровневую архитектуру системы для совместного использования объектов между компьютерами через сеть согласно одному иллюстративному варианту
40 осуществления настоящего изобретения.

Фиг.4В - функциональная блок-диаграмма, перечисляющая различные по времени стадии, которые могут иметь место в течение последовательности создания объекта согласно одному иллюстративному варианту осуществления настоящего изобретения.

45 Фиг.5 - функциональная блок-диаграмма, показывающая иллюстративные компоненты, которые могут быть обнаружены в контексте, согласно одному иллюстративному варианту осуществления настоящего изобретения.

Фиг.6А - функциональная блок-диаграмма, иллюстрирующая разнообразные отличающиеся сеансы, которые могут иметь место между компьютерами, имеющими
50 объекты, которые являются частью контекста, согласно одному иллюстративному варианту осуществления настоящего изобретения.

Фиг.6В - функциональная блок-диаграмма, иллюстрирующая детали первого сеанса, являющегося частью первого контекста по Фиг.6А, согласно одному

иллюстративному варианту осуществления настоящего изобретения.

Фиг.6С - функциональная блок-диаграмма, иллюстрирующая детали второго сеанса, являющегося частью второго контекста по Фиг.6А, согласно одному иллюстративному варианту осуществления настоящего изобретения.

5 Фиг.7 - функциональная блок-диаграмма, показывающая иллюстративную архитектуру системного сетевого объекта, согласно одному иллюстративному варианту осуществления настоящего изобретения.

10 Фиг.8 - логическая схема последовательности операций, обобщенно показывающая иллюстративный процесс совместного использования объектов между компьютерами через сеть, где второй компьютер присоединяется к сеансу с первым компьютером, согласно одному иллюстративному варианту осуществления настоящего изобретения.

15 Фиг.9 - иллюстративная функциональная блок-диаграмма, показывающая совместно используемый объект и то, как к этому совместно используемому объекту может быть осуществлен доступ другими узлами компьютерной сети, согласно одному иллюстративному варианту осуществления настоящего изобретения.

Подробное описание образцовых вариантов осуществления.

20 Способ и система могут совместно использовать объекты, которые могут находиться на разных компьютерах. К этим объектам может быть осуществлен доступ, и они могут совместно использоваться с помощью компьютерной сети, такой как Интернет. Объекты могут содержать компьютерные объекты программирования, которые могут включать в себя, но не в ограничительном смысле, интерфейсы прикладного программирования (API), библиотеки объектов программирования, 25 определения объектов компьютерных программ и другую подобную информацию для приложений, основывающихся на компьютерной сети. Способ и система действуют подобно одноранговой или многоточечной компьютерной сети вместо известной архитектуры “клиент-сервер”.

30 Способ и система могут работать как с одноранговыми сетями, так и с сетями “клиент-сервер”, не требуя идентификации компьютеров в сети как серверов или как компьютеров несерверного типа (клиентов). Применения способа и системы могут включать в себя совместную работу в реальном времени (RTC), что может позволить компьютерам, соединенным через сеть, осуществлять связь и совместно использовать 35 информацию в реальном времени и почти одновременно.

Иллюстративная рабочая среда

40 Иллюстративные варианты осуществления настоящего изобретения будут далее описаны со ссылкой на чертежи, на которых одинаковые ссылочные номера представляют одинаковые элементы на нескольких чертежах. Фиг.2 показывает иллюстративную рабочую среду 200 для реализации настоящего изобретения.

45 Иллюстративная рабочая среда 200 включает в себя вычислительное устройство общего назначения в виде известного персонального компьютера 120. Обычно, персональный компьютер 120 включает в себя процессор 121, системную память 122 и системную шину 123, которая присоединяет различные системные компоненты, включая системную память 122, к процессору 121. Системная шина 123 может относиться к любому из нескольких типов структур шин, включая шину памяти или контроллер памяти, периферийную шину и локальную шину, с использованием любой 50 из ряда архитектур шин. Системная память включает в себя постоянное «запоминающее устройство» (ROM, ПЗУ) 124 и оперативное запоминающее устройство (RAM, ОЗУ) 125. Базовая система ввода/вывода (BIOS) 126, содержащая основные процедуры, которые помогают передавать информацию между элементами

в пределах персонального компьютера 120, например, во время пуска, находится в ПЗУ 124.

Персональный компьютер 120 далее включает в себя накопитель 127 на жестких дисках для чтения с жесткого диска и записи на него, не показан; магнитный
5 дисковод 128 для чтения со сменного магнитного диска 129 или записи на него, и оптический дисковод 130 для чтения со сменного оптического диска 131, такого как ПЗУ на компакт диске (CD-ROM) или другой оптический носитель, или записи на него. Накопитель 127 на жестких дисках, магнитный дисковод 128 и оптический
10 дисковод 130 подключены к системной шине 123 посредством интерфейса 132 накопителя на жестких дисках, интерфейсом 133 магнитного дисковода и интерфейсом 134 оптического дисковода, соответственно.

Хотя описанная здесь иллюстративная среда использует жесткий диск 127, сменный магнитный диск 129 и сменный оптический диск 131, специалистами в данной области
15 техники следует понимать, что и другие типы машиночитаемых носителей, которые могут хранить данные и к которым компьютер может осуществить доступ, например, магнитные кассеты, карты флэш-памяти, цифровые видеодиски, картриджи Бернулли, устройства ОЗУ, устройства ПЗУ и им подобные, могут также использоваться в
20 иллюстративной рабочей среде. Накопители и дисководы и связанные с ними машиночитаемые носители обеспечивают энергонезависимое хранение машиновыполняемых инструкций, структур данных, программных модулей и других данных для персонального компьютера 120.

Некоторое количество программных модулей может храниться на жестком
25 диске 127, магнитном диске 129, оптическом диске 131, в ПЗУ 124 или ОЗУ 125, включая операционную систему 135, прикладную программу 410, содержащую сетевой объект 315, включающий в себя программные модули. Программные модули могут включать в себя, не в ограничительном смысле, процедуры, подпрограммы,
30 программы, объекты, компоненты, структуры данных и т.п., которые выполняют конкретные задачи или реализуют определенные абстрактные типы данных.

Пользователь может ввести команды и информацию в персональный компьютер 120 через устройства ввода, например, клавиатуру 140 и координатно-указательное устройство 142. Координатно-указательные устройства
35 могут включать в себя мышшь, шаровой манипулятор и электронное перо, которое может использоваться вместе с электронным планшетом. Другие устройства ввода (не показаны), могут включать в себя микрофон, джойстик, игровую панель, спутниковую антенну, сканер или т.п. Эти и другие устройства ввода часто
40 подсоединены к процессору 122 через интерфейс 146 последовательного порта, который подключен к системной шине 123, но может быть подсоединен посредством других интерфейсов, таких как параллельный порт, игровой порт, универсальная последовательная шина (USB) или т.п. Дисплейное устройство 147 может быть также
45 подключено к системной шине 123 через интерфейс, такой как видеоадаптер 148. В дополнение к монитору, персональные компьютеры обычно включают в себя другие периферийные устройства вывода (не показаны), такие как, например, динамики и принтеры.

Персональный компьютер 120 может работать в сетевом окружении, используя
50 логические соединения с одним или более удаленными компьютерами 149. Удаленным компьютером 149 может быть другой персональный компьютер, сервер, клиент, маршрутизатор, сетевой ПК, одноранговое устройство или другой общий сетевой узел. Хотя удаленный компьютер 149 обычно включает в себя многие или все из

элементов, описанных выше в отношении персонального компьютера 120, только запоминающее устройство 150 показано на фигуре. Логические соединения, изображенные на фигуре, включают в себя локальную сеть (LAN) 151 и глобальную сеть (WAN) 152. Такие сетевые среды обычны в офисах, компьютерных сетях масштаба предприятия, интрасетях и Интернете.

При использовании в сетевой среде LAN персональный компьютер 120 часто подсоединен к локальной сети 151 через сетевой интерфейс или адаптер 153. При использовании в сетевой среде WAN персональный компьютер 120 обычно включает в себя модем 154 или другие средства для установления связи через WAN 152, такую как Интернет. Модем 154, который может быть внутренним или внешним, подключен к системной шине 123 через интерфейс 146 последовательного порта. В сетевой среде программные модули, изображенные относительно персонального компьютера 120, или их части, и могут храниться в удаленном запоминающем устройстве 150. Следует понимать, что показанные сетевые соединения являются иллюстративными и что могут быть использованы другие средства установления линии связи между компьютерами.

Кроме того, специалистам в этой области техники должно быть понятно, что настоящее изобретение может быть реализовано в других конфигурациях компьютерной системы, включая карманные устройства, многопроцессорные системы, основанную на микропроцессорах или программируемую бытовую электронику, сетевые персональные компьютеры, миникомпьютеры, универсальные вычислительные машины (тейнфреймы) и тому подобное. Изобретение может быть также реализовано на практике в распределенных вычислительных средах, где задачи выполняются удаленными устройствами обработки данных, связанными через сеть связи. В распределенной вычислительной среде программные модули могут быть расположены как на локальных, так и на удаленных запоминающих устройствах.

Общий обзор системы

Рассмотрим теперь Фиг.3, где показана иллюстративная система 300 для совместного использования объектов 315 между компьютерами 310 через сеть. Каждый компьютер 310 может быть связан с другим соответствующим компьютером 310 посредством линии 305 связи. Линии 305 связи могут содержать соединение, реализованное через Internet, или прямое соединение, например телефонную линию. Линии 305 связи могут также быть частью транспортного абстрактного уровня (TAL), что будет обсуждаться ниже со ссылкой на Фиг.4.

Каждый компьютер или машина 310 могут хранить один или более объектов 315 локально, например, на запоминающем устройстве, таком как жесткий диск, или в памяти в пределах соответствующего компьютера 310. Каждый компьютер 310 может хранить одни объекты, такие как первый Объект X 315, за исключением пятого компьютера 310В. Кроме того, каждый компьютер 310 может также хранить разные объекты 315, например, второй и третий Объекты Y и Z. Как показано на Фиг. 3, второй и третий Объекты Y и Z не находятся на всех компьютерах 310.

Способ и система 300 настоящего изобретения позволяют компьютерам 310 осуществлять связь друг с другом, причем таким непосредственным и прозрачным образом, что система может образовать воспринимаемое совместно используемое пространство 320, в котором компьютеры совместно используют виртуальные объекты 325. Другими словами, с точки зрения программирования, строки компьютерного кода, которые написаны для доступа к объектам 315 согласно настоящему изобретению, могут воспринимать объекты 315 как виртуальные

объекты 325, которые находятся в совместно используемом пространстве 320, а не в других компьютерах. Компьютерный код, использующий объекты 315 согласно настоящему изобретению, не требует идентификации того, находятся ли эти объекты в одноранговой сети или на сервере. Любой компьютерный код, использующий

5 объекты 315 согласно настоящему изобретению, не нуждается в отнесении себя к механизму связи между компьютерами 310, которые образуют сеть 300.

Как упомянуто выше, настоящее изобретение функционирует с любым типом компьютера 310, независимо от того, является ли компьютер сервером, таким как

10 сервер 310D, либо автономным или самостоятельным компьютером, подобно первому, второму, третьему и пятому компьютерам 310A-C и E-F. Далее, соответствующая изобретению система 300 допускает связь объектов безотносительно того, может ли компьютер 310 хранить объект 315. Например, третий компьютер 310C

15 может функционировать как виртуальный сервер для второго компьютера 310B, поскольку второй компьютер 310B может представлять собой устройство с малым объемом памяти, такое как персональный цифровой ассистент (PDA), который может быть неспособен хранить любой из объектов 315, поддерживаемых сетью 300.

Фиг.3 показывает то, как компьютеры 310 могут быть соединены между собой с помощью их линий 305 связи. Например, третий компьютер 310C может быть связан со вторым компьютером 310B с малым объемом памяти через линию 305 связи. Третий компьютер 310C может быть связан с сервером 310D, пятым

20 компьютером 310E и шестым компьютером 310F через дополнительные линии 305 связи. Второй, третий, пятый компьютеры (310B, 310C и 310E) и сервер 310D могут каждый иметь два одних и тех же объекта, например, локально расположенных Объекта X и Y. Тем временем, третий компьютер 310C может не иметь те же два

25 объекта X, Y 315C1 и 315C2, как шестой компьютер 310F. То есть, третий компьютер 310C может не иметь третий объект Z, размещенный в этом компьютере, тогда как шестой компьютер 310F может иметь третий объект Z, размещенный в этом

30 компьютере.

Хотя каждый объект 315 может не находиться в каждом компьютере 310, компьютеры 310 без конкретного объекта 315 могут осуществлять доступ к нелокальным объектам 315 в других компьютерах 310, если конкретный

35 компьютер 310 имеет разрешение на доступ к этому нелокальному объекту 315. Разрешение на доступ к любому объекту 315 (включая нелокальные объекты 315) может быть поддержано в контексте 500, как будет обсуждаться в дальнейших деталях ниже, со ссылкой на Фиг.5.

Иллюстративная высокоуровневая архитектура системы

Обратимся теперь к Фиг.4, где показана иллюстративная высокоуровневая архитектура 400 системы для совместного использования объектов 315 между

40 компьютерами 310 по компьютерной сети. Каждый компьютер 310 имеет транспортный абстрактный уровень (TAL), который может формировать часть

45 линии 305 связи, обсуждавшейся выше со ссылкой на Фиг.3. С транспортным абстрактным уровнем 305 могут быть связаны системные сетевые объекты (SNO) 315, как показано на Фиг.3.

Системные сетевые объекты (SNO) 315 могут поддерживать программы 410 приложений, которые могут включать в себя, но не в ограничительном смысле, программы совместной работы в реальном времени (RTC) для обмена информацией

50 через программы интерактивного общения, которые могут находиться в разных компьютерах, совместного использования документов в пределах программ

обработки текстов, которые могут находиться в разных компьютерах, совместного использования электронных таблиц в пределах программ работы с электронными таблицами, которые могут находиться в разных компьютерах, совместного использования презентаций между приложениями презентаций, которые могут находиться в разных компьютерах, совместного использования данных между игровыми программами, которые могут находиться в разных компьютерах, и другие подобные приложения RTC там, где выгодно совместно использовать данные в режиме реального времени среди нескольких разных компьютеров.

Транспортный абстрактный уровень 305 может включать в себя набор интерфейсов, которые могут позволить одному компьютеру 310 осуществлять связь с одним или более другими компьютерами 310 в сети. Транспортный абстрактный уровень может использовать, но не в ограничительном смысле, интерфейсы, такие как TCP (протокол управления передачей), Peer net (одноранговая сеть), HTTP (протокол передачи гипертекста) или любую другую подобную технологию связи, позволяющую компьютеру 310 посылать сообщения на другие компьютеры 310 по сети и принимать сообщения от них. Транспортный абстрактный уровень 305 может формировать виртуальную линию связи или сеанс 415 между системными сетевыми объектами 315, находящимися в разных компьютерах 310. Сеанс 415 будет описан ниже.

Иллюстративная схема последовательности операций и способ совместного использования объектов 315 между компьютерами в сети

Процессы и операции, описанные ниже по отношению ко всем логическим схемам последовательностей операций, могут включать в себя выполняемое процессором манипулирование сигналами и сохранение этих сигналов в структурах данных, находящихся на одном или более запоминающих устройствах. В целях этого обсуждения, процесс может быть в целом представлен как последовательность выполняемых компьютером этапов, ведущих к желаемому результату.

Эти этапы обычно требуют физических манипуляций физическими величинами. Обычно, хотя и не обязательно, эти величины приобретают форму электрических, магнитных или оптических сигналов, которые могут быть сохранены, переданы, объединены, сравнены или обработаны иным способом. Традиционно специалисты в данной области техники говорят о представлениях этих сигналов как о битах, байтах, словах, информации, элементах, символах, знаках, числах, точках, данных, входах, объектах, образах, файлах или т.п. Следует иметь в виду, тем не менее, что эти и аналогичные термины ассоциированы с соответствующими физическими величинами для компьютерных операций и что эти термины являются просто традиционными обозначениями, примененными к физическим величинам, существующим в пределах и в течение работы компьютера.

Следует также понять, что манипуляции в пределах компьютера часто называются такими терминами как создание, добавление, вычисление, сравнение, перемещение, прием, определение, идентификация, заполнение, загрузка, исполнение и т.п., что часто связано с ручными операциями, выполняемыми оператором. Операции, описанные здесь, могут быть машинными операциями, выполняемыми в сочетании с разнообразными входными данными, обеспечиваемыми оператором или пользователем, взаимодействующим с компьютером.

Кроме того, следует понять, что программы, процессы, методы и т.п., описанные выше, не связаны с каким-либо конкретным компьютером или устройством и не ограничены им. Напротив, разные типы машин общего назначения могут быть

использованы в следующем процессе в соответствии с описанными здесь идеями.

Настоящее изобретение может содержать компьютерную программу, или аппаратные средства, или их комбинацию, которые воплощают функции, описанные здесь и показанные на приложенных блок-диаграммах. Тем не менее, должно быть очевидно, что может быть много разных путей реализации изобретения в компьютерном программировании или проектировании аппаратных средств, и изобретение не следует толковать как ограниченное каким-либо одним набором инструкций компьютерной программы.

Далее, квалифицированный программист должен быть способен написать такую компьютерную программу или идентифицировать подходящие аппаратные схемы, чтобы без труда реализовать раскрытое изобретение на основе, например, блок-диаграмм и соответствующего описания в тексте заявки. Следовательно, раскрытие конкретного набора инструкций программного кода или детальное описание аппаратных устройств не считается необходимым для адекватного понимания того, как реализовать и использовать изобретение. Соответствующие изобретению функциональные возможности заявленных процессов, выполняемых компьютером, будут объяснены более подробно в последующем описании вместе с остальными Фигурами, иллюстрирующими другие последовательности операций.

Дальнейшие, конкретные шаги в процессах или последовательности операций, описанные ниже на всех логических схемах последовательности операций, должны естественным образом предшествовать другим, чтобы настоящее изобретение функционировало, как описано. Тем не менее, настоящее изобретение не ограничено порядком описанных этапов, если такой порядок или последовательность не меняет функциональных возможностей настоящего изобретения. То есть, признается, что некоторые этапы могут быть выполнены перед, после или параллельно другим этапам, не выходя за рамки объема и сущности настоящего изобретения.

Транспортный Абстрактный Уровень 305

Транспортный Абстрактный Уровень (TAL) 305 может содержать интерфейс, который могут использовать сетевые объекты SNO 315, для осуществления связи с другими машинами 310 по компьютерной сети. Компьютерные сети могут использовать разные стандарты и разные протоколы для обмена разными типами данных. Согласно известной технологии приложения 410, которым необходимо осуществлять связь с другими машинами 310 в сети, часто вынуждены придерживаться разных стандартов/протоколов. Например, некоторые сети используют протокол TCP/IP поверх сети Ethernet для надежной передачи, а другие используют HTTP поверх TCP/IP поверх Ethernet.

Это означает, что приложению 410 обычно требуется осуществлять связь с помощью разных типов протоколов, чтобы передавать их пакеты данных.

Противоположно известной технологии, TAL 305 может абстрагировать любые различия в протоколах, чтобы любой машине 310, использующей SNO 315, не требовалось модифицировать отдельный SNO 315 для работы с другой сетью или в другой сети. Взамен, уровни TAL 305 могут быть поставлены в соответствие их сетям. TAL 305 может абстрагировать сетевые протоколы, используя простые интерфейсы, которые другие разработчики могут модифицировать и обеспечивать реализацию, соответствующую их потребностям. Уровни TAL 305 могут быть непосредственно поставлены в соответствие возможностям сетевых протоколов. Например, каналы 417 (обсуждающиеся ниже) в пределах уровней TAL 305 могут соответствовать портам TCP.

Разработчику, кодирующему сетевое приложение, использующее SNO 315, обычно требуется создать сеанс 415, который устанавливает базовый контакт между разными машинами 310. Как часть установления сеанса 415, приложение 410 может потребовать аутентификацию, например, обмен подписями/мандатами, чтобы установить подлинность. Как только установлен сеанс 415, объекты SNO 315 и данные в пределах объектов SNO 315 могут совместно использоваться. Для каждого SNO 315, подлежащего совместному использованию, создается контекст 500, при этом SNO 315 регистрируется в пределах контекста 500. Дальнейшие детали контекстов 500 будут описаны ниже со ссылкой на Фиг.6.

Теперь обратимся к Фиг.6А-6С, где каждый SNO 315 совместно используется в пределах контекста 500, который имеет один канал 417. Сеанс 415 может быть охарактеризован как большая магистраль, связанная с другими машинами 310, тогда как каналы 417 могут быть охарактеризованы как меньшие подмагистрали, которые могут осуществлять обмен информацией об отдельном SNO 315 между машинами 310. Согласно одному иллюстративному варианту осуществления от каждой машины или узла 310 в сеансе 415 не требуется присоединяться ко всем каналам 417 сеанса 415. Поднабор узлов 310 может участвовать в одном канале 417, а другой или перекрывающийся поднабор может участвовать в другом канале 417.

Когда SNO 315 создается и совместно используется в контексте 500, он может быть нереализован (преобразован в линейную последовательность, подходящую для передачи в удаленное местоположение) в формат транспортируемого по сети сообщения (потенциально двоичный формат или формат XML (расширяемого языка разметки) - см. 1130 на Фиг.9, обсуждаемой ниже) и может быть послан в другие машины 310 в канале 417, используя метод, упоминаемый как IChannel.SendMessage, который будет описан с дополнительными деталями ниже. Когда в SNO 315 сделаны и сохранены изменения, новые изменения могут быть также посланы, используя метод IChannel.SendMessage.

Для доставки сообщений в пределах канала 417 и сеанса 415 в некоторых случаях может быть использовано полное упорядочивание этих сообщений, тогда как в остальных может быть использовано частичное упорядочивание или вообще упорядочивание может не быть использовано. Полное упорядочивание, частичное упорядочивание и отсутствие упорядочивания являются стандартными терминами, используемыми в компьютерных сетевых приложениях и известными обычным специалистам в этой области техники. Но в целях ясности и полноты, ниже приводятся следующие типичные краткие определения/примеры для этих слов: Полное упорядочивание может относиться к сценарию, когда Узел А посылает первый пакет в момент времени T1, а Узел В посылает второй пакет в момент времени T2 (T2 позже чем T1). Все узлы 310 в канале 417 наблюдают, что первый и второй пакеты прибывают точно в том же порядке, как они были посланы (то есть, первый пакет прибывает перед вторым пакетом).

Частичное упорядочивание может относиться к сценарию, где Узел А посылает первый пакет в T1 и второй пакет в T2, а Узел В посылает третий пакет в T3 и четвертый пакет в T4. Разные узлы могут наблюдать первый, второй, третий и четвертый пакеты в разном порядке, но все пакеты, исходящие с одного и того же узла, придут в том порядке, в каком они были посланы. Например, Узел С мог бы принять сначала третий и четвертый пакеты, а потом первый и второй пакеты. Но они не принимают второй пакет перед первым пакетом или четвертый пакет перед третьим пакетом.

Отсутствие упорядочивания является сценарием, где нет гарантии какого-либо порядка. Пакеты могут прибывать в любом порядке из любого источника.

Интерфейс 436 ISession

5 Интерфейс 436 ISession может использоваться приложением 410, чтобы создавать сетевой сеанс 415 связи. Приложение 410 может выбрать установление подлинности и полномочий аутентификации с узлами, с которыми оно находится в сеансе 415. Обмен информацией, относящейся к сеансу, осуществляется с использованием метода SendMessage интерфейса 436 ISession.

10 Интерфейс 436 ISession может также включать в себя механизм для приема уведомлений, когда другой узел осуществляет связь с локальным узлом 310. Каждый узел 310 может зарегистрировать обработчик, который управляет любыми принятыми уведомлениями. Когда сеанс создан, уведомление SessionOpenedEvent посылается в узлы 310 в пределах канала 417. Когда сеанс закрыт, уведомление SessionClosedEvent
15 посылается в узлы 310 в пределах канала 417. Когда сообщение принято, уведомление SessionMessageEvent посылается в остальные узлы 310 в пределах канала 417.

20 Экземпляр многоточечного сеанса 415 содержит методы управления сеансом; например, отправка/прием сообщений, создание новых каналов и мониторинг/поддержание статуса сеанса.

Следующее является иллюстративным псевдокодом для открытого интерфейса 436 ISession. Термин "иницируется", использованный ниже, означает, что сообщение
25 послано во все узлы 310 в конкретном канале 417. Специалисту в этой области техники должно быть понятно, что изменения этого псевдокода могут быть сделаны, не выходя за рамки объема и сущности настоящего изобретения:

```

{
// Иницируется, когда сообщение получено из другого узла.
30 event SessionMessageEvent Message;
// Иницируется, когда сеанс был не в состоянии послать // сообщение.
event SessionMessageErrorEvent MessageError;
// Иницируется, когда сеанс открыт.
event SessionOpenedEvent Opened;
35 // Иницируется, когда сеанс закрыт.
event SessionClosedEvent Closed;
// Получает узел, представленный локальным пользователем в // этом сеансе.
INode MyNode
40 // Получает значение, служащее показателем того, открыт ли // сеанс.
bool IsOpen
// Закрывает сеанс.
void Close();
// Проверяет соединение узла.
45 bool CheckConnection(INode who);
// Создает новый неоткрытый канал.
IChannel CreateChannel(string name);
// Посылает сообщение в отдельный узел в сеансе.
50 IMessage SendMessage(object message, INode to, bool ordered, object cookie);
// Останавливает события сообщений и организует очередь // поступающих
сообщений.
void BeginDeferMessages();

```

// Убирает из очереди все сообщения и возобновляет события // сообщений.

```
void EndDeferMessages();
```

```
}
```

IChannel 433: Методы Управления Каналами

5 Если кратко обратиться к Фиг.4В, то на ней показаны различные по времени стадии, которые могут существовать в течение последовательности создания объекта. После того, как создан сеанс 415, приложение 410 может создать один или более объектов SNO 315 и может дать интерфейс 433 IChannel каждому содержащему
10 объекты SNO 315 контексту 500. Контекст 500 может создать канал 417, используя метод CreateChannel из состава ISession 436.

IChannel 433 содержит методы управления каналами, такие как закрытие, мониторинг статуса канала, отправка/прием сообщений и идентификация узлов. Каждый контекст 500 может присоединять обработчики к событиям, представляемым
15 открытым каналом 417, и этот контекст извещается, когда происходят события. Когда другой узел присоединяется к каналу 417 или оставляет канал 417, инициируются ChannelJoinedEvent и ChannelLeftEvent, соответственно. Когда узел сочтен приемлемым для присоединения к каналу,
20 инициируется ChannelJoinRequestEvent.

Следующее является иллюстративным псевдокодом для открытого интерфейса 433 IChannel. Термин "инициируется", используемый ниже, означает, что сообщение послано во все узлы 310 по конкретному каналу 417. Специалистам в этой области
25 техники должно быть понятно, что изменения в этом псевдокоде могут быть сделаны, не выходя за рамки объема и сущности настоящего изобретения:

общедоступный (public) интерфейс 433 IChannel

```
{
```

// Вызывается, когда канал принимает сообщение.

```
30 event ChannelMessageEvent Message;
```

// Вызывается, когда канал был не в состоянии послать // сообщение.

```
event ChannelMessageErrorEvent MessageError;
```

// Вызывается, когда канал открыт.

```
event ChannelOpenedEvent Opened;
```

35 // Вызывается, когда канал закрыт. Все узлы 310 // отсоединены.

```
event ChannelClosedEvent Closed;
```

// Вызывается, когда локальный узел присоединяется к // каналу.

```
event ChannelJoinedEvent Joined;
```

40 // Вызывается, когда локальный узел оставляет канал.

```
event ChannelLeftEvent Left;
```

// Вызывается, когда узел сочтен приемлемым для // присоединения к каналу.

```
event ChannelJoinRequestEvent JoinRequest;
```

// Вызывается, когда узел присоединяется к каналу.

45

```
event ChannelNodeJoinedEvent NodeJoined;
```

// Вызывается, когда узел оставляет канал.

```
event ChannelNodeLeftEvent NodeLeft;
```

// Вызывается, когда узлы канала не имеют одинаковую версию // объектов SNO.

50 // Возможно, имела место потеря данных, что могло привести // к тому, что объекты SNO в канале стали разными версиями друг // друга.

```
event ChannelSynchronizationFailureEvent SynchronizationFailure;
```

// Вызывается, когда канал восстанавливается после сбоя // синхронизации.

```

event ChannelSynchronizationRecoveryEvent SynchronizationRecovery;
// Получает уникальное имя канала.
string Name
// Получает узел, представленный локальным пользователем в // этом канале.
5 INode MyNode
// Получает узлы, имеющиеся в канале.
INode[] Nodes
// Получает значение, служащее показателем того, открыт ли // канал.
10 bool IsOpen
// Открывает канал.
void Open();
// Отсоединяет все узлы от канала.
void Close(object info);
15 // Присоединяет к каналу удаленных узлов.
void Join(INode from, object info);
// Отсоединяет канал.
void Leave(object info);
20 // Посылает сообщение во все узлы в канале.
IMessage SendMessage(object message, MessageOrderingType orderingType, object cookie);
// Посылает сообщение в отдельный узел в канале.
IMessage SendMessage(object message, INode to, bool ordered, object cookie);
// Останавливает события сообщения и организует очередь // поступающих
25 сообщений.
void BeginDeferMessages();
// Убирает из очереди все сообщения и возобновляет события // сообщений.
void EndDeferMessages();
30 }
Интерфейс INode
Каждая машина 310 в сеансе 415 идентифицируется интерфейсом INode. Этот
интерфейс содержит URI (Универсальный Идентификатор Информационного Ресурса)
и интерфейс IIdentity. IIdentity - стандартный интерфейс, определяемый
35 инфраструктурой.Net.
// Задает стандартный интерфейс для идентификации сетевого // узла.
общедоступный интерфейс INode
{
40 // уникальный URI для узла.
string URI
// Идентификационные данные пользователя узла.
IIdentity Identity
}
45 Типы упорядочивания сообщений:
Следующее перечисление определяет типы упорядочивания сообщений.
Общедоступные перечисления (enum) MessageOrderingType
{
50 // Нет упорядочивания
None,
// Сообщения, посланные из одного узла, принимаются в том // же порядке, в каком
они были посланы.

```

Partial (частичное),

// Сообщения, посланные из всех узлов, принимаются в одном // и том же порядке.

Full (полное),

}

5 IMessage:

Одиночное многоточечное (сетевое) сообщение; содержит свойства для идентификации:

10 // Задает стандартный интерфейс для обмена сообщениями, // общедоступный интерфейс IMessage

{

// Получает, кто послал сообщение.

INode From

15 // Получает содержание сообщения.

object Message

// Получает тип упорядочивания сообщений.

MessageOrderingType Ordering

// Получает необязательный cookie сообщения.

20 object Cookie

}

Краткое изложение TAL 305 и его взаимосвязи с объектами SNO 315

25 Для использования объектов SNO 315 обычно требуются отдельные реализации интерфейсов TAL, поскольку объекты SNO 315 используют TAL 305 для взаимодействия с другими машинами 310 в сети.

30 Чтобы обеспечить реализацию TAL 305, нужна версия реализации сетевого протокола. Если доступной версии нет, то следует рассмотреть следующее: идентифицировать стандартный сетевой протокол, который будет использован для осуществления связи с другими машинами 310. Например, если связь будет осуществляться с Web-сервером, предпочтительным протоколом может быть HTTP. Интерфейсы 305 TAL могут быть затем поставлены в соответствие интерфейсам HTTP. Например, когда TAL 305 создан для сеанса 415, может быть создан соответствующий сеанс HTTP. Когда вызван подметод ISession.SendMessage, это сообщение может быть

35 послано с использованием методов put (помещение) и get (получение) протокола HTTP. TAL 305 обеспечивает уровень абстракции для объектов SNO 315, так что не требуются изменения, если желательно использовать SNO 315 с другим сетевым протоколом. Кроме этого, может быть возможно совместно использовать реализацию TAL другого лица или совместно использовать реализации, если сетевой

40 протокол является подходящим. Иллюстративный контекст 500 для системных сетевых объектов 315

45 Обратимся теперь к Фиг.5, где показаны иллюстративные компоненты, которые могут быть обнаружены в контексте 500. Каждый объект 315 системы может иметь один ассоциированный контекст 500. И каждый объект 315 может быть частью одного контекста 500. Каждый контекст 500 может содержать информацию, которая прослеживает одну или более взаимосвязей между множеством системных сетевых объектов (SNO) 315 и взаимосвязей между компьютерами 310, которые могут иметь

50 копии объектов 315 или доступ к объектам 315, либо и то, и другое. Каждый контекст 500 может содержать свой собственный уникальный идентификатор 505, который может включать в себя число или последовательность букв, либо их комбинацию.

Каждый контекст 500 может также содержать одно или более из следующего:
подписок 510 параметров безопасности и разрешений для каждого компьютера 310,
имеющего объекты 315 в контексте 500; информацию 515 событий, которая может
включать в себя информацию присоединении компьютера 310 к контексту 500 или
оставлении компьютером 310 контекста 500, обозначения объекта, такие как
используемый совместно или неиспользуемый совместно, изменения состояния для
объектов 315 в контексте 500 и изменения состояния для самого контекста 500;
подписок 520 сетевых объектов 315, являющихся частью конкретного контекста 500;
подписок 525 идентификаторов компьютеров для каждого компьютера 310,
являющегося частью контекста 500.

Подписок 510 безопасности может включать в себя набор принципалов
(пользователей или процессов, имеющих учетную запись для контроля доступа),
который может включать в себя список разрешений, идентифицирующих то, какие
компьютеры 310 имеют доступ к объектам 315 в контексте 500, и какие уровни
доступа может иметь компьютер 310. Например, один компьютер 310 может иметь
право только на считывание по отношению к конкретному объекту 315, тогда как
другой компьютер 310 может иметь права как на считывание, так и на запись по
отношению к конкретному объекту 315.

Контекст 500 может идентифицировать, какие компьютеры 310 имеют один или
несколько конкретных объектов 315, и может помочь управлять объектами 315,
являющимися частью конкретного контекста 500. Каждый компьютер 310 имеет
частичную или полную копию контекста 500, поэтому компьютер 310 является его
частью. Контекст 500 может помочь удостовериться в том, что каждый
компьютер 310, являющийся частью контекста 500, имеет ту же самую версию
упомянутых одного или более объектов 315, которые могут составлять контекст 500.
С подписком 510 параметров безопасности контекст 500 может помочь определить,
какие компьютеры 310 могут осуществлять доступ к определенным объектам 315 в
контексте 500 и какие компьютеры 310 могут изменять объекты 315 в контексте 500.

Контекст 500 может иметь активные и неактивные состояния. В активном
состоянии каждый компьютер 310, являющийся частью отдельного контекста 500,
может быть обеспечен информацией об объектах 315. Например, если сделаны
какие-либо изменения в отношении объектов 315, являющихся частью контекста 500,
то эта информация может быть предоставлена тем компьютерам 310, которые имеют
копии представляющих интерес объектов 315. В течение активного состояния
контекста 500, каждый компьютер 310 может обмениваться (получать или передавать)
любой информацией относительно объектов 315, являющихся частью контекста 500.

Иллюстративный сеанс 415, содержащий различные контексты 500

Обратимся теперь к Фиг.6А, где показан сеанс 415, существующий между
компьютерами 1-4. Сеанс 415 содержит уникальный идентификатор, который
используется для машин 310, желающих присоединиться к этому сеансу. Машина 310
может находиться во множестве сеансов в одно и то же время. Сеанс 415 может
содержать множество контекстов 500. Сеанс 415 представляет совокупность
машин 310, которые осуществляют связь друг с другом. Это сообщение может иметь
место по различным контекстам 500 в течение сеанса.

Обратимся теперь к Фиг.6В, где показаны детали первого контекста 500А,
являющегося частью большего сеанса 415 по Фиг.6А. Все компьютеры 310 в
контексте 500А, имеющие разрешение использовать первый и второй объекты 315,
имеют копию или, по меньшей мере, доступ к первому и второму объектам 315

независимо от их местоположения. Для иллюстративного варианта осуществления, изображенного на Фиг. 6А и 6В, только первый, третий и четвертый компьютеры 310 имеют первый и второй объекты 315. Машина 2 не является частью контекста 500А, и, следовательно, машина 2 не может осуществлять доступ или получать копию второго объекта (но машина 2 является частью большего сеанса 415, как проиллюстрировано на фиг.6А).

Обратимся теперь к Фиг.6С, где показаны детали второго контекста 500В, являющегося частью большего сеанса 415 по Фиг.6А. Все компьютеры 310 в контексте 500В, имеющие разрешения использовать первый и третий объекты 315, имеют копию или по меньшей мере доступ к первому и третьему объектам 315 независимо от их местоположения. Для иллюстративного варианта осуществления, показанного на Фиг. 6А и 6С, только первый, второй и четвертый компьютеры 310 имеют первый и третий объекты 315. Машина 3 не является частью контекста 500А, и, следовательно, машина 3 не может осуществлять доступ или получать копию объекта три (но машина 3 является частью большего сеанса 415, как показано на Фиг.6А).

Иллюстративная архитектура системного сетевого объекта 315

Обратимся теперь к Фиг.7, где показана иллюстративная архитектура системного сетевого объекта 315 согласно первому иллюстративному варианту осуществления настоящего изобретения. Системный сетевой объект 315 может содержать совместно используемые методы, которые вызываются в каждом узле в контексте, локальные методы, которые вызываются только на локальной машине, и другой специальный код синхронизации, который является специфическим для совместно используемого сетевого объекта 315.

Создание контекста 500 с системными сетевыми объектами 315

Сначала машина 310, как правило, должна присоединиться к сеансу 415. Это может быть выполнено использованием TAL 305. Набор принципалов может быть создан и добавлен к контексту 500. Они описывают списки безопасности/доступа, поддерживаемые средством управления (менеджером) 705 списков контекста для совместно используемого контекста и реализуются локально. Для системы масштаба предприятия набор принципалов доступа может быть получен из заслуживающего доверия местоположения, например, Active Directory (активного каталога) от Microsoft или заслуживающей доверия машины 310 в сеансе 415. Канал 417, который будет ассоциирован с контекстом, может быть создан с использованием сеанса 415.

Совместное использование контекста 500 по множеству машин 310

Для совместного использования контекста 500 по множеству машин 310, метод, именуемый как «Share («Совместное использование», может быть вызван из Shared objec Services (служб совместно используемых объектов) 439, которому в качестве аргументов могут быть переданы канал 417 и контекст 500. После того, как произойдет эта передача информации, контекст 500 считается «совместно используемым».

Присоединение к контексту 500

Для того, чтобы присоединиться к контексту 500, метод, именуемый как “Join” (“Присоединение”), может быть вызван из служб 439 совместно используемых объектов, принимая в качестве аргументов канал 417, контекст 500 и узел 310, в котором желательно соединение. Обычно узел 310 должен уже существовать в канале 417. И запрос на присоединение обычно отсылается на заданный узел 310. Как только узел 310 принимает это сообщение "OnJoinRequest", которое может содержать полномочия/идентификационные данные присоединяющегося узла и дополнительные

свойства, подлежащие заданию создателем контекста 500, принимающий узел 310 может выбрать принятие или отклонение запроса.

И в любом случае, принимающий узел 310 обычно посылает полностью упорядоченное сообщение в присоединяющийся узел 310, которое может содержать дополнительную информацию, специфическую для реализации этого совместно используемого контекста 500. Это сообщение обычно полностью упорядочено так, чтобы любые данные, посланные в этом сообщении, были синхронизированы с другими сообщениями, посылаемыми в канале 417. Таким образом, присоединяющийся узел 310 может стартовать со своим контекстом 500 в синхронизированном состоянии.

Иллюстративный контекст 500 может использовать эти данные, чтобы посылать существующие совместно используемые объекты 315 в контексте 500 в присоединяющийся узел 310, таким образом сохраняя контекст синхронизированным. Присоединяющийся узел 310 может принять эти данные как сообщение OnJoinResponse.

Если сообщение содержит принятие в контекст 500, присоединяющийся узел 310 теперь считается находящимся в контексте 500.

Контекст 500 может принимать сообщения, показывающие состояние каждого узла 310 в контексте 500. Состояния узлов могут включать в себя, но не в ограничительном смысле, следующее: соединяющийся, соединенный, отсоединяющийся, отсоединенный и сброшенный. Порядок событий для узла 310, присоединяющегося к контексту 500, почти всегда выглядит следующим образом: отсоединенный -> соединяющийся -> соединенный.

Для отсоединения узла 310 от контекста 500 порядок событий идет обычно следующим образом: соединенный -> отсоединяющийся -> отсоединенный. На любой стадии узел 310 может перейти в сброшенное состояние, указывая, что узел 310 неожиданно был сброшен из контекста.

Иллюстративный способ совместного использования объектов между компьютерами - присоединение к контексту 500

Обратимся теперь к Фиг.8, которая является логической схемой последовательности операций, в общем виде показывающей иллюстративный процесс 800 совместного использования объектов 315 между компьютерами 310 по сети, где второй компьютер 310 присоединяется к сеансу 415 с первым компьютером 310, согласно первому иллюстративному варианту осуществления настоящего изобретения. Этап 803 является первым этапом иллюстративного процесса 800, на котором создают контекст 500 с уникальным идентификатором, относящимся по меньшей мере к первому компьютеру 310. На этапе первый компьютер или узел 310 может создать контекст 500, имеющий уникальный буквенный, или числовой, или буквенно-цифровой идентификатор, как обсуждалось выше в отношении Фиг.5. Уникальный идентификатор контекста может быть записан в контекст 500 на этапе 806.

На этапе 809 ветвления определяют, послал ли второй узел 310 запрос на присоединение к контексту 500, созданному первым узлом 310. Если результат этапа 809 ветвления негативен, то процесс следует по ветви "Нет" и возвращается обратно на этап 809. Если результат этапа 809 ветвления положителен, то по ветви "Да" следуют к этапу 812 ветвления.

На этапе 812 ветвления определяют, имеет ли второй узел 310 необходимое разрешение (разрешения) для присоединения к сеансу с первым узлом 310. На этом этапе первый узел 310 может проверить свой список 500 контекста, и более конкретно,

подсписок 510 безопасности контекста 500, чтобы определить, имеет ли первый узел 310 разрешение (разрешения) на доступ к объектам 315 контекста 500.

Если результат этапа 812 ветвления решения негативен, то по ветви "Нет" следуют к этапу 815, где сообщение об отказе передается на второй узел 310. Далее, на этапе 818, процесс 800 заканчивается.

Если результат этапа 812 принятия решения положителен, то по ветви "Да" следуют к этапу 821, где к контексту 500 добавляют уникальный идентификатор второго узла. Затем, на этапе 824, генерируют ответ на сообщение запроса на присоединение и форматируют его как полностью упорядоченный пакет, который включает в себя данные текущего сеанса и список последних номеров пакетов из текущих узлов 310 в сеансе 415.

Список последних номеров пакетов будет использован на этапе 839, который будет обсуждаться ниже с дальнейшими деталями. На этапе 827 ответ, предназначенный второму узлу 310 и включающий в себя последние номера пакетов, записывают в контекст 500. На этапе 836 ответ в виде полностью упорядоченного пакета доставляется во второй узел 310.

На этапе 833 в любые другие узлы в контексте 500 может быть послано сообщение в отношении того в отношении того, что второй узел 310 присоединяется к контексту 500. На этапе 836 ветвления определяют, приняты ли какие-либо пакеты во втором узле 310 в течение контекста 500. Если результат этапа 836 ветвления негативен, то по ветви "Нет" следуют к этапу 848.

Если результат этапа 836 ветвления положителен, то по ветви "Да" следуют к этапу 839 ветвления, где определяют, являются ли принятые пакеты более старыми, чем номера пакетов, заданные в полностью упорядоченном пакете из первого узла 310. Другими словами, на этапе 839 ветвления определяют, принимал ли второй узел 310 пакеты или информацию, которые имели место до присоединения второго узла 310 к контексту 500. Второй узел 310 не должен использовать информацию относительно объектов 315 контекста 500, которая датирована и которая имела место до момента времени, в который второй узел 310 присоединился к сеансу 415 и контексту 500.

Если результат этапа 839 ветвления положителен, означая, что принятые пакеты содержат датированную информацию, то по ветви "Да" следуют к этапу 842, на котором пакеты игнорируются вторым узлом 310. Если результат этапа 839 ветвления негативен, означая, что пакеты содержат новую информацию относительно момента времени, в который второй узел 310 присоединился к контексту 500, то по ветви "Нет" следуют к этапу 845, на котором второй узел 310 обрабатывает принятые пакеты.

Затем на этапе 848 второй узел 310 может послать сообщение во все другие узлы 310 в контексте 500 в отношении того, что он присоединился к контексту 500. Затем на этапе 851 ветвления определяют, оставил ли какой-либо узел 310 сеанс 415 или был ли какой-либо узел 310 сброшен из сеанса 415. Если результат этапа 851 ветвления негативен, то по ветви "Нет" следуют к окончанию процесса 800.

Если результат этапа 851 ветвления положителен, то по ветви "Да" следуют к процедуре 857 и посылают сообщение во все узлы 310 в отношении того, что узел 310 покинул контекст 500. На этапе 860 уникальный идентификатор для узла 310, который выбыл, записывают в контекст 500. На этом процесс 800 заканчивается.

Описание совместно используемого объекта 315S

Обратимся теперь к Фиг.9, где дана иллюстративная функциональная блок-диаграмма, показывающая совместно используемый объект 315S и то, как к

этому объекту 315S может быть осуществлен доступ другими узлами 310 в компьютерной сети. Совместно используемый объект 315S может содержать объект 315, который может быть использован в качестве посредника (то есть, в случае инфраструктуры .net, объект MarshalByRef), и обычно должен объявлять себя как

5 допускающий совместное использование. Согласно иллюстративному способу объявления объектов 315 как допускающих совместное использование можно использовать атрибут или реализовать интерфейс. Совместно используемый объект 315S может содержать набор совместно используемых методов 1105.

10 Совместно используемые методы 1105 могут содержать методы, которые при вызове активируются на всех узлах 310, подсоединенных к этому совместно используемому объекту 315S в контексте 500.

Совместно используемые методы 1105 объявляются с атрибутами, которые описывают, как должен быть вызван или адресован совместно используемый метод.

15 Атрибуты могут быть классифицированы по меньшей мере на три категории: флаги упорядоченности сообщений, флаги активирования сообщения и заведения сообщения. Для дальнейших классификаций атрибутов смотри приведенный ниже раздел подробного описания, озаглавленный "Атрибуты совместно используемых

20 методов."

Атрибуты совместно используемых методов

Флаги "упорядоченность сообщений" могут описывать путь, которым сообщения активации посылаются по совместно используемому контексту 500S в другие узлы 310. Три типа упорядоченности сообщений могут включать в себя, но не в

25 ограничительном смысле: отсутствие упорядоченности, частичную упорядоченность и полную упорядоченность. Эти типы упорядоченности сообщений описаны выше, в связи с описанием TAL 305. Иллюстративный вариант осуществления настоящего изобретения может содержать полное упорядочивание как опцию по умолчанию, так как полное упорядочивание обеспечивает возможность отсутствия потери

30 синхронизации.

Флаги активации совместно используемого метода могут определять, когда должен быть активирован совместно используемый метод 1105. Совместно используемый объект 315S может принимать сообщения активации совместно используемого метода

35 многократно, всякий раз соответственно изменению в стадии упорядочивания сообщений. Упорядочивание сообщений, как изложено в описании TAL 305 выше, обычно имеет место следующим образом: отсутствие упорядоченности -> частичная упорядоченность -> полная упорядоченность -> гарантировано. В любой момент на

40 некоторой стадии сообщение может иметь сбой, изменяя состояния на "failed" ("сбойное").

Совместно используемый метод 1105 может объявлять, в каких состояниях было бы желательно его активировать. Совместно используемый контекст 500S обычно не гарантирует, что он вызовет совместно используемый метод 1105 на какой-либо

45 стадии, кроме конечной стадии, которая будет либо стадией сбоя, либо стадией, заданной флагом порядка метода. В течение активации совместно используемый метод 1105 может определить узел 310, который активирует этот метод, текущую стадию активирования, и какому совместно используемому контексту 500S принадлежит узел 310.

50

Иллюстративный вариант осуществления может включать используемые по умолчанию значения, установленные в "полное упорядочивание" и "гарантировано". Такая установка по умолчанию может сделать совместно используемый объект 315S

таким, как если бы он существовал в локальной машине 310 и не мог бы (выйти из синхронизма) терять свое зеркальное состояние с другими узлами 310. Установка значения в "гарантировано" обеспечивает, что совместно используемый метод 1105 не

будет вызываться, если его нельзя вызвать с каждого узла 310 в контексте 500.
 "Заведение сообщения" является установкой, которая служит показателем заботы о том, доставлено ли сообщение во все машины 310 или нет.

Регистрация и отмена регистрации совместно используемых объектов 315S в контексте 500

Совместно используемый контекст 500S может регистрировать совместно используемые объекты 315S. Когда совместно используемый объект 315S регистрируется, обычно требуется уникальный идентификатор. Этот идентификатор может использоваться для однозначного различения совместно используемого объекта 315 в совместно используемом контексте 500S. В этом сценарии, объект-посредник 315P возвращается, и совместно используемый объект 315 отныне соединен в контексте 500. Объект-посредник 315P обычно должен использоваться клиентами/узлами 310 совместно используемого объекта 315P. Вызовы методов объекта-посредника 315P могут быть упакованы в сообщение активации методов и посланы в совместно используемый контекст 500S. Когда сообщение 1110 активации методов принимается в совместно используемом контексте 500S, в сообщении используется уникальный идентификатор для нахождения совместно используемого объекта 315, метод которого должен быть активирован. Но совместно используемый контекст 500S обычно не активирует совместно используемый метод 1105 немедленно.

Вместо этого, совместно используемый контекст 500S обычно сначала проверяет на этапе 1115 ветвления свой набор принципов в подписке 510 параметров безопасности и разрешений для определения того, имеет ли посылающий узел 310 разрешение активировать совместно используемый метод 1105. Если результат этапа 1115 ветвления отрицательный, то по ветви "Нет" следуют к этапу 1120, где сообщение сбрасывается и событие локально иницируется в совместно используемом контексте 500S. В противном случае, если результат этапа 1115 ветвления положителен, по ветви "Да" следуют к этапу 1125, где общий контекст 500S активирует метод совместно используемого объекта 315S. Иллюстративная система должна использовать принципы .Net в качестве принципов безопасности.

Следует понять, что вышесказанное относится только к иллюстративным вариантам осуществления настоящего изобретения и что многочисленные изменения могут быть сделаны в них без выхода за рамки сущности и объема изобретения, определяемого приложенной формулой изобретения.

Формула изобретения

1. Способ совместного использования системных сетевых объектов между компьютерами, причем каждый системный сетевой объект имеет множество копий, хранящихся на различных компьютерах, включающий в себя этапы, на которых предоставляют совместно используемый контекст, причем этот совместно используемый контекст имеет уникальный идентификатор, некоторое количество ассоциированных с ним системных сетевых объектов, идентификационные данные компьютеров, на которых хранятся копии каждого системного сетевого объекта, и относящиеся к безопасности разрешения, указывающие права доступа компьютеров к контексту и системным сетевым объектам;

принимают от первого компьютера запрос на присоединение к совместно

используемому контексту, идентифицируемому упомянутым уникальным идентификатором;

5 в ответ на прием от первого компьютера запроса на присоединение к совместно используемому контексту определяют, имеет ли первый компьютер разрешение на присоединение к совместно используемому контексту, и если первый компьютер имеет разрешение на присоединение к совместно используемому контексту, разрешают

10 первому компьютеру присоединиться к совместно используемому контексту; и после того, как первый компьютер присоединился к совместно используемому

принимают от первого компьютера запрос на модификацию первого системного сетевого объекта, ассоциированного с совместно используемым контекстом,

в ответ на прием от первого компьютера запроса на модификацию первого

15 системного сетевого объекта, ассоциированного с совместно используемым контекстом, определяют, имеет ли первый компьютер разрешение на модификацию первого системного сетевого объекта, и если первый компьютер имеет разрешение на модификацию первого системного сетевого объекта, разрешают первому компьютеру модифицировать первый системный сетевой объект,

20 когда первому компьютеру разрешено модифицировать первый системный сетевой объект, определяют, доступен ли каждый компьютер, на котором хранится копия первого системного сетевого объекта, как указывается совместно используемым контекстом, для модификации его копии, при этом совместно используемый контекст указывает, что по меньшей мере три компьютера хранят копию первого системного

25 сетевого объекта, когда определено, что каждый компьютер, на котором хранится копия первого системного сетевого объекта, доступен для модификации его копии, синхронизируют первый системный сетевой объект посредством уведомления каждого компьютера, на котором хранится копия первого системного сетевого объекта, об упомянутой модификации, с тем, чтобы каждый компьютер мог модифицировать свою копию первого системного сетевого объекта, и

30 когда по меньшей мере один компьютер, на котором хранится копия первого системного сетевого объекта, не доступен для модификации его копии, не модифицируют никакие копии первого системного сетевого объекта.

2. Способ по п.1, в котором запрос на модификацию первого системного сетевого объекта включает в себя запрос на вызов метода первого системного сетевого объекта.

3. Способ по п.1, в котором на первом компьютере хранится по меньшей мере одна

40 копия первого системного сетевого объекта. 4. Способ по п.1, в котором на первом компьютере не хранится копия первого системного сетевого объекта.

5. Способ по п.1, дополнительно содержащий этап, на котором уведомляют компьютеры, ассоциированные с совместно используемым контекстом, когда первый

45 компьютер присоединяется к совместно используемому контексту. 6. Машиночитаемый носитель, содержащий инструкции для совместного использования системных сетевых объектов между компьютерами, причем каждый системный сетевой объект имеет множество копий, хранящихся на различных

50 компьютерах, при этом упомянутыми инструкциями реализуется способ, включающий в себя этапы, на которых предоставляют совместно используемый контекст, причем этот совместно используемый контекст имеет некоторое количество ассоциированных с ним

системных сетевых объектов;

принимают от первого компьютера запрос на присоединение к совместно используемому контексту;

5 определяют, имеет ли первый компьютер разрешение на присоединение к совместно используемому контексту, и если первый компьютер имеет разрешение на присоединение к совместно используемому контексту, разрешают первому компьютеру присоединиться к совместно используемому контексту; и

10 после того, как первый компьютер присоединился к совместно используемому контексту,

принимают от первого компьютера запрос на модификацию первого системного сетевого объекта, ассоциированного с совместно используемым контекстом,

15 определяют, имеет ли первый компьютер разрешение на модификацию первого системного сетевого объекта, и если первый компьютер имеет разрешение на модификацию первого системного сетевого объекта, разрешают первому компьютеру модифицировать первый системный сетевой объект,

20 когда первому компьютеру разрешено модифицировать первый системный сетевой объект, определяют, доступен ли каждый компьютер, на котором хранится копия первого системного сетевого объекта, для модификации его копии,

25 когда определено, что каждый компьютер, на котором хранится копия первого системного сетевого объекта, доступен для модификации его копии, синхронизируют первый системный сетевой объект посредством уведомления каждого компьютера, на котором хранится копия первого системного сетевого объекта, об упомянутой модификации, с тем, чтобы каждый компьютер мог модифицировать свою копию первого системного сетевого объекта, при этом по меньшей мере три компьютера хранят копию первого системного сетевого объекта.

30 7. Машиночитаемый носитель по п.6, в котором совместно используемый контекст дополнительно содержит идентификационные данные компьютеров, на которых хранятся копии каждого системного сетевого объекта, и относящиеся к безопасности разрешения, указывающие права доступа компьютеров к контексту и системным сетевым объектам.

35 8. Машиночитаемый носитель по п.6, в котором запрос на модификацию первого системного сетевого объекта включает в себя запрос на вызов метода первого системного сетевого объекта.

9. Машиночитаемый носитель по п.6, в котором на первом компьютере хранится по меньшей мере одна копия первого системного сетевого объекта.

40 10. Машиночитаемый носитель по п.6, в котором на первом компьютере не хранится копия первого системного сетевого объекта.

45 11. Машиночитаемый носитель по п.6, в котором способ дополнительно содержит этап, на котором уведомляют компьютеры, ассоциированные с совместно используемым контекстом, когда первый компьютер присоединяется к совместно используемому контексту.

50 12. Машиночитаемый носитель по п.6, в котором, когда по меньшей мере один компьютер, на котором хранится копия первого системного сетевого объекта, не доступен для модификации его копии, никакие копии первого системного сетевого объекта не модифицируются.

13. Компьютерная система для совместного использования системных сетевых объектов между компьютерами, причем каждый системный сетевой объект имеет множество копий, хранящихся на различных компьютерах, содержащая

процессор;

компонент, который предоставляет совместно используемый контекст, причем этот совместно используемый контекст имеет некоторое количество ассоциированных с ним системных сетевых объектов;

5 компонент, который принимает от первого компьютера запрос на присоединение к совместно используемому контексту;

компонент, который определяет, имеет ли первый компьютер разрешение на присоединение к совместно используемому контексту, и если первый компьютер имеет разрешение на присоединение к совместно используемому контексту, разрешает первому компьютеру присоединиться к совместно используемому контексту; и

компонент, который, после того, как первый компьютер присоединился к совместно используемому контексту,

15 принимает от первого компьютера запрос на доступ к первому системному сетевому объекту, ассоциированному с совместно используемым контекстом, при этом первый компьютер и по меньшей мере два других компьютера хранят копию первого системного сетевого объекта, и

определяет, имеет ли первый компьютер разрешение на доступ к первому системному сетевому объекту и, если первый компьютер имеет разрешение на доступ к первому системному сетевому объекту, разрешает первому компьютеру осуществить доступ к первому системному сетевому объекту; и

компонент, который, когда определено, что каждый компьютер, на котором хранится копия первого системного сетевого объекта, доступен для модификации его копии, синхронизирует копии первого системного сетевого объекта посредством отправки уведомления на каждый компьютер, на котором хранится копия первого системного сетевого объекта, с тем, чтобы первый системный сетевой объект был модифицирован только тогда, когда каждый компьютер, на котором хранится копия первого системного сетевого объекта, является доступным для модификации его копии.

14. Компьютерная система по п.13, в которой совместно используемый контекст дополнительно содержит идентификационные данные компьютеров, на которых хранятся копии каждого системного сетевого объекта, и относящиеся к безопасности разрешения, указывающие права доступа компьютеров к контексту и системным сетевым объектам.

15. Компьютерная система по п.13, в которой запрос на доступ к первому системному сетевому объекту включает в себя запрос на вызов метода первого системного сетевого объекта.

16. Компьютерная система по п.13, дополнительно содержащая компонент, который ассоциирует по меньшей мере один системный сетевой объект, хранящийся на первом компьютере, с совместно используемым контекстом, когда первый компьютер присоединяется к совместно используемому контексту.

17. Компьютерная система по п.13, в которой на первом компьютере не хранится копия первого системного сетевого объекта.

18. Компьютерная система по п.13, дополнительно содержащая компонент, который уведомляет компьютеры, ассоциированные с совместно используемым контекстом, когда первый компьютер присоединяется к совместно используемому контексту.

19. Компьютерная система по п.13, в которой запрос на доступ к первому системному сетевому объекту включает в себя запрос на модификацию первого

системного сетевого объекта.

20. Компьютерная система по п.13, в которой компонент, который осуществляет синхронизацию, содержит

5 компонент, который, после того, как первому компьютеру разрешено модифицировать первый системный сетевой объект, определяет, доступен ли каждый компьютер, на котором хранится копия первого системного сетевого объекта, для модификации его копии, и

10 компонент, который, когда определено, что каждый компьютер, на котором хранится копия первого системного сетевого объекта, доступен для модификации его копии, синхронизирует первый системный сетевой объект посредством уведомления каждого компьютера, на котором хранится копия первого системного сетевого объекта, об упомянутой модификации, с тем, чтобы каждый компьютер мог модифицировать свою копию первого системного сетевого объекта, и, когда по
15 меньшей мере один компьютер, на котором хранится копия первого системного сетевого объекта, не доступен для модификации его копии, не модифицирует никакие копии первого системного сетевого объекта.

20

25

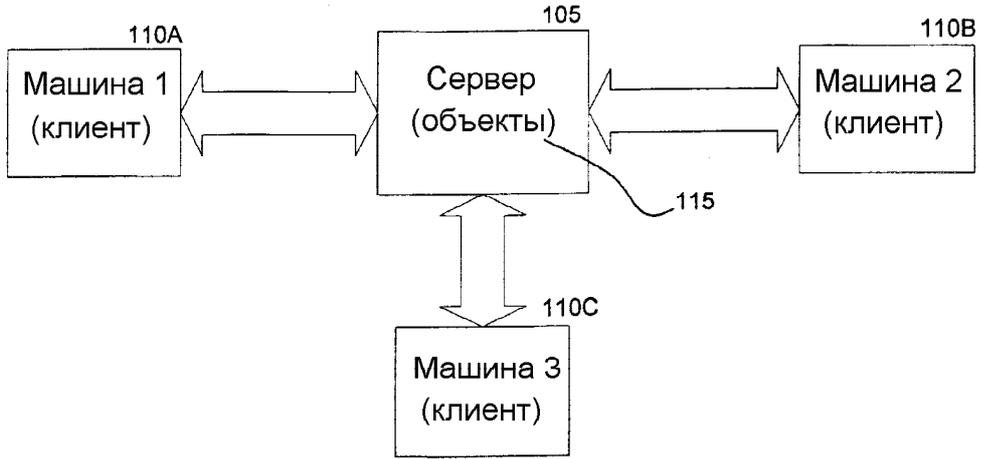
30

35

40

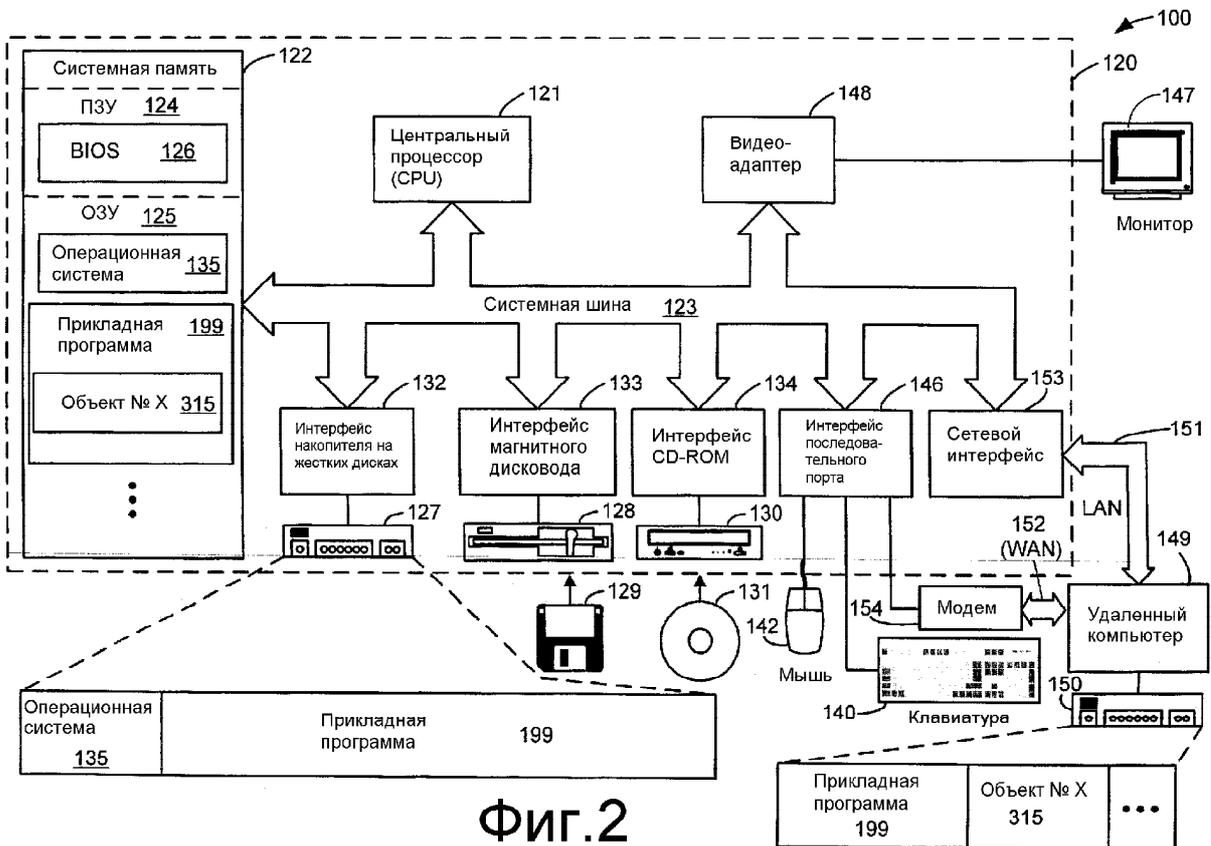
45

50

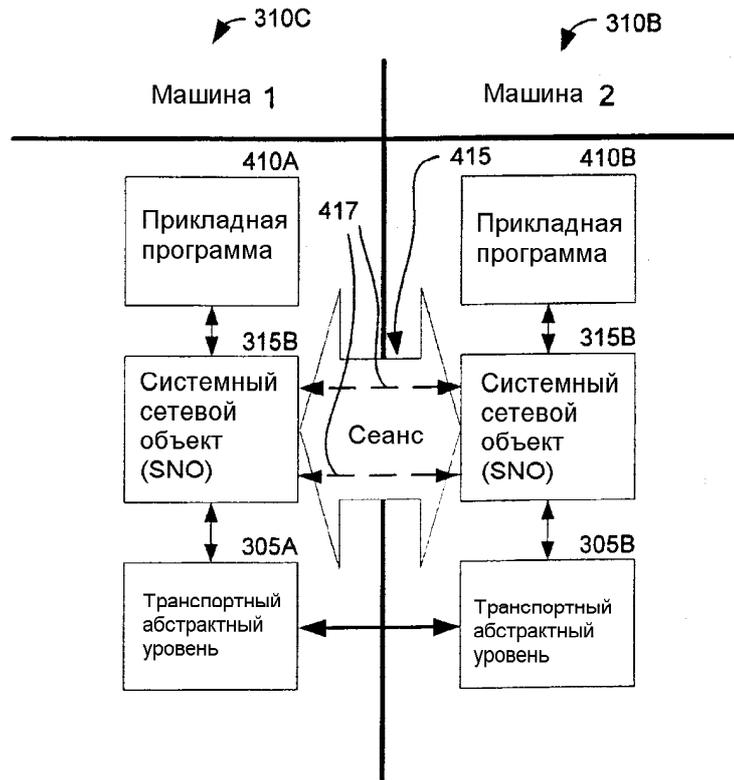


Предшествующий уровень техники

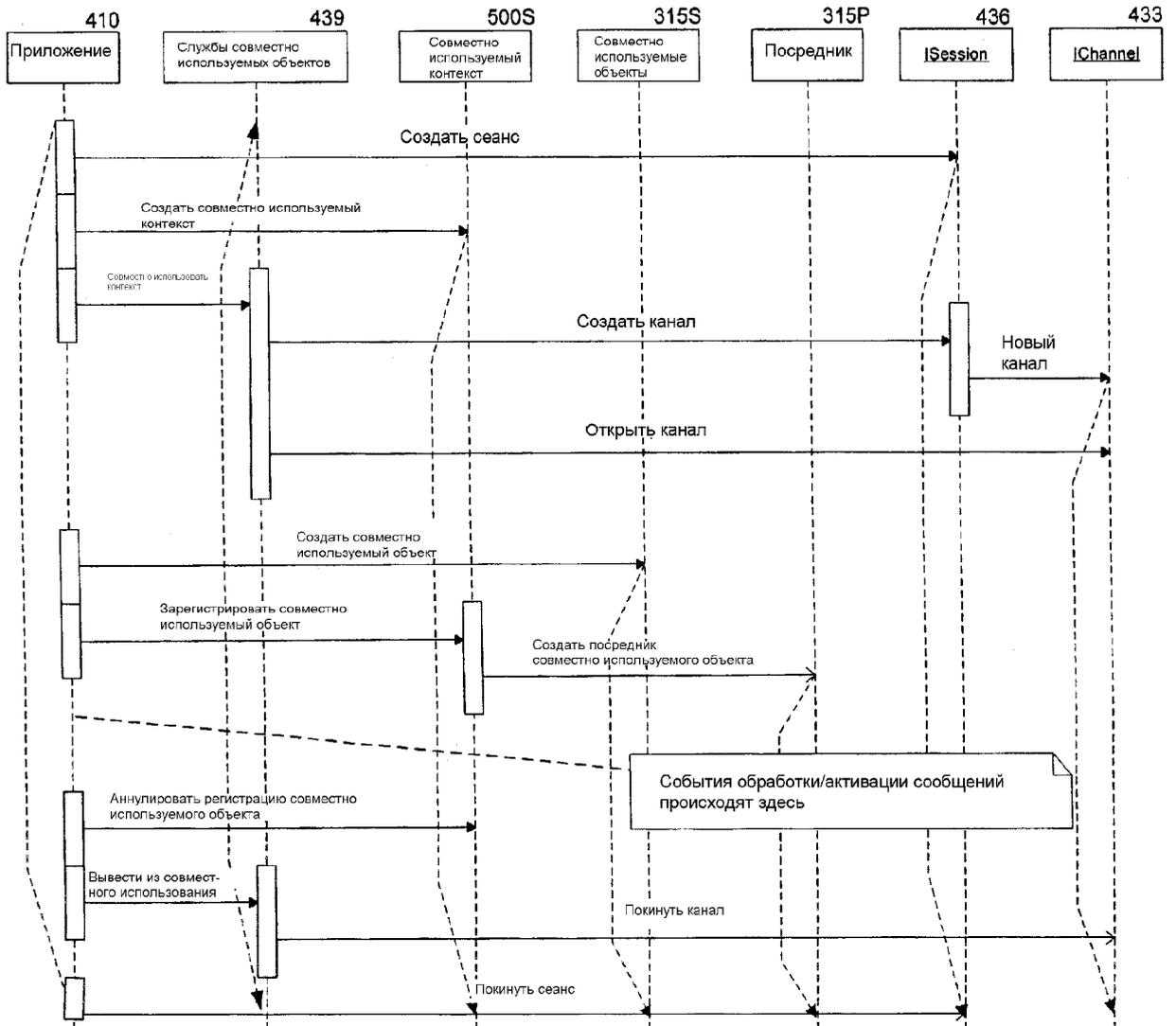
ФИГ.1



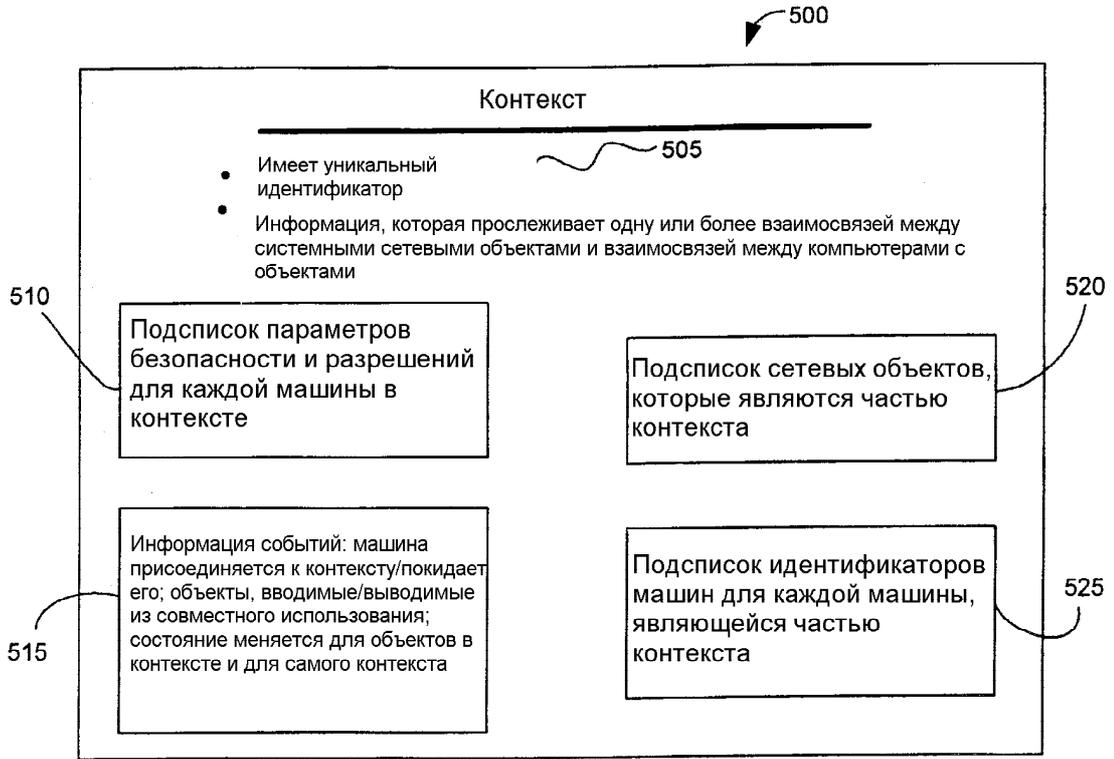
ФИГ.2



ФИГ.4А



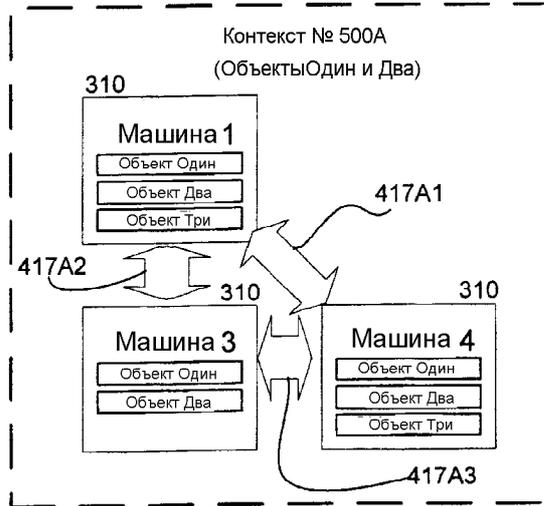
Фиг.4В



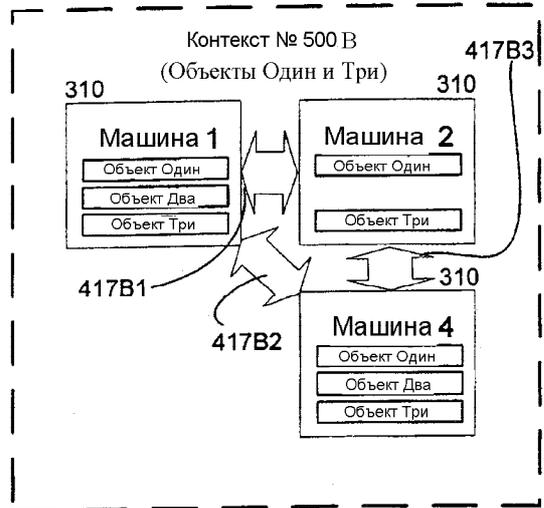
Фиг.5



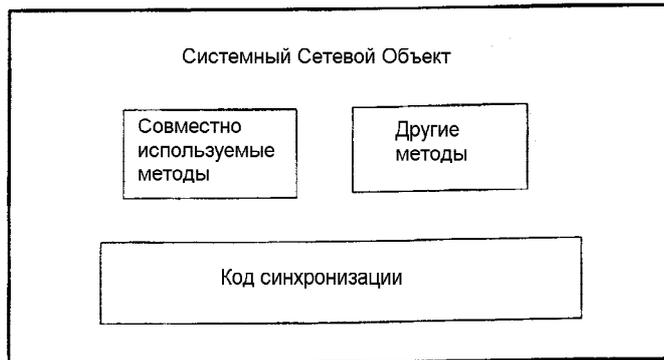
Фиг.6А



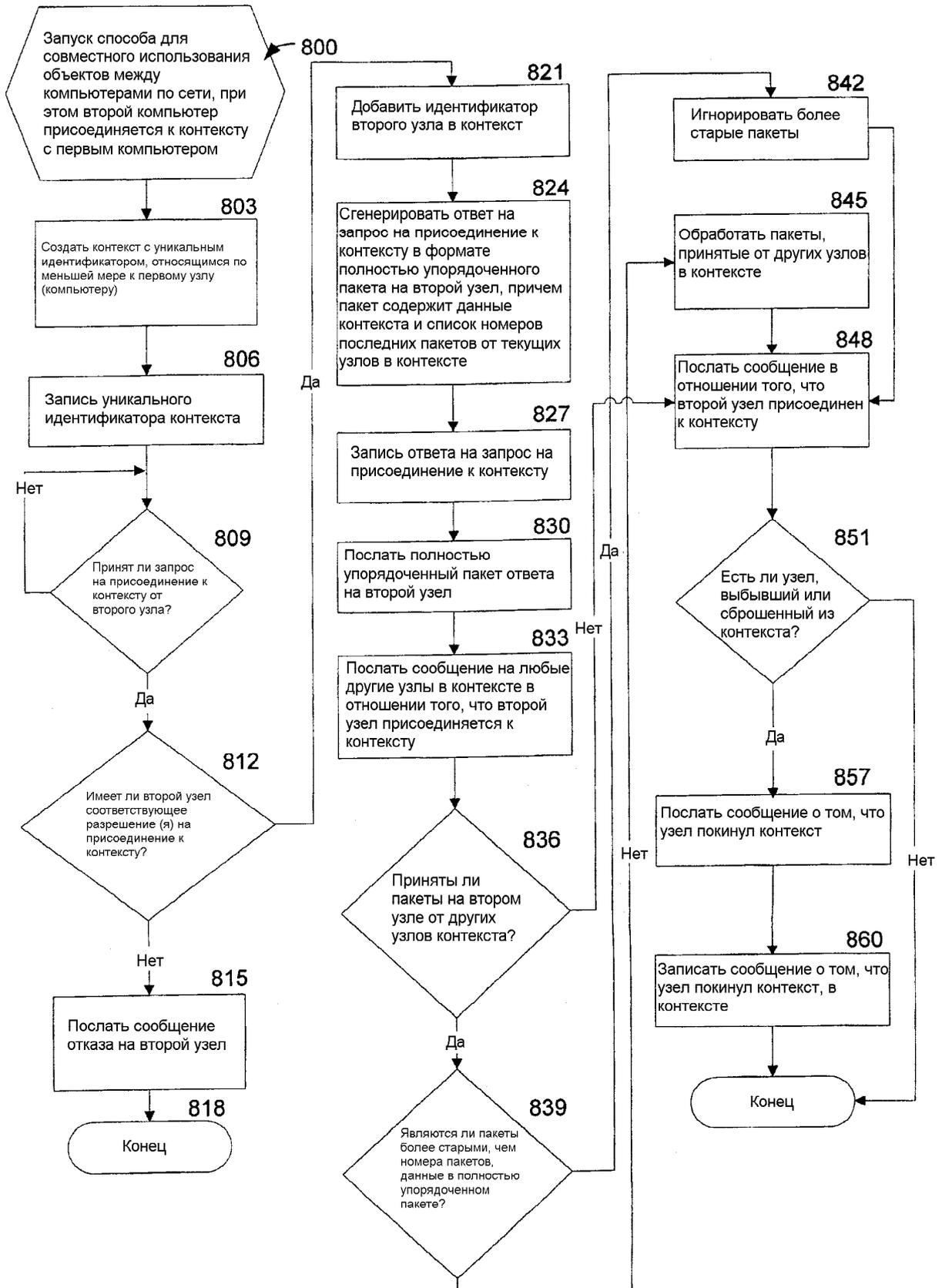
Фиг.6В



Фиг.6С

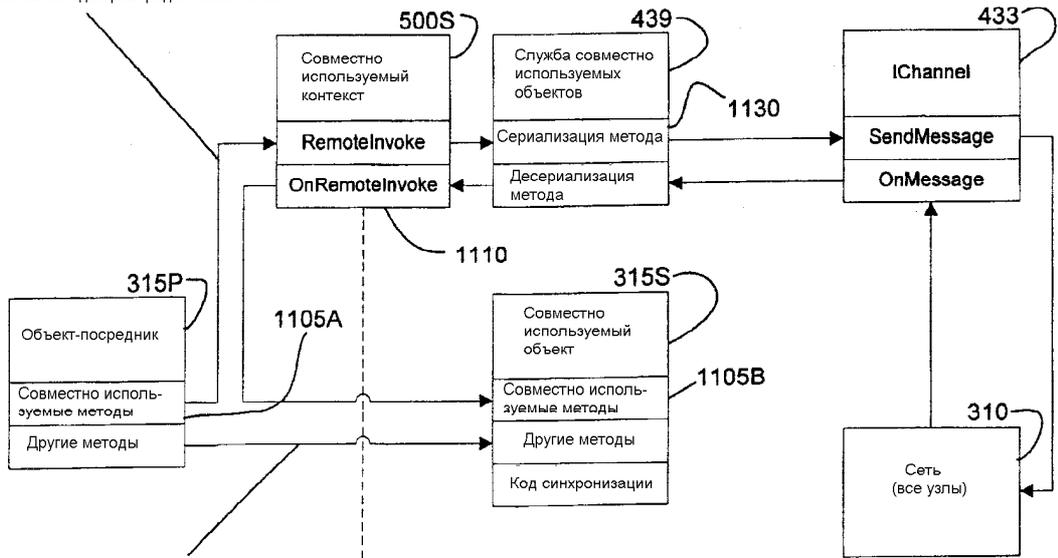


Фиг.7

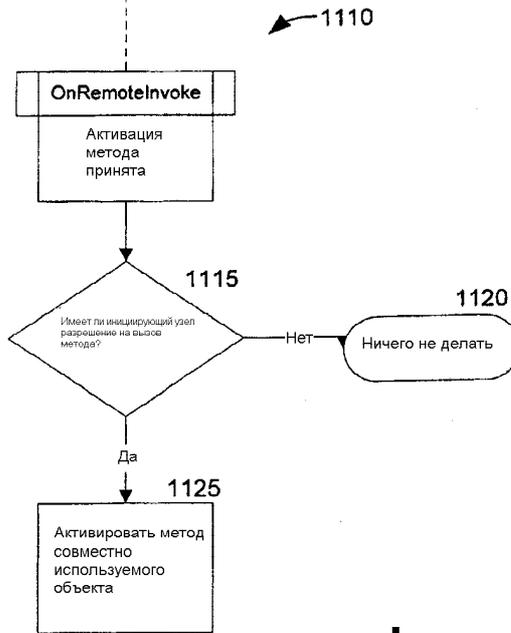


Фиг.8

Здесь показано, как вызовы совместно используемых методов распределяются по сети



Способы, не являющиеся совместно используемыми, передаются непосредственно объекту



ФИГ.9