



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

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21), (22) Заявка: **2006110932/09**, **31.12.2003**

(24) Дата начала отсчета срока действия патента:
31.12.2003

(30) Конвенционный приоритет:
05.09.2003 US 60/500,669

(43) Дата публикации заявки: **20.10.2007**

(45) Опубликовано: **20.04.2009** Бюл. № 11

(56) Список документов, цитированных в отчете о поиске: **RU 14089 U1**, **27.06.2000**. **JP 10-322775**, **04.12.1998**. **CN 124747**, **29.03.2000**. **KR 2002-0024824 A**, **03.04.2002**. **US 5821936 A**, **13.10.1998**.

(85) Дата перевода заявки РСТ на национальную фазу: **05.04.2006**

(86) Заявка РСТ:
KR 03/02934 (31.12.2003)

(87) Публикация РСТ:
WO 2005/025081 (17.03.2005)

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

(72) Автор(ы):

**ЛИ Дзонг-Гоо (KR),
ТОЛЕДАНО Эял (IL),
ЛИНДЕР Натан (IL),
ЭЙЗЕНБЕРГ Ярив (IL),
БЕН-ЯИР Ран (IL)**

(73) Патентообладатель(и):

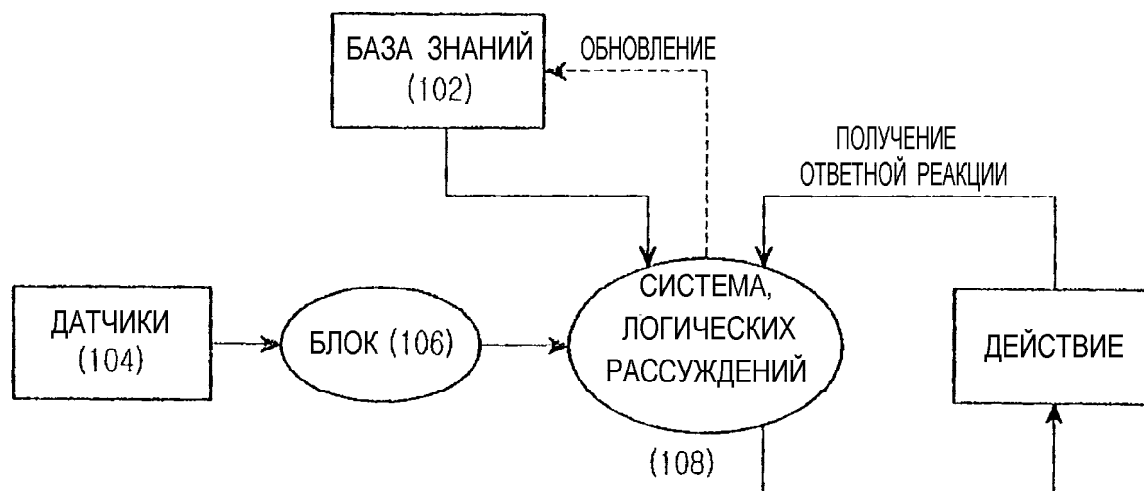
**САМСУНГ ЭЛЕКТРОНИКС КО., ЛТД.
(KR)**

(54) ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ С УПРЕЖДАЮЩИМ ДЕЙСТВИЕМ

(57) Реферат:

Изобретение относится к мобильным информационным устройствам. Техническим результатом является обеспечение изменения меню или его части, путем создания различных меню для их отображения и/или путем изменения функциональных возможностей сенсорного экрана. Заявленный интерфейс пользователя с упреждающим действием может быть установлен в вычислительном устройстве любого типа (или иным образом осуществлять управление этим устройством и/или быть

связанным с ним). В зависимости от поведения пользователя интерфейс пользователя с упреждающим действием активно делает предложения пользователю на основании предшествующего опыта, связанного с конкретным пользователем, и/или различных заранее запрограммированных шаблонов, из которых вычислительное устройство может производить выбор. В интерфейсе также может быть обеспечена звуковая форма предложения. 2 н. и 38 з.п. ф-лы, 45 ил.



Фиг. 1



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS

(51) Int. Cl.

H04M 1/247 (2006.01)**G06F 3/00** (2006.01)**(12) ABSTRACT OF INVENTION**(21), (22) Application: **2006110932/09, 31.12.2003**(24) Effective date for property rights:
31.12.2003(30) Priority:
05.09.2003 US 60/500,669(43) Application published: **20.10.2007**(45) Date of publication: **20.04.2009 Bull. 11**(85) Commencement of national phase: **05.04.2006**(86) PCT application:
KR 03/02934 (31.12.2003)(87) PCT publication:
WO 2005/025081 (17.03.2005)

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):

**LI Dzong-Goo (KR),
TOLEDANO Ehjal (IL),
LINDER Natan (IL),
EhJZENBERG Jariv (IL),
BEN-JaIR Ran (IL)**

(73) Proprietor(s):

SAMSUNG EhLEKTRONIKS KO., LTD. (KR)**(54) ANTICIPATORY USER INTERFACE**

(57) Abstract:

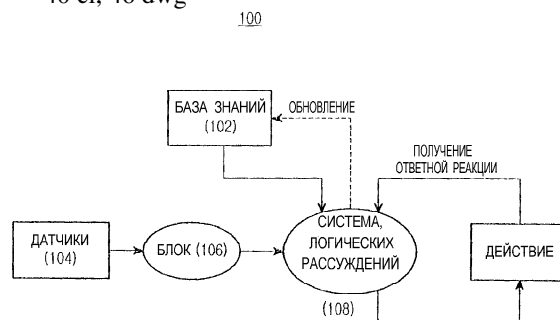
FIELD: information systems.

SUBSTANCE: invention refers to mobile data devices. Applied anticipatory user interface can be installed on any computer (or otherwise control this device and/or be connected). Depending on user behaviour, anticipatory user interface actively offers to the user from the background experience associated with the particular user, and/or the various pre-programmed templates to be chosen by the computer. Besides the interface can support sound data offer.

EFFECT: ensured of change of the menu or its

part by creating various display menus and/or by changing the touch screen functionality.

40 cl, 46 dwg



Фиг. 1

Изобретение относится к интерфейсу пользователя с упреждающим действием и к системе и способам его реализации, в частности, для использования в мобильных информационных устройствах.

ПРЕДПОСЫЛКИ СОЗДАНИЯ ИЗОБРЕТЕНИЯ

5 За последние годы произошел резкий рост использования мобильных и портативных беспроводных устройств. В настоящее время существует множество подобных устройств, имеющих различные функции, внутренние ресурсы и возможности, в том числе телефонные аппараты мобильной связи, персональные
10 цифровые информационные устройства, медицинская и лабораторная измерительная аппаратура, интеллектуальные карточки, компьютерные приставки к телевизору, но эти примеры не являются ограничивающим признаком. Все такие устройства представляют собой мобильные информационные устройства. Существует тенденция, заключающаяся в том, что они становятся все более специализированными
15 устройствами с ограниченными функциональными возможностями, а не универсальными вычислительными устройствами из известного уровня техники. Многие из этих устройств подключены к сети Интернет и используются для разнообразных прикладных задач.

20 Одним из примеров таких мобильных информационных устройств является телефонный аппарат сотовой связи. Телефонные аппараты сотовой связи быстро становятся повсеместно используемыми; использование телефонных аппаратов сотовой связи даже превосходит использование телефонных аппаратов для традиционной коммутируемой телефонной сети общего пользования, КТСОП (PSTN),
25 или телефонных аппаратов для "наземных линий связи". Сами телефонные аппараты сотовой связи становятся более сложными, и на самом деле они фактически представляют собой вычислительные устройства со встроенными операционными системами.

30 Поскольку телефонные аппараты сотовой связи становятся более сложными, то диапазон предоставляемых ими функций также потенциально становится все более широким. Однако в настоящее время эти функции обычно связаны с расширениями функций, уже имеющихся в обычных телефонных аппаратах (для наземных линий связи), и/или с объединением некоторых функций персонального информационного
35 устройства с функциями телефонных аппаратов сотовой связи. Аналогичным образом, интерфейс пользователя, которым снабжены телефонные аппараты сотовой связи, является несложным, обычно характеризуется тем, что имеет кнопочную панель для перемещения по нескольким простым меню. Индивидуальная настройка параметров, хотя она и является явно желательной для потребителей, которые тратят
40 значительные суммы средств на индивидуализированные тональные мелодии звонков и иные аксессуары для телефонного аппарата сотовой связи, по-прежнему ограничена очень небольшим количеством функций телефонного аппарата сотовой связи. Кроме того, в настоящее время у телефонных аппаратов сотовой связи отсутствует
45 автоматическая индивидуализация, например, интерфейса пользователя устройства и заказных/специализированных функциональных возможностей, требующихся для лучшего использования мобильного информационного устройства, и/или способность реагировать в соответствии с поведением пользователя.

50 Однако этот недостаточный уровень совершенства также замечен и в интерфейсах пользователя для персональных (настольных или портативных) компьютеров и иных вычислительных устройств. Индивидуальная настройка параметров в этих вычислительных устройствах обычно может быть произведена только лишь очень

простыми способами. Кроме того, такая индивидуальная настройка параметров должна быть выполнена пользователем, который может не понимать функции компьютера и/или может ощущать неудобства при выполнении таких задач индивидуальной настройки параметров. В настоящее время вычислительные устройства не способны знать стереотипы поведения пользователя и регулировать соответствующим образом свое собственное поведение в качестве адаптивных систем для интерфейса пользователя. Если пользователь не может производить настройку компьютера вручную, то этот пользователь должен согласовать его/ее поведение таким образом, чтобы приспособиться к компьютеру, а не наоборот.

Было разработано программное обеспечение, способное обеспечивать обучение, хотя только лишь для специализированных лабораторных функций. Например, было разработано программное обеспечение "искусственного интеллекта", ИИ (AI). Термину "искусственный интеллект" было дано несколько определений, одним из которых является следующее: "искусственный интеллект - это изучение вычислений, которое позволяет воспринимать, делать логические рассуждения и действовать", (цитата из книги "Artificial Intelligence A Modern Approach (second edition) by Stuart Russell, Peter Norvig (Prentice Hall, Pearson Education Inc, 2003). Программные средства искусственного интеллекта (ИИ) объединяют несколько различных понятий, например, восприятие, которое обеспечивает средство взаимодействия с окружающим миром, в котором необходимы программные средства искусственного интеллекта для логических рассуждений и действий. Примерами являются, в том числе, обработка естественного языка - передача информации, понимание содержания документа и контекста на естественном языке; машинное зрение - восприятие объектов из источника образов и системы датчиков - восприятие объектов и признаков воспринятых объектов путем анализа данных, полученных из датчиков, и т.д., но эти примеры не являются ограничивающим признаком.

Другим важным понятием является понятие "база знаний". Представление знаний является ответственным за представление извлечения и хранения знаний. Эта дисциплина также обеспечивает способы обобщения знаний, выделения и перечисления признаков, построения и определения состояния объекта. Обычно сама реализация может быть выполнена с использованием известных структур данных, например, графиков, векторов, таблиц и т.д.

Еще одним важным понятием является понятие "логических рассуждений". Автоматическое проведение логических рассуждений объединяет алгоритмы, использующие представление знаний и восприятие для получения новых логических выводов, для построения умозаключений по ответам на вопросы и для достижения целей агента. Примерами логических рассуждений в системах искусственного интеллекта (ИИ) являются следующие концептуальные структуры: базы правил - производят оценку системных правил по базе знаний и воспринятому состоянию для логических рассуждений; системы поиска - использование хорошо известных структур данных для поиска интеллектуального умозаключения согласно воспринятому состоянию, доступным знаниям и цели (примерами являются, в том числе, деревья решений, графы состояния, минимаксное решение и т.д.); классификаторы - целью системы логических рассуждений классификатора является классификация воспринятого состояния, отображенного как опыт, не имеющий признака классификации. Согласно базе знаний с предварительной классификацией классификатор делает логический вывод о классификации нового опыта (примерами являются, в том числе, эвристика расстояния между векторами, метод опорных

векторов (Support Vector Machine), нейронная сеть классификатора и т.д.).

Другим важным понятием является обучение. Целью обучения является улучшение потенциально возможного качества функционирования системы искусственного интеллекта (ИИ), реализующей механизм логических рассуждений, путем обобщения по событиям. Ввод алгоритма обучения является экспериментом, а выходные данные представляют собой модификации базы знаний согласно результатам (примерами являются, в том числе, обучение с пополнением (Reinforcement learning), "пакетное" обучение (Batch learning), метод опорных векторов и т.д.).

Некоторыми примерами реализации программных средств искусственного интеллекта, не являющимися ограничивающим признаком, являются, в том числе (все приведенные ниже примеры могут быть найдены в книге "An Artificial Intelligence: A Modern Approach", S. Russell and P. Norvig (eds), Prentice Hall, Pearson Education Inc., NJ, USA, 2003):

Автономное планирование и составление расписания действий: программа "Удаленный агент" ("Remote Agent") Национального управления США по авионавигации и исследованию космического пространства (NASA) стала первой автономной бортовой программой планирования, осуществляющей управление календарным планированием операций для космического корабля. Программа "Удаленный агент" осуществляла генерацию планов, исходя из задач высокого уровня, заданных с Земли, и она осуществляла мониторинг функционирования космического корабля, по мере выполнения планов - обнаружение, диагностику и решение проблем при их возникновении.

Игра в игру: программа "Deep Blue" фирмы IBM стала первой компьютерной программой, победившей чемпиона мира по шахматам.

Автономное управление: система машинного зрения "ALVINN" была обучена для управления автомобилем таким образом, чтобы он отслеживал полосу движения.

Система "ALVINN" вычисляет наилучшее направление движения руля на основании опыта, полученного из предыдущих обучающих поездок.

Диагностика: программы установления медицинского диагноза на основании вероятностного анализа уже были способны действовать на уровне опытного врача в некоторых областях медицины.

Планирование материально-технического снабжения: во время кризиса в Персидском заливе в 1991 году вооруженные силы США развернули средство динамического анализа и корректировки планирования, именуемое DART (Dynamic Analysis and Replanning Tool), для обеспечения автоматизированного планирования материально-технического снабжения и расписания транспортных операций.

Робототехника: в настоящее время в микрохирургии множество хирургов используют роботизированных помощников. Система "HipNav" представляет собой систему, в которой используют способы машинного зрения для создания трехмерной модели внутренней анатомии пациента, а затем используют роботизированное управление для задания направления при введении протеза для эндопротезирования тазобедренного сустава.

Понимание языка и решение задач: программа "PROVERB" представляет собой компьютерную программу, которая решает кроссворды лучше, чем люди, с использованием ограничивающих связей для клеток, заполняемых возможными словами, большой базы данных о прошлых кроссвордах и разнообразных источников информации.

Также уже была проведена работа по созданию генетических алгоритмов и

эволюционных алгоритмов для программного обеспечения. Один из примеров таого программного обеспечения описан в статье Карла Симса (Karl Sims) "Evolving Virtual Creatures" (*Computer Graphics, SIGGRAPH'94 Proceedings*, July 1994, pp.15-22). Это описанные в ссылке программные "виртуальные существа" ("creatures"), которые могут перемещаться по трехмерному виртуальному миру, который представляет собой смоделированную версию реального физического мира. Виртуальные существа могут обучаться и развиваться с использованием генетических алгоритмов, изменяя, тем самым, свое поведение без внешнего управления, осуществляемого путем ввода данных. Следовательно, эти генетические алгоритмы описывают гиперпространство потенциальных вариантов поведения, имеющих различную "приспособленность" или различные поощрения в виртуальном мире. Сами алгоритмы были реализованы с использованием ориентированных графов, описывающих как генотипы (компоненты) виртуальных существ, так и их поведение.

В начале моделирования осуществляют моделирование множества различных виртуальных существ с различными генотипами. Виртуальным существам дают возможность изменять свое поведение в ответ на различные стимулы в виртуальном мире. В каждом "поколении" возможность выживания предоставляют только некоторым виртуальным существам в соответствии либо с относительным, либо с абсолютным минимально допустимым уровнем показателей, определяемым согласно приспособленности поведения виртуальных существ. Разрешено возникновение мутаций, которые могут повысить приспособленность (и, следовательно, жизнеспособность) видоизмененных виртуальных существ или наоборот. Мутации также реализуют посредством ориентированного графа, например, путем изменения значения, связанного с узлом, по случайному закону и/или путем добавления или удаления узлов. Аналогичным образом, "спаривание" между виртуальными существами может привести к изменениям ориентированного графа.

Результаты, описанные в приведенной ссылке, показали, что виртуальные существа на самом деле могут изменяться и развиваться. Однако виртуальные существа могут функционировать только в пределах их виртуального мира и не имеют никакой точки привязки или контакта с реальным физическим миром и/или с людьми - операторами вычислительных машин.

СУЩНОСТЬ НАСТОЯЩЕГО ИЗОБРЕТЕНИЯ

В предшествующем уровне техники отсутствуют идеи или предложения относительно системы или способа, дающих возможность интеллектуальному программному обеспечению, по меньшей мере, для мобильных информационных устройств, обучаться и развиваться, в частности, для взаимодействия с людьми-пользователями. В предшествующем уровне техники также отсутствуют идеи или предложения относительно интрфейса пользователя с упреждающим действием для вычислительного устройства, в котором интерфейс пользователя с упреждающим действием изучает поведение пользователя, а затем способен активно предлагать пользователю возможные опции. В предшествующем уровне техники также отсутствуют идеи или предложения относительно адаптивной системы для мобильного информационного устройства, в которой интерфейс пользователя активно изменяют в соответствии с поведением пользователя. В предшествующем уровне техники также отсутствуют идеи или предложения относительно интеллектуального агента для мобильного информационного устройства, который способен взаимодействовать с человеком-пользователем посредством виртуального персонажа (avatar).

В настоящем изобретении эти недостатки предшествующего уровня техники преодолены за счет создания интерфейса пользователя с упреждающим действием, который может быть при желании установлен в вычислительном устройстве любого типа (или иным образом осуществлять управление этим устройством и/или быть
5 связанным с ним). В зависимости от поведения пользователя интерфейс пользователя с упреждающим действием активно делает предложения пользователю и/или иным образом участвует в недетерминированном или неожиданном поведении на основании предшествующего опыта, связанного с конкретным пользователем, и/или различных
10 заранее запрограммированных шаблонов, из которых вычислительное устройство может производить выбор. Эти предложения могут быть, в частности, сделаны путем изменения внешнего вида, по меньшей мере, части изображения, выводимого на экран дисплея, например, путем изменения меню или его части, путем создания различных
15 меню для отображения и/или путем изменения функциональных возможностей сенсорного экрана. Предложения также могут быть при желании сделаны в звуковой форме. Возможны и иные типы предложений или средств подачи информации.

Следует отметить, что путем "предложения" система фактически может, но не обязательно, выполнять действие автоматически с учетом определенных
20 предпочтений пользователя и также в зависимости от того, позволяет ли состояние системы выполнять действия конкретным способом.

Важно подчеркнуть, что обычно является предпочтительным, чтобы интерфейс пользователя с упреждающим действием, по меньшей мере, выглядел как интеллектуальный и интерактивный, и предпочтительно, чтобы он был способен
25 осуществлять, по меньшей мере, в некоторой степени "свободный" (например, не по сценарию или лишь частично согласно сценарию) обмен информацией с пользователем. Интеллектуальное представление важно в том смысле, что предпочтительно выполняются ожидания пользователя относительно взаимодействий с "интеллектуальным" агентом/устройством. На эти ожидания могут, но не
30 обязательно, оказывать воздействие такие факторы, как способность осуществлять обмен информацией, возможный внешний вид интерфейса, использование человекоподобного атрибута (человекоподобных атрибутов) и т.д., которые предпочтительно используют для усиления ощущения интеллектуальности при
35 взаимодействиях между пользователем и интерфейсом пользователя с упреждающим действием. С точки зрения информации, получаемой от пользователя, предпочтительно, чтобы интерфейс пользователя с упреждающим действием был способен воспринимать то, каким образом пользователь желает взаимодействовать с
40 мобильным информационным устройством. Обмен информацией может, но не обязательно, производиться только в одном направлении, например, возможно, но не обязательно, интерфейс может предоставлять сообщения или информацию пользователю, но не получать информацию от пользователя, или в альтернативном варианте может быть реализована противоположная ситуация. В предпочтительном
45 варианте обмен информацией является двунаправленным для предпочтительных взаимодействий с пользователем.

Для обмена информацией с пользователем предпочтительно, но не обязательно, интерфейс с упреждающим действием способен отображать или демонстрировать
50 имитируемые эмоции для взаимодействий с пользователем в качестве части обмена информацией с пользователем. Как более подробно описано ниже, имитацию этих эмоций для воспроизведения предпочтительно осуществляет интеллектуальный агент, а более предпочтительно их отображают посредством виртуального персонажа или

виртуального существа. Предпочтительно, эмоции создают посредством системы формирования эмоций, управление которой, по меньшей мере, частично, может быть осуществлено, но не обязательно, в соответствии, по меньшей мере, с одним предпочтением пользователя. Систему формирования эмоций предпочтительно используют для обеспечения правдоподобности реакций и связей интеллектуального агента с точки зрения ощущений пользователя; например, если интеллектуальный агент представлен в виде виртуального существа, подобного собаке, то система формирования эмоций предпочтительно обеспечивает соответствие эмоций ожиданиям пользователя относительно поведения, "подобного поведению собаки".

Аналогичным образом, предпочтительно, чтобы интеллектуальный агент, по меньшей мере, выглядел для пользователя как интеллектуальный.

Интеллектуальность может быть, но не обязательно, обеспечена посредством полностью детерминистического механизма, однако, базис, по меньшей мере, для внешнего представления интеллектуальности, предпочтительно содержит, по меньшей мере, один или большее количество случайных или квазислучайных элементов. И вновь, присутствие таких элементов является предпочтительным для обеспечения соответствия ожиданиям пользователя относительно интеллектуальности, связанной с представлением интеллектуального агента.

Предпочтительно, имеется адаптивность для того, чтобы интеллектуальный агент был способен, по меньшей мере, немного изменять поведение для удовлетворения запроса или иного обмена информацией с пользователем. Даже в том возможном, но не обязательном случае, когда интерфейс пользователя с упреждающим действием не содержит интеллектуального агента для обмена информацией с пользователем, адаптивность предпочтительно дает возможность интерфейсу являться интерфейсом с упреждающим действием. Наблюдение взаимодействия пользователя с мобильным информационным устройством предпочтительно позволяет реализовать такую адаптивность, хотя управление реакцией интерфейса пользователя с упреждающим действием на такое наблюдение может предпочтительно, но не обязательно, осуществлять база знаний и/или база правил.

В качестве конкретного примера такой адаптивности, не являющегося ограничивающим, но предпочтительным, в особенности для мобильного информационного устройства, содержащего множество меню, такая адаптивность может предпочтительно содержать способность изменять, по меньшей мере, один аспект меню. Например, может быть, но не обязательно, предусмотрен один ярлык или большее количество ярлыков (shortcuts), дающих пользователю возможность напрямую достигнуть пункта выбора меню в обход, по меньшей мере, одного (а более предпочтительно - всех) предыдущего меню или подменю, которые в иерархии меню находятся на более высоком уровне, чем конечный выбираемый пункт меню. Возможно, но не обязательно (в альтернативном варианте или в дополнение к этому) одно или большее количество меню могут быть перегруппированы в соответствии с адаптивностью интерфейса пользователя с упреждающим действием, например, в соответствии с частотой использования. Такая перегруппировка может, но не обязательно, содержать перемещение части меню, например, пункта выбора меню и/или подменю, в новое местоположение, находящееся в иерархии меню на более высоком уровне, чем текущее местоположение. Доступ к подменю, которые в иерархии меню находятся на более высоком уровне, осуществляют более быстро за счет выбора меньшего количества пунктов выбора меню, чем в том случае, когда они расположены на нижнем (более низком) уровне в иерархии.

Содействие адаптивности и/или эмоциям предпочтительно, но не обязательно, оказывают за счет использования поощрений за обучение посредством интерфейса пользователя с упреждающим действием. Предложения или производимые в соответствии с ними действия, которые одобряет пользователь, предпочтительно обеспечивают поощрение или положительный стимул для интерфейса с упреждающим действием для того, чтобы он продолжал выдавать такие предложения или выполнять такие действия; неодобрение пользователем предпочтительно вызывает создание препятствий для продолжения такого поведения (вариантов поведения) интерфейсом пользователя с упреждающим действием. Создание положительных или отрицательных стимулов/препятствий для интерфейса пользователя с упреждающим действием предпочтительно дает возможность обеспечения более тонких нюансов поведения интерфейса, а не только подхода, более близкого к принципу "черное или белое", в котором линию поведения либо разрешают, либо запрещают. Такие нюансы также являются предпочтительными для обеспечения возможности обработки противоположных или противоречащих линий поведения в том случае, когда такие линии поведения все вместе одобрены/не одобрены пользователем, по меньшей мере, в некоторой степени.

Другая необязательная, но предпочтительная функция интерфейса пользователя с упреждающим действием содержит обучение пользователя. Такое обучение может быть, но не обязательно, проведено для информирования пользователя о способностях мобильного пользовательского устройства. Например, если пользователь не может правильно управлять устройством, например, вводя неправильный вариант выбора, то функция обучения предпочтительно помогает пользователю научиться тому, как следует правильно использовать устройство. Однако в более предпочтительном варианте функция обучения способна предоставлять пользователю инструкцию, по меньшей мере, об одном объекте, не связанном с устройством. Согласно предпочтительному варианту осуществления функции обучения может быть предоставлена инструкция предпочтительно, но не обязательно, о множестве объектов (или, по меньшей мере, путем изменения объекта, не связанного с устройством), а более предпочтительно - посредством гибкой структуры приложений.

Согласно необязательному, но предпочтительному варианту осуществления настоящего изобретения, модель пользователя предпочтительно создают посредством взаимодействия интерфейса пользователя с упреждающим действием и пользователя. Такая модель предпочтительно, но не обязательно, объединяет базы знаний системы искусственного интеллекта (ИИ), определенные исходя из поведения пользователя и/или являющиеся заранее запрограммированными. Кроме того, возможно, но не обязательно, эта модель также обеспечивает для интерфейса пользователя с упреждающим действием возможность измерять реакцию пользователя на конкретные предложения, сделанные интерфейсом пользователя, посредством чего она адаптируется к неявным предпочтениям пользователя.

Примерами таких вычислительных устройств, не являющимися ограничивающими, являются, в том числе, банкоматы (они также имеют подразумеваемые средства обеспечения безопасности, поскольку некоторые стереотипы поведения пользователя могут, например, вызывать сигнал тревоги), обычные компьютеры любого типа (например, настольный компьютер, портативный компьютер, "тонкие" клиенты, носимые компьютеры и т.д.), мобильные информационные устройства, например, телефонные аппараты сотовой связи, пейджеры, иные устройства беспроводной связи,

обычные телефонные аппараты, имеющие операционную систему, персональные информационные устройства и беспроводные персональные информационные устройства, и бытовые приборы, имеющие операционную систему. Используемый ниже термин "вычислительное устройство" охватывает собой любое электронное устройство, имеющее операционную систему и способное выполнять вычисления. Операционная система может, но не обязательно, представлять собой встроенную систему, и/или программное обеспечение иного типа, и/или аппаратную среду выполнения программы. Используемый ниже термин "мобильное информационное устройство" охватывает собой устройства беспроводной связи любого типа без ограничений, в том числе телефонные аппараты сотовой связи, беспроводные пейджеры, беспроводные персональные информационные устройства и т.п., но эти примеры не являются ограничивающим признаком.

Настоящее изобретение предпочтительно реализовано таким образом, что предоставляет пользователю более широкий опыт работы и взаимодействия с вычислительным устройством, а также обеспечивает замену существующего в настоящее время универсального негибкого интерфейса пользователя таких устройств на гибкий интерфейс, который является действительно удобным для пользователя. В более предпочтительном варианте настоящее изобретение реализовано таким образом, что предоставляет более глубокие эмоциональные переживания при взаимодействии пользователя с вычислительным устройством, например, согласно необязательному, но предпочтительному варианту осуществления построения интерфейса пользователя в виде виртуального персонажа, взаимодействующего с пользователем. Следовательно, настоящее изобретение способно обеспечивать жизненный опыт "живого устройства", в частности, для мобильных информационных устройств, таких как, например, телефонные аппараты сотовой связи. Согласно этому варианту осуществления изобретения у пользователя может даже формироваться эмоциональная привязанность к "живому устройству".

Согласно другому варианту осуществления настоящего изобретения в нем предложено мобильное информационное устройство, содержащее адаптивную систему. Подобно описанному выше интерфейсу пользователя оно также основано на предшествующем опыте взаимодействия с пользователем и/или на заранее запрограммированных шаблонах. Однако, адаптивная система предпочтительно, но не обязательно, является более ограниченной по функционированию в рамках функций и среды мобильного информационного устройства.

Любой или оба из объектов, которыми являются адаптивная система мобильного информационного устройства и интерфейсы пользователя с упреждающим действием, предпочтительно, но не обязательно, могут быть реализованы посредством генетических алгоритмов, алгоритмов искусственного интеллекта (ИИ), алгоритмов обучения машины (ОМ), поведения, приобретенного в результате обучения, и программных средств/вычислительных устройств, которые способны к эволюции. Любой или оба из этих объектов также могут, но не обязательно, обеспечивать расширенный уровень речевых команд, команд сенсорного экрана и комбинаций клавиш клавиатуры для быстрого вызова.

Согласно другому необязательному, но предпочтительному варианту осуществления настоящего изобретения предложен один или большее количество интеллектуальных агентов для использования в мобильном информационном устройстве в сети мобильных информационных устройств, предпочтительно содержащих виртуальный персонаж (или "виртуальное существо", ниже эти термины

используют как взаимозаменяемые), посредством которого агент может осуществлять обмен информацией с человеком-пользователем. Следовательно, виртуальный персонаж предпочтительно обеспечивает интерфейс пользователя для взаимодействия с пользователем. Интеллектуальный агент также предпочтительно содержит агента для управления, по меньшей мере, одним взаимодействием мобильного информационного устройства по сети. Этот вариант осуществления может, но не обязательно, содержать множество таких интеллектуальных агентов, связанных через сеть мобильных информационных устройств, в силу чего возможно, но не обязательно, образуется сеть таких агентов. Посредством этого варианта осуществления изобретения также может быть, но не обязательно, обеспечено выполнение различных прикладных задач, которыми являются, в том числе, следующие: обучение вообще и/или, в частности, обучение тому, как следует использовать мобильное информационное устройство, обучение языкам, прикладные задачи обмена информацией, прикладные задачи коммунального хозяйства, игры, развлечения, покупка товаров (получение купонов и т.д.), определение местонахождения магазина или иного места, фильтрация рекламных объявлений и иных сообщений без навязывания услуг, ролевые или иные интерактивные игры по сотовой телефонной сети, функции "интерактивной переписки" и встреч, способность покупать "подарки" интеллектуальным агентам и иным образом дополнять персонаж аксессуарами и т.д., но эти примеры не являются ограничивающим признаком. Теоретически, самим агентам в качестве аксессуаров можно давать "домашних животных".

Интеллектуальные агенты также могут, но не обязательно, помогать в создании различных возможностей деловой деятельности/стимулирования сбыта для операторов сотовых телефонов. Предпочтительно, но не обязательно, агенты также могут оказывать содействие при установке системного программного обеспечения в сотовые телефоны, что является новой областью торговли. Например, агенты могут, но не обязательно, содействовать определению надлежащего типа мобильного информационного устройства и иных подробностей, которые являются основными для правильной загрузки и работы программного обеспечения.

Предпочтительно, но не обязательно, интеллектуальный агент также может осуществлять обучение пользователя путем обучения пользователя тому, каким образом следует приводить в действие различные функции в самом мобильном информационном устройстве, например, каким образом следует передавать или получать сообщения, использовать будильник и т.д. Как более подробно описано ниже, такие функции обучения также могут быть, но не обязательно, расширены на обучение пользователя информации/функциям, которые являются внешними относительно самого мобильного информационного устройства. Такие функции обучения предпочтительно являются расширенными за счет обмена информацией между множеством агентов в сети, что дает агентам возможность получать информацию, распределенную между агентами в сети.

Предпочтительно, но не обязательно, оплата за агентов может быть произведена по подписке, а в альтернативном варианте "подпитка" агентов может, но не обязательно, быть произведена посредством действий оплаты с предоплаченного счета пользователя и/или оплаты путем выставления счета пользователю в конце месяца.

Следовательно, согласно различным вариантам осуществления настоящего изобретения возможны несколько различных взаимодействий. Эти взаимодействия

содержат любое одно или большее количество взаимодействий между пользователем устройства и виртуальным персонажем, или иным персонажем, или персонифицированным образом устройства; взаимодействий между пользователем устройства и устройством для работы с устройством посредством виртуального персонажа, или иного персонажа, или персонифицированного образа; взаимодействий между двумя пользователями посредством их соответствующих устройств путем обмена информацией через виртуальный персонаж, или иной персонаж, или персонифицированный образ устройства; и взаимодействий между двумя устройствами через соответствующие им интеллектуальные агенты, возможно, но не обязательно, без какого-либо обмена информацией между пользователями или даже между агентом и пользователем. Возможное взаимодействие или возможные взаимодействия определены согласно варианту осуществления настоящего изобретения, а их более подробное описание приведено ниже.

Настоящее изобретение извлекает выгоду из относительно ограниченной среды вычислительного устройства и/или мобильного информационного устройства, такого как, например, телефонный аппарат сотовой связи, поскольку параметры такой среды известны заранее. Даже в том случае, если такие устройства поддерживают связь через сети, такие как, например, сеть сотовой телефонной связи, параметры среды могут быть, тем не менее, заранее заданными. В настоящее время вычислительные устройства обеспечивают только лишь типовой интерфейс, в котором разрешена лишь незначительная индивидуальная настройка параметров даже путем ручного, прямого вмешательства пользователя, или же такая настройка вообще не разрешена.

Следует отметить то, что термин "программное обеспечение" также может, но не обязательно, содержать аппаратно реализованное программное обеспечение или команды, управление которыми осуществляют аппаратными средствами.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Изобретение описано здесь только лишь посредством примеров со ссылкой на сопроводительные чертежи, на которых изображено следующее:

на Фиг.1 изображена блок-схема обучающегося модуля согласно настоящему изобретению, который приведен в качестве примера;

на Фиг.2 изображена блок-схема системы согласно настоящему изобретению для использования интерфейса пользователя с упреждающим действием, которая приведена в качестве примера;

на Фиг.3 показан пример варианта реализации системы интерфейса пользователя с упреждающим действием согласно настоящему изобретению;

на Фиг.4 изображена блок-схема варианта реализации адаптивной системы согласно настоящему изобретению, который приведен в качестве примера;

на Фиг.5А и Фиг.5В показаны, соответственно, блок-схема системы управления приложениями согласно настоящему изобретению, которая приведена в качестве примера, и схема последовательности выполняемых в ней операций;

на Фиг.6А и Фиг.6В показан пример инфраструктуры, требуемой для выполнения адаптивной системой согласно настоящему изобретению одного или большего количества действий посредством операционной системы мобильного информационного устройства, и пример схемы последовательности выполняемых в ней операций согласно настоящему изобретению;

на Фиг.7А - Фиг.7С показаны приведенные в качестве примера события и то, каким образом осуществляют их обработку путем взаимодействий между мобильным информационным устройством (посредством операционной системы устройства) и

системой из настоящего изобретения;

на Фиг.8 изображены пример структуры интеллектуального агента (Фиг.8А), и приведенная в качестве примера схема последовательности операций, выполняемых интеллектуальным агентом (Фиг.8В);

на Фиг.9А и Фиг.9В показаны два способа выбора действия согласно настоящему изобретению, которые приведены в качестве примеров;

на Фиг.10 показана схема последовательности операций, выполняемых в способе выполнения действия согласно настоящему изобретению, который приведен в качестве примера;

на Фиг.11А - Фиг.11С изображены схемы, описывающие иллюстративный пример варианта реализации системы формирования эмоций согласно настоящему изобретению;

на Фиг.12 показан пример схемы последовательности операций для обмена текстовой информацией согласно настоящему изобретению;

на Фиг.13А и Фиг.13В показаны, соответственно, пример схемы классов и пример схемы последовательности операций для обработки телефонного вызова согласно настоящему изобретению;

на Фиг.14А и Фиг.14В изображены иллюстративные, не являющиеся ограничивающими примеры, соответственно, класса и схемы последовательности операций обработки SMS-сообщений (сообщений службы коротких сообщений) согласно настоящему изобретению;

на Фиг.15 показан пример схемы классов обработки меню согласно настоящему изобретению;

на Фиг.16 показан пример схемы классов игр согласно настоящему изобретению;

на Фиг.17А показан пример схемы классов обучающей машины, а на Фиг.17В показан пример схемы последовательности операций, выполняемых в обучающей машине, согласно настоящему изобретению;

на Фиг.18А - Фиг.18С показаны, соответственно, пример схемы классов эволюции, пример схемы последовательности операций при мутации и пример схемы последовательности операций при гибридизации согласно настоящему изобретению;

на Фиг.19 показан пример последовательности гибридизации между интеллектуальными агентами в двух мобильных информационных устройствах;

на Фиг.20 - Фиг.26 показаны примеры снимков экрана дисплея с изображениями виртуального персонажа или существа согласно настоящему изобретению;

на Фиг.27 показана схема системы интеллектуального агента согласно настоящему изобретению, которая приведена в качестве примера;

на Фиг.28 показана система по Фиг.27 в более подробном виде;

на Фиг.29 показана блок-схема варианта реализации системы выбора действия согласно настоящему изобретению, который приведен в качестве примера; и

на Фиг.30А и Фиг.30В показано несколько примеров снимков экрана с изображениями виртуального персонажа согласно настоящему изобретению, выведенными на экран мобильного информационного устройства.

ОПИСАНИЕ ПРЕДПОЧТИТЕЛЬНЫХ ВАРИАНТОВ ОСУЩЕСТВЛЕНИЯ ИЗОБРЕТЕНИЯ

В настоящем изобретении предложен интерфейс пользователя с упреждающим действием, который может быть, но не обязательно, установлен в вычислительном устройстве любого типа (или иным образом осуществлять управление этим устройством и/или быть связанным с ним). В зависимости от поведения пользователя

интерфейс пользователя с упреждающим действием активно делает предложения пользователю на основании предшествующего опыта, связанного с конкретным пользователем, и/или различных заранее запрограммированных шаблонов, из которых вычислительное устройство может производить выбор. Эти предложения могут быть, но не обязательно, сделаны путем изменения внешнего вида, по меньшей мере, части изображения, выводимого на экран дисплея, например, путем изменения меню или его участка, путем создания различных меню для дисплея и/или путем изменения функциональных возможностей сенсорного экрана. Предложения также могут быть, но не обязательно, сделаны в звуковой форме.

Как описано выше, интерфейс пользователя с упреждающим действием предпочтительно реализован для вычислительного устройства, содержащего операционную систему. Интерфейс предпочтительно, но не обязательно, содержит интерфейс пользователя для обмена информацией между пользователем и операционной системой. Интерфейс также предпочтительно содержит обучающийся модуль для выявления, по меньшей мере, одного шаблона взаимодействия пользователя с интерфейсом пользователя и для упреждающего изменения, по меньшей мере, одной функции интерфейса пользователя в соответствии с выявленным шаблоном. Следовательно, интерфейс пользователя с упреждающим действием может предугадывать запросы пользователя и, тем самым, содействовать пользователю в выборе желаемой функции вычислительного устройства.

Предпочтительно, но не обязательно, выбирают, по меньшей мере, один шаблон из группы, состоящей из шаблона, определенного согласно, по меньшей мере, одному предыдущему взаимодействию пользователя с интерфейсом пользователя, и заранее заданного шаблона, или из их комбинации. Шаблон первого типа представляет поведение, приобретенное в результате обучения, а шаблон второго типа может, но не обязательно, быть заранее запрограммированным или заданным заранее иным способом, в частности, для помощи пользователю тогда, когда пользователь впервые приводит в действие конкретное вычислительное устройство. Необязательный шаблон третьего возможного типа объединяет эти два аспекта и дает возможность задавать шаблон таким образом, чтобы он, по меньшей мере, частично соответствовал поведению пользователя, но не полностью; например, выбор шаблона может, но не обязательно, быть произведен согласно множеству правил и/или согласно ограничивающему определению возможного состояния среды окружающего мира и/или состоянию устройства и/или интерфейса пользователя (более подробное объяснение этого приведено ниже).

Интерфейс пользователя предпочтительно содержит такой графический дисплей, в котором производят упреждающее изменение, по меньшей мере, одной функции графического дисплея согласно шаблону. Например, предпочтительно, но не обязательно, может быть изменен, по меньшей мере, участок графического дисплея, а в более предпочтительном варианте - путем выбора отображаемого меню согласно выявленному шаблону, и произведено отображение меню. Меню может быть выбрано, но не обязательно, путем построения меню из множества пунктов меню, например, для создания меню "в оперативном режиме".

Дополнительно к этому или в альтернативном варианте интерфейс пользователя может содержать средство воспроизведения звука (audio display), при этом изменение, по меньшей мере, одной функции интерфейса пользователя приводит к изменению, по меньшей мере, одного слышимого звука, созданного вычислительным устройством.

Предпочтительно, но не обязательно, интерфейс пользователя с упреждающим

действием может быть реализован согласно способу из настоящего изобретения, который предпочтительно реализован для обеспечения упреждающего взаимодействия между пользователем и вычислительным устройством посредством интерфейса пользователя. Способ предпочтительно содержит следующие операции:

5 выявляют шаблон поведения пользователя согласно, по меньшей мере, одному взаимодействию пользователя с интерфейсом пользователя; и производят упреждающее изменение, по меньшей мере, одной функции интерфейса пользователя согласно шаблону.

10 Согласно другому варианту осуществления настоящего изобретения в нем предложено мобильное информационное устройство, содержащее адаптивную систему. Подобно описанному выше интерфейсу пользователя она также основана на предшествующем опыте взаимодействия с пользователем и/или на заранее запрограммированных шаблонах. Однако адаптивная система предпочтительно, но

15 не обязательно, является более ограниченной по функционированию в рамках функций и среды мобильного информационного устройства, такого как, например, телефонный аппарат сотовой связи, который в настоящее время может также содержать некоторые основные функции персонального информационного

20 устройства.

Адаптивная система предпочтительно работает в мобильном информационном устройстве, содержащем операционную систему. Операционная система может, но не обязательно, содержать встроенную систему. Мобильное информационное устройство может, но не обязательно, содержать телефонный аппарат сотовой связи.

25 Адаптивная система предпочтительно является способной анализировать поведение пользователя путем анализа множества взаимодействий пользователя с мобильным информационным устройством, после чего в более предпочтительном варианте адаптивная система сравнивает множество взаимодействий пользователя, по меньшей мере, с одним заранее заданным шаблоном для того, чтобы увидеть, связан ли заранее заданный шаблон с изменением, по меньшей мере, одной функции интерфейса

30 пользователя. В альтернативном варианте или в дополнение к этому операция анализа может, но не обязательно, содержать операцию сравнения множества взаимодействий пользователя, по меньшей мере, с одним шаблоном заранее выявленного поведения пользователя, при этом шаблон заранее выявленного поведения пользователя связан с изменением, по меньшей мере, одной функции интерфейса пользователя.

Функция интерфейса пользователя может, но не обязательно, содержать операцию создания мобильным информационным устройством слышимого звука, который в

40 более предпочтительном варианте выбран из группы, состоящей, по меньшей мере, из одного из следующих звуков: тональная мелодия звонка, тональный сигнал будильника и тональный сигнал входящего сообщения. В альтернативном варианте или в дополнение к этому эта операция может быть, но не обязательно, связана с визуальным отображением информации мобильным информационным устройством.

45 Например, визуальное отображение информации может, но не обязательно, содержать операцию отображения меню.

Управление адаптивной системой может, но не обязательно, осуществлять само мобильное информационное устройство. В альтернативном варианте, если мобильное

50 информационное устройство соединено с сетью, то возможно, но не обязательно, управление адаптивной системой, по меньшей мере, частично осуществляют согласно командам, переданным из сети в мобильное информационное устройство. Для этого варианта реализации данные, связанные, по меньшей мере, с одной операцией

адаптивной системы, предпочтительно хранят в ином месте, чем мобильное информационное устройство, причем доступ к этому месту может быть осуществлен через сеть.

5 Согласно предпочтительным вариантам осуществления настоящего изобретения адаптивная система также содержит обучающийся модуль, предназначенный для выполнения анализа согласно принятой входной информации и перед этим
полученным знаниям. Такие знания могут быть, но не обязательно, уже полученными
10 заранее, исходя из поведения пользователя, и/или могут быть уже переданными из другой адаптивной системы, поддерживающей связь с адаптивной системой конкретного мобильного информационного устройства. Адаптивная система предпочтительно, но не обязательно, адаптируется к поведению пользователя согласно какому-либо одному или большему количеству следующих алгоритмов: алгоритму
15 искусственного интеллекта (ИИ), алгоритму обучения машины или генетическому алгоритму.

Согласно другому необязательному, но предпочтительному варианту осуществления настоящего изобретения в нем предложен один или большее количество интеллектуальных агентов для использования в мобильном
20 информационном устройстве в сети мобильных информационных устройств, предпочтительно содержащем виртуальный персонаж, посредством которого агент может производить обмен информацией с человеком-пользователем. Следовательно, виртуальный персонаж предпочтительно обеспечивает интерфейс пользователя для взаимодействия с пользователем. Интеллектуальный агент предпочтительно также
25 содержит агента для управления, по меньшей мере, одним взаимодействием мобильного информационного устройства по сети. Этот вариант осуществления может, но не обязательно, содержать множество таких виртуальных персонажей, связанных через сеть мобильных информационных устройств.

30 Согласно предпочтительным вариантам осуществления настоящего изобретения управление как виртуальным персонажем, так и агентом, осуществляет мобильное информационное устройство. В альтернативном варианте или в дополнение к этому мобильное информационное устройство поддерживает связь, по меньшей мере, с одним другим мобильным информационным устройством, имеющим второго агента,
35 при этом первый и второй агенты предпочтительно способны осуществлять обмен информацией друг с другом. Такой обмен информацией может, но не обязательно, производиться непосредственно, например, посредством оптической связи в инфракрасном диапазоне непосредственно между двумя мобильными
40 информационными устройствами, либо, в альтернативном варианте или в дополнение к этому, - через сеть мобильных информационных устройств. Например, если сетью является сеть сотовой телефонной связи, то обмен информацией может, но не обязательно, производиться с использованием стандартных протоколов связи, протокола сети Интернет/протокола передачи гипертекста (IP/HTTP), протокола
45 службы коротких сообщений (SMS) и т.д.

Кроме того, в необязательном и в более предпочтительном варианте пользователи соответствующих мобильных информационных устройств предпочтительно являются способными осуществлять обмен информацией посредством своих соответствующих
50 виртуальных персонажей. Такой обмен информацией может быть, но не обязательно, связан с игрой, например, с ролевой игрой.

Подобно описанной выше адаптивной системе управление одним из объектов, которыми являются виртуальный персонаж и агент, или обоими этими объектами

может быть, но не обязательно, осуществлено, по меньшей мере, частично, согласно командам, переданным из сети мобильных информационных устройств в мобильное информационное устройство. Кроме того, данные, связанные, по меньшей мере, с одной операцией, по меньшей мере, одного из объектов, которыми являются

5 виртуальный персонаж и агент, возможно, но не обязательно, хранят в ином месте, чем мобильное информационное устройство, причем доступ к упомянутому месту может быть осуществлен через сеть мобильных информационных устройств.

Согласно предпочтительным вариантам осуществления настоящего изобретения,

10 предпочтительно, по меньшей мере, что одна характеристика внешнего вида виртуального персонажа является изменяемой, например, возможно, но не обязательно, согласно команде пользователя. В более предпочтительном, но не обязательном варианте множество характеристик внешнего вида виртуального персонажа являются изменяемыми согласно заранее заданной графической оболочке

15 виртуального персонажа. Графическая оболочка является, возможно, но не обязательно, заранее заданной пользователем. Под термином "графическая оболочка" ("skin") подразумевают, что множество характеристик изменяют вместе как совокупность, при этом данная совокупность образует графическую оболочку. Если

20 этот вариант осуществления изобретения объединен с предыдущим вариантом осуществления изобретения, в котором, по меньшей мере, часть данных, связанных с виртуальным персонажем, хранят в месте, доступ к которому может быть осуществлен через сеть, то пользователь может по выбору перемещать один и тот же

25 виртуальный персонаж в различные телефоны и/или производить индивидуальную настройку внешнего вида виртуального персонажа по различным причинам, например, для особых случаев, таких как вечеринка или иное торжественное событие. Конечно, подразумевают, что эти варианты приведены только лишь в качестве примеров и никоим образом не являются ограничивающими.

Согласно другим необязательным, но предпочтительным вариантам осуществления настоящего изобретения, предпочтительно, по меньшей мере, одна характеристика внешнего вида виртуального персонажа является изменяемой согласно

30 автоматизированному эволюционному алгоритму, например, генетическому алгоритму. Одним не ограничивающим примером способа обеспечения индивидуализации виртуального персонажа для пользователя является эволюционный алгоритм. Индивидуализация также может быть, но не обязательно, выполнена путем

35 непосредственного выбора пользователем одной или большего количества характеристик или графических оболочек (совокупности характеристик). Такая индивидуализация является желательной, по меньшей мере, частично, поскольку она усиливает эмоциональные переживания пользователя при его взаимодействии с

40 виртуальным персонажем и, следовательно, с мобильным информационным устройством.

Согласно еще одним необязательным, но предпочтительным вариантам

45 осуществления настоящего изобретения сеть мобильных информационных устройств содержит устройство определения местоположения для определения физического местоположения мобильного информационного устройства, поэтому пользователь предпочтительно является способным выдавать запрос на получение информации об

50 этом физическом местоположении посредством действия агента. Устройство определения местоположения также предпочтительно является способным определять второе физическое местоположение относительно физического местоположения мобильного информационного устройства, поэтому пользователь может выдавать

запрос на получение информации о втором физическом местоположении посредством действия агента.

Предпочтительно, но не обязательно, пользователь может выбирать второе физическое местоположение и выдавать запрос на получение информации о нем
5 согласно категории, которая может быть, но не обязательно, выбрана из группы, состоящей из местоположения торгового предприятия, местоположения медицинского учреждения и местоположения учреждения обеспечения общественной безопасности, например, пожарной станции или отделения полиции.

Например, если пользователь желает найти местоположение торгового
10 предприятия, то пользователь вводит запрос о местоположении такого предприятия, возможно, но не обязательно, по наименованию магазина, торгового центра и т.д., или в альтернативном варианте - по типу торгового предприятия, например, по типу магазина, или даже по общей категории, которой является, например, категория
15 "торговый центр". Возможно, но не обязательно, соответствующее торговое предприятие может передать в мобильное информационное устройство сообщение согласно упомянутому действию упомянутого агента, которое, например, предпочтительно, но не обязательно, содержит, по меньшей мере, одно из следующих
20 сообщений: рекламное объявление или купон, либо оба эти сообщения вместе. Для таких сообщений или, на самом деле, для сообщения любого типа, агент предпочтительно отфильтровывает сообщение согласно, по меньшей мере, одному критерию, который в более предпочтительном варианте введен пользователем посредством виртуального персонажа, и/или которому обучен виртуальный персонаж
25 в ответ на предыдущее действие пользователя после получения сообщения. Затем виртуальный персонаж может, но не обязательно, предоставить пользователю если не само сообщение (полностью или частично), то, по меньшей мере, информацию о сообщении.

Предпочтительно, пользователь также выдает запрос на получение информации о
30 втором физическом местоположении посредством виртуального персонажа.

Конечно же, местоположение торгового предприятия не обязательно должно
являться физическим местоположением; им также может, но не обязательно, являться
35 виртуальное местоположение торгового предприятия, например, для торговли через сеть мобильной связи, при этом пользователь поддерживает связь с виртуальным местоположением торгового предприятия посредством виртуального персонажа. Предпочтительно, пользователь может производить покупку в виртуальном торговом предприятии посредством виртуального персонажа. Пользователь также может, но не
40 обязательно, производить поиск в виртуальном торговом предприятии с использованием агента, хотя снова он предпочтительно использует виртуальный персонаж в качестве интерфейса. Как описано выше, возможно, но не обязательно, виртуальный персонаж даже способен получать аксессуар, приобретенный в виртуальном торговом предприятии.

К тому же, если мобильное информационное устройство способно производить
45 прием программного обеспечения, то предпочтительно агент выполняет, по меньшей мере, часть процедуры установки программного обеспечения в мобильном информационном устройстве. Пользователь может, но не обязательно,
50 взаимодействовать с виртуальным персонажем для выполнения, по меньшей мере, части операции конфигурирования программного обеспечения.

С точки зрения технической реализации настоящее изобретение предпочтительно является способным функционировать в ограниченной системе (с точки зрения объема

памяти, производительности обработки данных, размера экрана дисплея, разрешающей способности и т.д.) в устройстве, которое также является очень личным для пользователя. Например, устройством предпочтительно является мобильное информационное устройство, например, телефонный аппарат сотовой связи, который по необходимости выполнен таким образом, что является портативным и простым в использовании и, следовательно, может иметь одно или большее количество вышеупомянутых ограничений, или же все эти ограничения. Аспекты реализации настоящего изобретения предпочтительно приспособлены к этой комбинации характеристик. Следовательно, ниже предложены различные технические решения для преодоления ограничений, присущих самому устройству, обеспечивающие при этом поддержку желательной индивидуализации и "личных ощущений" для пользователя. Следует отметить, что эти технические решения являются только лишь примерами и подразумевают, что они никоим образом не являются ограничивающим признаком.

ПРИМЕР 1: ИНТЕРФЕЙС С УПРЕЖДАЮЩИМ ДЕЙСТВИЕМ - Общее описание

Интерфейс пользователя с упреждающим действием из настоящего изобретения предпочтительно является способным осуществлять управление вычислительным устройством любого типа и/или быть связанным с ним для того, чтобы в зависимости от поведения пользователя активно делать пользователю предложения на основании предшествующего опыта, связанного с конкретным пользователем и/или различных заранее запрограммированных шаблонов, из которых вычислительное устройство может производить выбор. Эти предложения могут быть, но не обязательно, сделаны путем изменения внешнего вида, по меньшей мере, части изображения, выводимого на экран дисплея, например, путем изменения меню или его участка, путем создания различных меню для дисплея и/или путем изменения функциональных возможностей сенсорного экрана. Предложения также могут быть, но не обязательно, сделаны в звуковой форме.

Как описано выше, интерфейс пользователя с упреждающим действием предпочтительно реализован для вычислительного устройства, содержащего операционную систему. Предпочтительно, но не обязательно, интерфейс содержит интерфейс пользователя для обмена информацией между пользователем и операционной системой. Интерфейс предпочтительно является способным выявлять, по меньшей мере, один шаблон взаимодействия пользователя с интерфейсом пользователя, например, посредством действия обучающегося модуля, и, следовательно, предпочтительно является способным осуществлять упреждающее изменение, по меньшей мере, одной функции интерфейса пользователя согласно выявленному шаблону. Следовательно, интерфейс пользователя с упреждающим действием может предугадывать запросы пользователя и, тем самым, содействовать пользователю в выборе желаемой функции вычислительного устройства.

Этот тип поведения с упреждающим действием, в частности, применительно к изучению поведения и желаний пользователя, требует определенной способности к обучению со стороны интерфейса с упреждающим действием. Такие способности к обучению могут быть, но не обязательно, предоставлены посредством алгоритмов и методологий из известного уровня техники, относящихся к обучению (посредством программного обеспечения) и взаимодействиям программного объекта с окружающей средой. Можно сказать, что программное обеспечение является обучаемым в том случае, когда оно может совершенствовать свои действия с течением времени. Искусственный интеллект должен продемонстрировать интеллектуальный выбор

действия (логические рассуждения), поэтому программное обеспечение предпочтительно является способным исследовать свою окружающую среду (свой "окружающий мир") и найти возможности действий. Программное обеспечение предпочтительно также является способным представлять состояние окружающего мира и свое собственное внутреннее состояние. Следовательно, программное обеспечение предпочтительно является способным выбирать интеллектуальное действие (с использованием вышеупомянутых знаний) и действовать.

Обучение, например, посредством обучающегося модуля интерфейса может быть предпочтительно, но не обязательно, подкреплено поощрениями, которыми поощряют обучающийся модуль за то, что он предпринимает конкретные действия согласно состоянию окружающей среды. Этот тип обучения фактически подразумевает обучение обучающегося модуля определенной линии поведения. Если разрешено более одной линии поведения, то процесс обучения является недетерминированным и может создавать различные линии поведения. Что касается, например, интерфейса пользователя с упреждающим действием, то поощрение, предпочтительно, но не обязательно, содержит операцию, обуславливающую выявление обучающимся модулем того факта, когда предлагаемый вариант выбора приводит к его выбору пользователем, в противоположность тому случаю, когда предложенный вариант выбора приводит к тому, что пользователь ищет иной набор из одного или большего количества вариантов выбора, например, путем выбора иного меню, чем то, которое предложено интерфейсом пользователя с упреждающим действием. Очевидно, что интерфейс пользователя с упреждающим действием должен попытаться максимально увеличить процентную долю тех предложений, которые приводят к прямому выбору пользователем варианта выбора из этого предложения, поскольку это свидетельствует о том, что интерфейсом правильно понято поведение пользователя.

Для того чтобы способствовать этому процессу, обучение, осуществляемое обучающимся модулем, предпочтительно является обучением с пополнением, например, согласно приведенным ниже необязательным, но предпочтительным главным отличительным особенностям обучения с пополнением:

Процесс адаптивного обучения является итерационным процессом обучения, при этом при каждой итерации обучающийся модуль обучается выполнению надлежащего действия. Изменение окружающей среды предпочтительно приводит к изменениям поведения. Обучающийся модуль может быть обучен выполнению некоторых действий.

Низкий потребляемый объем памяти - алгоритмы системы, реализующей механизм логических рассуждений, такие как, например, нейронные сети или MAS, имеют запоминающие устройства невысокого уровня сложности, так как данные о состоянии окружающей среды и внутреннем состоянии сведены к небольшому набору численных значений. Во время процесса обучения алгоритм не требует большего объема памяти.

Быстрое взаимодействие - возможно, но не обязательно, при каждой итерации действие выбирают на основании вычисления предыдущей итерации, таким образом, для выбора следующего действия производят небольшой объем вычислений. Пользователь воспринимает программу как программу с быстрым откликом. После выполнения действия выполняют процедуру обучения для следующей итерации.

Использование времени простоя устройства - поскольку обучающийся модуль может, но не обязательно, обучаться самостоятельно, исходя из состояния окружающей среды, без взаимодействия с пользователем, то время простоя

вычислительного устройства предпочтительно может быть использовано для обучения.

На фиг.1 показана блок-схема примера обучающегося модуля согласно настоящему изобретению для обучения по реакции. Как показано на чертеже, обучающийся модуль 100 предпочтительно содержит базу 102 знаний, которая предпочтительно действует в качестве запоминающего устройства обучающегося модуля 100, храня информацию, собранную обучающимся модулем 100 в результате взаимодействий с окружающей средой. База 102 знаний может быть, но не обязательно, запомнена в энергонезависимой памяти (на чертеже не показана). В базе 102 знаний предпочтительно хранят информацию, которая помогает обучающемуся модулю 100 выбирать надлежащее действие. Эта информация может, но не обязательно, содержать такие значения, как, например, числовые весовые коэффициенты для внутренней нейронной сети или таблицы со значениями поощрений за действия, или информацию любого иного типа.

Для того чтобы обучающийся модуль 100 был способен получать информацию об окружающей среде, обучающийся модуль 100 предпочтительно содержит множество датчиков 104. Предпочтительно, датчики 104 позволяют обучающемуся модулю 100 воспринимать состояние окружающей его среды. Датчики 104 соединены с окружающей средой и осуществляют вывод измеренных значений. Источником значений может являться сама программа (например, положение на экране, уровень энергии и т.д.), или значения параметров реального устройства (например, величина заряда аккумулятора и рабочее состояние, например, состояние откидной панели (flipper) телефонных аппаратов сотовой связи, в которых приведение в действие устройства или ответ на входящий телефонный звонок могут быть произведены путем открывания "откидной панели").

Датчики 104, конечно же, предоставляют ценную информацию, однако, прежде чем обучающийся модуль 100 сможет понять эту информацию, она должна быть подвергнута обработке. Следовательно, обучающийся модуль 100 предпочтительно также содержит блок 106 восприятия, предназначенный для обработки текущих данных с выхода датчиков 104 с преобразованием в единообразное представление мира, именуемое "состоянием". Следовательно, состояние предпочтительно является входной информацией для системы 108, реализующей механизм логических рассуждений, которая может быть описана как "мозг" обучающегося модуля 100. Такая конструкция обеспечивает поддержку расширения состояния, отражающего окружающий мир, и механизма датчиков, а также обеспечивает поддержку простого подключения системы через порты к нескольким главным платформам (различными вычислительным устройствам и средам), что обеспечивает возможность изменения состояния, отражающего окружающий мир, в соответствии с устройством.

Система 108, реализующая механизм логических рассуждений, предпочтительно осуществляет обработку текущего состояния с использованием базы 102 знаний, вырабатывая, таким образом, решение о том, какое именно действие следует выполнять. Система 108, реализующая механизм логических рассуждений, получает сведения о текущем состоянии окружающего мира, осуществляет вывод данных действия, подлежащего выполнению, и получает ответную реакцию на выбранное действие. Система 108, реализующая механизм логических рассуждений, предпочтительно обновляет базу 102 знаний на основании ответной реакции. Этот процесс представляет собой итерационный процесс, в котором обучающийся модуль 100 обучается связывать действия с состояниями.

Согласно необязательному варианту осуществления настоящего изобретения вычислительное устройство может содержать один или большее количество биологических датчиков, предназначенных для регистрации различных типов биологической информации о пользователе, например, эмоционального состояния, физического состояния, движения и т.д. Эта информация может быть затем подана на датчики 104 для содействия определению состояния пользователя блоком 106 восприятия и, следовательно, для определения надлежащего состояния для устройства. Такие биологические датчики могут содержать датчики для определения температуры тела, частоты сердечбиений, степени насыщения кислородом или датчик любого иного типа, измеряющий биологические параметры пользователя, но эти примеры не являются ограничивающим признаком.

На фиг.2 показан приведенный в качестве примера вариант осуществления системы 200 согласно настоящему изобретению для обеспечения интерфейса пользователя с упреждающим действием, опять-таки содержащей обучающийся модуль 100. Обучающийся модуль 100 показан осуществляющим связь с операционной системой 202 вычислительного устройства (на чертеже не показано), с которым обучающийся модуль 100 связан, и/или которое осуществляет управление им и/или посредством которого приводят в действие обучающийся модуль 100. Операционная система 202 предпочтительно управляет функционированием блока 204 интерфейса, а также, по меньшей мере, одной другой прикладной программы 206 (хотя, конечно же, может иметься множество таких прикладных программ, но это не является обязательным).

Пользователь предпочтительно осуществляет обмен информацией посредством блока 204 интерфейса, например, путем выбора выбираемого пункта меню. Операционная система 202 обеспечивает возможность приема этой передаваемой информации и преобразования ее в данные. Затем обучающийся модуль 100 предпочтительно получает эти данные и, возможно, но не обязательно, передает обратно в операционную систему 202 команду, например, изменения какого-либо аспекта блока 204 интерфейса (например, путем предложения иного меню) и/или управления прикладной программой 206. Затем пользователь реагирует посредством блока 204 интерфейса, по этой реакции обучающийся модуль 100 предпочтительно узнает о том, было ли действие (команда, переданная обучающимся модулем 100) выполнено надлежащим образом.

На фиг.3 изображена блок-схема, на которой показан приведенный в качестве примера вариант реализации системы 300 интерфейса пользователя с упреждающим действием согласно настоящему изобретению. Как показано на чертеже, система 300 предпочтительно содержит трехуровневую архитектуру, при этом уровень приложений поддерживается структурой искусственного интеллекта (ИИ), которая, в свою очередь, осуществляет обмен информацией с вычислительным устройством главной платформы (показанным на чертеже как "главная платформа").

Уровень приложений предпочтительно, но не обязательно, характеризуется тем, что содержит множество различных приложений, из которых на чертеже показано лишь несколько не ограничивающих примеров, которыми являются, например, приложение 302 "мутация" (MutateApp), приложение 304 "возврат в предыдущее состояние" (PreviousApp) и обучающее приложение 306 (TeachingApp).

Приложение 302 "мутация" (MutateApp) предпочтительно вызывают для управления мутацией и/или инициирования мутаций в системе 300. Как указано выше, обучающийся модуль может, но не обязательно, изменять свое поведение посредством

управляемой или частично управляемой эволюции, например, посредством генетических алгоритмов. Приложение 302 "мутация" (MutateApp) предпочтительно осуществляет управление такими мутациями посредством эволюции и/или инициирует их. Более подробное описание варианта осуществления эволюции приведено ниже.

Приложение 304 "возврат в предыдущее состояние" (PreviousApp) предпочтительно дает возможность вызывать предыдущее состояние системы 300 или его часть (например, состояние обучающегося модуля) вместо текущего состояния. В частности, приложение 304 "возврат в предыдущее состояние" (PreviousApp) дает возможность пользователю возвращаться к предыдущей эволюционной операции в том случае, если настоящее изобретение реализовано посредством эволюционного алгоритма. В более широком толковании, система 300 предпочтительно является системой с запоминанием состояния и, следовательно, может по выбору возвращаться в предыдущее состояние, поскольку предыстория этих состояний предпочтительно является запомненной.

Более подробное описание обучающего приложения 306 (TeachingApp) приведено ниже после примера 3, но оно может быть, но не обязательно, реализовано таким образом, что обучает пользователя тому, каким образом следует управлять вычислительным устройством, и/или тому, каким образом следует обращаться с иным объектом, являющимся внешним по отношению к вычислительному устройству. Обучающее приложение 306 (TeachingApp) обеспечивает обучающее приложение, которое в совокупности с описанной ниже инфраструктурой искусственного интеллекта (ИИ) обеспечивает индивидуализированный опыт обучения. Обучающее приложение 306 (TeachingApp) предпочтительно может подстраивать тип обучения, методы обучения, темп сообщения новой информации, действия по пополнению, действия по практической реализации и т.д. таким образом, чтобы обеспечить удовлетворение индивидуальных потребностей конкретного пользователя. Кроме того, обучающее приложение 306 (TeachingApp) также может быть способным, но не обязательно, к адаптации своего функционирования для множества различных пользователей, например, в случае группового обучения или обучения в аудитории.

Обучающее приложение 306 (TeachingApp) является только лишь одним из не ограничивающих примеров типового приложения, которое может быть реализовано на уровне структуры искусственного интеллекта (ИИ).

Сам уровень структуры искусственного интеллекта (ИИ) содержит один или большее количество компонентов, которые обеспечивают возможность упреждающего поведения интерфейса пользователя. Эта структура предпочтительно, но не обязательно, содержит средство 308 отображения собственного мира устройства (DeviceWorldMapper), предназначенное для определения состояния вычислительного устройства и также состояния виртуального мира, а также зависимости между этими двумя состояниями. Для определения состояния виртуального мира и состояния устройства средство 308 отображения собственного мира устройства (DeviceWorldMapper) предпочтительно получает в качестве входных данных данные о различных событиях, поступающие из обработчика 310 событий (EventHandler).

Средство 308 отображения собственного мира устройства (DeviceWorldMapper) также предпочтительно осуществляет обмен информацией с модулем 312 искусственного интеллекта/обучения машины, ИИ/ОМ (AI/ML) для анализа входных данных. Модуль 312 ИИ/ОМ предпочтительно также определяет поведение системы 300 в ответ на различные стимулирующие воздействия, а также дает

возможность обучать систему 300, например, по реакции пользователя на различные типы действий интерфейса пользователя. Предпочтительно, но не обязательно, поведение системы 300 может быть также усовершенствовано согласно модулю 314 эволюции.

5 Вариант, в котором реализована эволюция, является особенно предпочтительным применительно к использованию интеллектуального агента в мобильном информационном устройстве (в качестве примера см. приведенное ниже описание), но также может быть, но не обязательно, использован в любом интерфейсе пользователя с упреждающим действием для вычислительного устройства. Этот вариант
10 осуществления предпочтительно используют в том случае, когда интерфейс пользователя с упреждающим действием также содержит виртуальный персонаж, или когда его используют в совокупности с виртуальным персонажем.

15 Эволюцию предпочтительно моделируют посредством набора генетических алгоритмов. Базис этих алгоритмов описывает свойства интерфейса с упреждающим действием (и, в частности, внешний вид виртуального персонажа), выраженные в терминах "гены", "хромосомы" и "фенотипы". Ген представляет собой дискретное свойство, которое имеет уровень выражения, например, нога некоторого типа.
20 Уровнем выражения может являться количество этих ног.

Фенотип представляет собой внешнее выражение гена; например, ген ноги может иметь различные фенотипы, выраженные как длина ноги или ее размер.

Ген может, но не обязательно, проходить через процесс мутации. Этот процесс
25 (предпочтительно согласно определенной вероятности) изменяет один или большее количество параметров гена, создавая, тем самым, различные новые фенотипы.

Хромосома представляет собой набор генов, функционирующих совместно. Хромосома может создавать гибриды (скрещиваться) с хромосомой того же самого типа от иного виртуального существа, создавая, таким образом, новую хромосому,
30 которая представляет собой комбинацию хромосом ее генетических прародителей.

Эта методология помогает в создании общей инфраструктуры для моделирования визуальной эволюции (например, внешнего вида виртуального персонажа) и/или эволюции поведения интерфейса пользователя с упреждающим действием. Эти алгоритмы также могут быть, но не обязательно, использованы для определения
35 характеристик поведения, не воспринимаемых зрением, например, ловкости, выносливости и т.д. Этот эффект может, но не обязательно, приводить, например, к созданию более быстрого виртуального существа или более эффективного виртуального существа. Эти алгоритмы могут быть, но не обязательно, использованы
40 для любых подобных характеристик, которые могут быть описаны согласно вышеупомянутой структуре ген/генотип/фенотип, поэтому, например, поведенческие гены могут, но не обязательно, определять поведение алгоритмов искусственного интеллекта (ИИ), используемых в настоящем изобретении.

Данные, получаемые на выходе алгоритма, предпочтительно обеспечивают
45 множество возможных потомков виртуальных персонажей и/или интерфейсов пользователя с упреждающим действием.

Генетические алгоритмы используют процесс естественного отбора для принятия решения о том, какие именно из генетических потомков будут продолжать
50 существование в качестве следующего поколения. Решение относительно процесса отбора может быть принято пользователем или может быть заранее заданным. Таким образом, виртуальное существо может отображать интересное эволюционное поведение. Для эволюции генов, кодирующих иные не воспринимаемые зрением

свойства виртуального существа, например, целевые установки или характер, может быть использована структура типового алгоритма.

Модуль 314 эволюции обеспечивает поддержку такой эволюции, а также предпочтительно управляет ею, например, посредством действия приложения 302 "мутация" (MutateApp).

Между этими различными приложениями искусственного интеллекта (ИИ) и обработчиком 310 событий (EventHandler) один или большее количество различных администраторов низкого уровня предпочтительно обеспечивают поддержку получения сведений о различных событиях и их обработку, а также выполнения различных действий системой 300. Эти администраторы могут, но не обязательно, содержать администратор 316 действий (ActionManager), администратор 318 интерфейса пользователя (UIManager), администратор 320 хранения данных (StorageManager) и администратор 322 приложений (ApplicationManager), а эти примеры не являются ограничивающим признаком.

Более подробное описание администратора 316 действий (ActionManager) приведено ниже, но в кратком изложении, он предпочтительно обеспечивает для системы 300 возможность определения, например, посредством операций, выполняемых модулем 312 ИИ/ОМ (искусственного интеллекта/обучения машины), того, какое именно действие следует произвести.

Администратор 318 интерфейса пользователя (UIManager) предпочтительно управляет внешним видом и функциями интерфейса пользователя, например, управляя вышеописанными изменениями этого интерфейса.

Администратор 320 хранения данных (StorageManager) предпочтительно управляет хранением и обработкой данных, например, для базы знаний системы 300 (на чертеже не показана).

Администратор 322 приложений (ApplicationManager) предпочтительно осуществляет обработку данных, обмен которыми происходит с вышеописанными приложениями на уровне приложений.

Все эти различные управляющие программы предпочтительно получают сведения о событиях из обработчика 310 событий (EventHandler).

В пределах уровня структуры искусственного интеллекта (ИИ) инфраструктура 324 искусственного интеллекта (ИИ) предпочтительно, но не обязательно, обеспечивает поддержку обмена информацией с главной платформой. Сама главная платформа предпочтительно содержит интерфейс 326 главной платформы, который предпочтительно, но не обязательно, может быть обеспечен, например, посредством операционной системы главной платформы.

Инфраструктура 324 искусственного интеллекта (ИИ) предпочтительно, но не обязательно, содержит модуль 328 ввода-вывода, предназначенный для приема входных данных из интерфейса 326 главной платформы, а также, но не обязательно, для передачи команд в интерфейс 326 главной платформы. Модуль 330 экрана предпочтительно осуществляет обработку, связанную с отображением интерфейса пользователя на экране вычислительного устройства главной платформы. Модуль 332 ресурсов предпочтительно обеспечивает для системы 300 возможность доступа к различным ресурсам главной платформы, например, к запоминающему устройству для данных и т.д.

Конечно же, на описанных выше чертежах представлен только один возможный вариант конфигурации обучающегося модуля. Например, обучающийся модуль может быть также представлен в виде набора отдельных агентов, в котором каждый агент

имеет простую целевую установку. Обучающийся модуль производит выбор агента для выполнения действия на основании текущего состояния. Обучающийся модуль также может быть обучен надлежащему соответствию между текущим состоянием и агентами путем обучения с пополнением.

Также может производиться необязательный административный контроль обучения. Обучающийся модуль может хранить набор примеров о том, какой линии поведения следует придерживаться, а затем может быть обучен шаблону, полученному от администратора.

После того, как обучающийся модуль обучен правилам, он пытается действовать на основании той информации, которую он уже видел, и обобщать новые состояния.

ПРИМЕР 2: АДАПТИВНАЯ СИСТЕМА ДЛЯ МОБИЛЬНОГО ИНФОРМАЦИОННОГО УСТРОЙСТВА

Этот пример относится к иллюстративному варианту реализации адаптивной системы из настоящего изобретения в мобильном информационном устройстве, хотя следует понимать, что этот вариант реализации является предпочтительным, но необязательным, и при этом подразумевают, что он никоим образом не является ограничивающим.

Адаптивная система может, но не обязательно, содержать любую из функциональных возможностей, описанных выше в Примере 1, и может быть также, но не обязательно, реализована описанным выше способом. Этот пример сосредоточен в большей степени на реальной архитектуре адаптивной системы применительно к функционированию мобильного информационного устройства.

Кроме того, в этом примере описан необязательный, но предпочтительный вариант реализации виртуального существа или виртуального персонажа согласно настоящему изобретению.

В последующих разделах описаны необязательные, но предпочтительные варианты осуществления конкретных вариантов технической реализации различных аспектов адаптивной системы согласно настоящему изобретению. Подразумевая, что эти варианты осуществления приведены только лишь в описательных целях и никоим образом не являются ограничивающим признаком, эти варианты осуществления основаны на необязательном, но предпочтительном варианте осуществления адаптивной системы, взаимодействующей с пользователем посредством интеллектуального агента, возможно, но не обязательно, визуально отображенного в виде виртуального персонажа или "виртуального существа".

Раздел 1: Система, управляемая событиями

В этом разделе приведено описание предпочтительного варианта осуществления системы, управляемой событиями, согласно настоящему изобретению, в том числе, описание программы управления приложениями и взаимодействий между самим устройством и системой из настоящего изобретения, находящейся под управлением устройства.

На фиг.4 показана блок-схема адаптивной системы 400 согласно настоящему изобретению, которая приведена в качестве примера, и показаны взаимодействия системы 400 с мобильным информационным устройством 402. К тому же, как показано на чертеже, как система 400, так и мобильное информационное устройство 402 предпочтительно взаимодействуют с пользователем 404.

Мобильное информационное устройство 402 предпочтительно, но не обязательно, имеет несколько стандартных функций, которые показаны разделенными на две категории только лишь для пояснения: данные и средства (механизмы). Средства

могут, но не обязательно, содержать такие функции (функциональные средства), как система 406 интерфейса пользователя (ИП) (экран, средство ввода в виде кнопочной панели или сенсорного экрана и т.д.); функция 408 входящих и исходящих телефонных звонков; функция 410 обмена сообщениями, например, для обмена короткими сообщениями (SMS); функции звука 412 и/или вибрации 414 для предупреждения пользователя 404 о входящем телефонном вызове или сообщении, и/или о будильнике и т.д.; и функция 416 хранения данных, но эти примеры не являются ограничивающим признаком.

Данные могут, но не обязательно, содержать такую информацию как адресная (телефонная) книга 418, информация 420 о входящих или исходящих телефонных звонках, данные о местоположении мобильного информационного устройства 402, показанные как местоположение 422, информация 424 о сообщениях, данные 426 из сети Интернет, находящиеся в кэш-памяти, и данные о пользователе 404, показанные как данные 428 о владельце.

Следует отметить, что мобильное информационное устройство 402 может, но не обязательно, содержать любые одни или большее количество вышеупомянутых данных/средств, но не обязательно должно содержать все из них, и/или может содержать дополнительные данные/средства, которые не показаны на чертеже. Подразумевают, что они просто приведены в качестве примеров, не являющихся ограничивающим признаком, которые относятся к мобильному информационному устройству 402, в частности, к телефонным аппаратам сотовой связи.

Адаптивная система 400 согласно настоящему изобретению предпочтительно взаимодействует с данными/средствами мобильного информационного устройства 402 для того, чтобы она была способна обеспечивать адаптивный интерфейс пользователя (а также предпочтительно интерфейс пользователя с упреждающим действием), посредством чего обеспечивают еще большее упрощение взаимодействия пользователя 404 с мобильным информационным устройством 402 и более высокую эффективность такого взаимодействия.

Адаптивная система 400 предпочтительно содержит логику 430, которая в предпочтительном варианте функционирует способом, подобным способу функционирования вышеописанного обучающегося модуля, и которая предпочтительно, но не обязательно, также функционирует согласно вышеописанным алгоритмам искусственного интеллекта (ИИ) и обучения машины.

Логика 430 предпочтительно является способной осуществлять обмен информацией с базой 102 знаний так, как изложено в описании фиг.1 (компоненты, обозначенные одинаковыми номерами позиций, имеют либо идентичные, либо аналогичные функциональные возможности, если не указано иное). Хранилище 432 информации предпочтительно содержит данные о действиях мобильного информационного устройства 402, информацию пользователя и т.д., и предпочтительно дополняет данные, имеющиеся в базе 102 знаний.

Адаптивная система 400 предпочтительно является способной к эволюции посредством эволюционной логики 434, которая может, но не обязательно, объединять вышеописанные функциональные возможности модуля 314 эволюции и приложения 302 "мутация" (MutateApp) из фиг.3 (которое на этом чертеже не показано).

Адаптивная система 400 возможно, но не обязательно, способна осуществлять непосредственный обмен информацией с пользователем 404 посредством текстового и/или слышимого языка, поддержку которого осуществляет языковый модуль 436.

В частности, как изложено в приведенном ниже описании варианта осуществления

настоящего изобретения в Пимере 3, но что также возможно и для адаптивной системы 400, пользователю 404 может быть, но не обязательно, предоставлен виртуальный персонаж (на чертеже не показан) для интерфейса пользователя. В случае наличия такого виртуального персонажа, он может быть, но не обязательно, создан посредством трехмерной графической модели 438 и анимационного модуля 440 (более подробное описание приведено ниже). Виртуальный персонаж может быть, но не обязательно, индивидуализирован для пользователя 404, посредством чего обеспечивают более сильные эмоциональные переживания для пользователя 404 при его взаимодействии с мобильным информационным устройством 402.

На фиг.5А показана блок-схема примера системы 500 управления приложениями, которая является основной инфраструктурой, обеспечивающей поддержку адаптивной системы из настоящего изобретения. Система 500 также может быть, но не обязательно, использована для обеспечения поддержки таких вариантов осуществления, как функциональные возможности обучающего приложения, описание которых приведено выше, а ниже также приведено более подробное их описание. Система 500 предпочтительно содержит администратор 502 приложений для управления приложениями различных типов, которые являются частью адаптивной системы согласно настоящему изобретению. Администратор 502 приложений осуществляет обмен информацией с интерфейсом приложений, именуемым базовым приложением 504 (BaseApp), который реализован всеми приложениями в системе 500. И администратор 502 приложений, и базовое приложение 504 (BaseApp) осуществляют обмен информацией о событиях посредством обработчика 506 событий (EventHandler).

Администратор 502 приложений является ответственным за управление и за предоставление времени выполнения для выполнения системных приложений (приложений, являющихся частью системы 500). Срок действия каждого такого приложения определен в базовом приложении 504 (BaseApp), которое разрешает администратору 502 приложений запускать, приостанавливать, возобновлять выполнение каждого такого приложения и выходить из него (прекращать выполнение). Администратор 502 приложений предпочтительно управляет использованием времени выполнения (рабочим циклом) программы пошаговым способом, который обеспечивает интерфейс базового приложения 504 (BaseApp). Следует отметить, что для выполнения предпочтительно, но не обязательно, используют пошаговый способ, так как система 500 предпочтительно является системой с запоминанием состояния, поэтому каждый шаг предпочтительно соответствует (приблизительно) одному или большему количеству состояний. Однако выполнение также может быть, но не обязательно, основано на цепочке выполняемых задач и/или на способе выполнения любого типа.

Администратор 502 приложений получает из мобильного информационного устройства данные о событии таймера. Как более подробно описано ниже, предпочтительно мобильное информационное устройство содержит такую операционную систему, что данные о событии таймера предпочтительно получают из уровня операционной системы. Когда активизирован таймер, администратор 502 приложений вызывает операцию выполнения текущего приложения. Когда пользователь активизирует иное приложение, например, когда он использует систему меню, то администратор 502 приложений предпочтительно переключается с одного приложения на другое приложение.

На чертеже показаны некоторые примеры системных приложений, не являющиеся ограничивающими, в том числе приложение 508 "обучающая машина"

(TeachingMachineApp), приложение 510 "мутация" (MutateApp), приложение 514 "мастерская генов" (GeneStudioApp), приложение 516 "мастер обучения" (TWizardApp), приложение 518 "плавающий агент" (FloatingAgentApp), приложение 522 "мир обучаемого виртуального существа" (TCWorldApp) и приложение 520 "гибридизация" (HybridApp). Более подробное описание этих приложений также приведено ниже в примере 3.

Приложение 510 "мутация" (MutateApp) предпочтительно вызывают для управления мутацией и/или для инициирования мутации в адаптивной системе, и/или во внешнем виде виртуального персонажа, представляющего адаптивную систему в виде интерфейса пользователя. Как упомянуто выше при описании примера 1, адаптивная система из настоящего изобретения может, но не обязательно, изменять свое поведение посредством управляемой или частично управляемой эволюции, например, посредством генетических алгоритмов. Приложение 510 "мутация" (MutateApp) предпочтительно осуществляет управление такими мутациями и/или иницирует их.

Приложение 514 "мастерская генов" (GeneStudioApp) предпочтительно предоставляет пользователю возможность выполнять управляемые или частично управляемые мутации посредством одной или большего количества команд, подаваемых вручную. Например, пользователь может пожелать осуществить управление адаптивной системой (посредством системы 500 управления приложениями) таким образом, чтобы выполнить конкретную последовательность задач после получения конкретных введенных данных. В альтернативном варианте пользователь может пожелать непосредственно изменить часть внешнего вида виртуального персонажа при его наличии. Согласно предпочтительным вариантам осуществления настоящего изобретения эти различные аспекты адаптивной системы предпочтительно реализованы посредством отдельных "генов", которые затем могут быть, но не обязательно, изменены пользователем.

Приложение 520 "гибридизация" (HybridApp) может быть, но не обязательно, вызвано в том случае, если пользователь желает получить информацию из внешнего источника, например, из адаптивной системы другого мобильного информационного устройства, и объединить эту информацию с существующей информацией в мобильном информационном устройстве пользователя. Например, пользователь может захотеть создать виртуальный персонаж, имеющий гибридный внешний вид с виртуальным персонажем другого мобильного информационного устройства. Приложение 520 "гибридизация" (HybridApp) предпочтительно, но не обязательно, также обеспечивает основное управляющее воздействие пользователя на все состояние эволюции виртуального персонажа. Более предпочтительно, но не обязательно, приложение 520 "гибридизация" (HybridApp) может быть использовано для инструктирования пользователя о свойствах "жизни" виртуального персонажа, который может, но не обязательно, иметь имя, индивидуальные особенности, поведение и внешний вид.

Приложение 508 "обучающая машина" (TeachingMachineApp) является иллюстративным, не ограничивающим примером приложения, которое может быть, но не обязательно, связано с предоставлением инструкций по использованию самого устройства, но предпочтительно оно предоставляет инструкцию относительно предмета, который непосредственно не связан с работой самого устройства. Следовательно, приложение 508 "обучающая машина" (TeachingMachineApp) представляет собой необязательный пример приложения, предусмотренного в мобильном информационном устройстве для иной цели, нежели использование самого

устройства.

Приложение 522 "мир обучаемого виртуального существа" (TCWorldApp) представляет собой приложение, запускающее интеллектуального агента, которое предпочтительно управляет как интеллектуальными особенностями агента, так и графическим отображением виртуального существа или виртуального персонажа (более подробное описание обеих этих функций приведено ниже).

Другими не ограничивающими примерами приложений согласно настоящему изобретению являются, в том числе, игры. Одним из не ограничивающих примеров игры, в которой адаптивная система и пользователь могут, но не обязательно, взаимодействовать друг с другом, является "игра в прятки". "Игру в прятки" предпочтительно проводят следующим образом: виртуальное существо или виртуальный персонаж "прячется" в иерархии меню, поэтому пользователь предпочтительно проходит, по меньшей мере, через одно подменю для того, чтобы найти виртуальный персонаж или существо, посредством чего вызывают получение пользователем большего объема знаний об иерархии и структуре меню. Возможно множество иных подобных игровых приложений, что не выходит за пределы объема патентных притязаний настоящего изобретения.

Приложение 516 "мастер обучения" (TWizardApp) представляет собой приложение другого типа, которое предоставляет информацию пользователю. Оно описано применительно к приложению "мастер начала работы" (Start Wizard) в приведенном ниже примере 4. В кратком изложении, это приложение содержит предпочтения пользователя и конфигурацию структуры искусственного интеллекта (ИИ), например, персонаж интеллектуального агента, в частности, для системы формирования эмоций (более подробное описание которой также приведено ниже), а также для задания приоритетов целевых установок (более подробное описание которых приведено ниже).

Приложение 518 "плавающий агент" (FloatingAgentApp) предпочтительно, но не обязательно, осуществляет управление внешним видом интерфейса пользователя, в частности, тем, что связано с внешним видом виртуального персонажа (при его наличии). Приложение 518 "плавающий агент" (FloatingAgentApp) позволяет отображать на дисплее визуальные особенности изображения интерфейса пользователя независимо от отображения виртуального персонажа, который, следовательно, может выглядеть, например, как "плавающий" по интерфейсу пользователя. Приложение 518 "плавающий агент" (FloatingAgentApp) предпочтительно представляет собой приложение, используемое по умолчанию, которое запускают тогда, когда не выполняется никакое иное приложение.

На фиг.5В показан пример схемы последовательности операций, выполняемых в администраторе приложений согласно настоящему изобретению. Как показано на чертеже, обработчик 506 событий (EventHandler) предпочтительно посылает уведомление о событии в администратор 502 приложений, что обозначено стрелкой 1. Если событие является событием таймера, то администратор 502 приложений вызывает этап запуска (действие) экземпляра соответствующего приложения, которое уже было вызвано, что обозначено стрелкой 1.1.1. Если событие должно инициировать выполнение приложения, то администратор 502 приложений вызывает экземпляр соответствующего приложения, что обозначено стрелкой 1.2.1. Если выполняющийся в настоящее время экземпляр приложения должен быть приостановлен, то администратор 502 приложений передает в приложение команду приостановки, что обозначено стрелкой 1.3.1. Если выполнение ранее приостановленного экземпляра приложения должно быть возобновлено, то администратор 502 приложений передает

в приложение команду "возобновить", что обозначено стрелкой 1.4.1. В любом случае данные об успешном выполнении этапа возвращаются в администратор 502 приложений, что показано соответствующими обратными стрелками для вышеуказанных стрелок. Затем администратор 502 приложений уведомляет
 5 обработчика 506 событий (EventHandler) об успешном выполнении или в противном случае - о неудаче.

Эти различные приложения важны для того, чтобы обеспечить для адаптивной системы возможность управления различными аспектами функционирования
 10 мобильного информационного устройства. Однако адаптивная система также должна быть способна осуществлять обмен информацией напрямую с различными компонентами мобильного информационного устройства посредством операционной системы мобильного информационного устройства. Такой обмен информацией может быть, но не обязательно, выполнен посредством системы 600 связи, показанной на
 15 фиг.6, предпочтительно с использованием описанных ниже алгоритмов действия.

На фиг.6А и фиг.6В показан пример варианта реализации инфраструктуры, требуемой для выполнения адаптивной системой согласно настоящему изобретению одного или большего количества действий посредством операционной системы
 20 мобильного информационного устройства (фиг.6А), а также схема последовательности операций для функционирования системы 600 связи (фиг.6В). Согласно необязательным, но предпочтительным вариантам осуществления настоящего изобретения, эта инфраструктура представляет собой пример более общего понятия "упаковщики искусственного интеллекта (ИИ)" или способности
 25 "объединения" существующей системы интерфейса пользователя (ИП) с новыми возможностями искусственного интеллекта (ИИ) и обучения машин.

Система 600 связи предпочтительно является способной осуществлять обработку событий различных типов с событием 602 базового класса, производя обмен
 30 информацией с обработчиком 506 событий (EventHandler), как описано выше. Затем диспетчер 604 событий (EventDispatcher) осуществляет маршрутизацию события в надлежащий объект в пределах системы из настоящего изобретения. Маршрутизация предпочтительно определяется регистрацией объекта в диспетчере 604 событий (EventDispatcher) для конкретного события. Диспетчер 604
 35 событий (EventDispatcher) предпочтительно управляет реестром обработчиков, реализующих интерфейс с обработчиком 506 событий (EventHandler) для такого уведомления.

Конкретные события, для которых реализованы конкретные обработчики, предпочтительно, но необязательно, содержат обработчик 606 события, связанного с
 40 откидной панелью, для телефонных аппаратов сотовой связи, в которых приведение в действие устройства или ответ на входящий телефонный звонок могут быть произведены путем открывания "откидной панели"; это событие происходит при открывании или закрывании откидной панели. Приложения, управление которыми
 45 осуществляют согласно настоящему изобретению, могут, но не обязательно, передавать друг другу информацию о событиях, обработку которой предпочтительно осуществляет обработчик 608 событий, передаваемых между приложениями (InterAppEvent). Обработку события, связанного с необязательной, но
 50 предпочтительной эволюцией (изменением) виртуального существа или виртуального персонажа, предпочтительно осуществляет обработчик 610 события, связанного с эволюцией (EvolutionEvent). Обработку входящего или исходящего телефонного вызова предпочтительно осуществляет обработчик 612 события, связанного с

телефонным вызовом, (CallEvent), который, в свою очередь, предпочтительно содержит два дополнительных обработчика, которыми являются обработчик 614 события "вызов начат" (CallStartedEvent), предназначенный для того, чтобы начать телефонный вызов, и обработчик 616 события "вызов закончен" (CallEndedEvent),
 5 предназначенный для завершения телефонного вызова.

Обработку события, связанного с SMS-сообщением (входящим или исходящим сообщением), предпочтительно осуществляет обработчик 618 события, связанного с SMS-сообщением (SMSEvent). Необязательными, но предпочтительными
 10 параметрами, которые могут быть включены в состав информации о событии, являются, в том числе, параметры, относящиеся к гибридизации виртуального существа или виртуального персонажа одного мобильного информационного устройства с виртуальным существом или виртуальным персонажем другого мобильного информационного устройства, а их более подробное описание приведено
 15 ниже.

Обработку событий, связанных с приведением в действие клавиш, предпочтительно осуществляет обработчик 620 событий, связанных с клавишами, (KeyEvent) и/или обработчик 622 событий, связанных с кодами клавиш (KeyCodeEvent). Например, если
 20 пользователь нажимает клавишу на мобильном информационном устройстве, обработчик 620 событий, связанных с клавишами, (KeyEvent) предпочтительно осуществляет обработку этого события, которое связано с вводимой информацией для функционирования системы согласно настоящему изобретению. В схеме последовательности операций объектом key_event (событие, связанное с клавишами)
 25 является объект из класса "события, связанные с клавишами" (KeyEvent), который представляет собой объект сообщения о событии, связанном с клавишами. Обработчик 620 событий, связанных с клавишами, (KeyEvent) сам осуществляет обработку объекта key_event, в то время как обработчик 622 событий, связанных с кодами клавиш, (KeyCodeEvent) отслеживает вводимый код (оба события ввода
 30 данных получены посредством обработчика прерываний (hook), внедренного в операционную систему).

Обработчик 624 событий, связанных с аккумулятором, (BatteryEvent) предпочтительно осуществляет обработку событий, связанных с аккумулятором,
 35 например, события "аккумулятор разряжен", или в альтернативном варианте производит переключение из режима низкого энергопотребления в режим высокого энергопотребления.

Обработчик 626 событий, связанных с временем суток, (DayTimeEvent)
 40 предпочтительно обрабатывает события, связанные с будильником, календарем или ежедневником для напоминаний/назначенных встреч.

На фиг.6В изображен пример схемы последовательности операций, на которой показано то, каким образом осуществляют обработку событий между операционной системой мобильного информационного устройства или иной управляющей
 45 структурой и системой из настоящего изобретения. В этом примере мобильное информационное устройство имеет операционную систему, хотя подобная последовательность операций может быть при желании реализована для устройств, в которых такая операционная система отсутствует. Операционная система, при ее
 50 наличии, осуществляет обработку информации, вводимой в устройство/выводимой из него, и управляет состоянием и событиями, происходящими с устройством. Показанная на фиг.6В схема последовательности операций представляет собой абстракцию для облегчения обработки этих событий и установления связи с ними.

Модуль 628 операционной системы (os_module) вызывает событие или устанавливает связь с событием; может, но не обязательно, существовать множество таких модулей, но для доходчивости показан только один такой модуль, и при этом подразумевают, что это никоим образом не является ограничивающим признаком.

Модуль 628 операционной системы является частью операционной системы мобильного информационного устройства. Модуль 628 операционной системы предпочтительно посылает уведомление о событии, принятом или созданном модулем 628 операционной системы, в обработчик прерываний (hook) 630.

Обработчик прерываний 630 представляет собой часть системы согласно настоящему изобретению и используется для выдачи разрешения на обмен информацией между операционной системой и системой согласно настоящему изобретению. Обработчик прерываний 630 отслеживает соответствующие события, исходящие из операционной системы. Обработчик прерываний 630 способен интерпретировать событие, исходящее из операционной системы, и встраивать данные о событии в сообщение, которое является понятным для события 602. Обработчик прерываний 630 также передает данные о событии диспетчеру 604 событий (EventDispatcher), который осуществляет обмен информацией с каждым обработчиком события, показанным на чертеже как обработчик 506 событий (EventHandler) (хотя может существовать множество таких обработчиков). Затем диспетчер 604 событий (EventDispatcher) посылает отчет в обработчик прерываний 630, который посылает отчет об обработке события в модуль 628 операционной системы.

На фиг.7А-фиг.7С показаны примеры событий и то, каким образом осуществляют их обработку путем взаимодействий между мобильным информационным устройством (посредством операционной системы устройства) и системой из настоящего изобретения. Следует отметить, что обработка некоторых событий может быть, но не обязательно, произведена в пределах системы из настоящего изобретения без обращения к мобильному информационному устройству.

На фиг.7А показан пример схемы последовательности операций для события, связанного с клавишами, описанной применительно к мобильному информационному устройству, которое имеет инфраструктуру операционной системы DMSS фирмы "Квэлкомм Инкорпорейтед" (Qualcomm Inc.), для их платформы для подвижных станций системы множественного доступа с кодовым разделением (МДКР, CDMA) типа MSM (конечный автомат с обменом сообщениями). Эта операционная система обеспечивает обслуживание операционной системы, например, обслуживание интерфейса пользователя, обслуживание ввода-вывода и интерактивного ввода данных с использованием клавиш (кнопочной панели) телефонного аппарата. Этот пример показывает, каким образом осуществляют генерацию события ввода данных посредством клавиши и его обработку системой из настоящего изобретения. Передачу данных о других событиях в систему осуществляют почти тем же самым способом, хотя функция обработчика прерываний 630 изменяется в соответствии с тем, какой именно модуль операционной системы передает данные о событии; предпочтительно, но не обязательно, имеется множество таких обработчиков прерываний, так что каждый обработчик прерываний имеет различную функцию относительно взаимодействия с операционной системой.

Как показано на фиг.7А, модуль 700 ui_do_event (создание события интерфейсом пользователя) представляет собой компонент операционной системы и его вызов производят периодически. При нажатии клавиши на мобильном устройстве структура интерфейса пользователя (ИП, UI), которая осуществляет передачу информации в

модуль 700 `ui_do_event`, содержит значение клавиши. Затем обработчик прерываний 630 получает значение клавиши, предпочтительно, но не обязательно, идентифицирует событие как событие, связанное с клавишами (в частности, если модуль 700 `ui_do_event` передает данные о глобальном событии), и осуществляет генерацию события 702, связанного с клавишами. Затем данные о событии 702, связанном с клавишами, передают в диспетчер 604 событий (`EventDispatcher`). Затем данные о событии передают в приложение 704, которое выдало запрос на получение уведомления о таком событии, предпочтительно посредством обработчика событий (на чертеже не показан), как описано выше. Затем уведомление об успешном (или неуспешном) выполнении обработки события предпочтительно возвращают в диспетчер 604 событий (`EventDispatcher`) и из него - в обработчик прерываний 630 и в модуль 700 `ui_do_event`.

На фиг.7В показан второй иллюстративный пример схемы последовательности операций для обработки события; в этом варианте данные о событии передают из системы согласно настоящему изобретению в операционную систему и связывают с изображением на экране мобильного информационного устройства. Информацию пропускают через метод доступа к экрану, имеющийся в операционной системе, в котором экран (обычно) представлен буфером кадров. Буфер кадров представляет собой сегмент памяти, который копируют с использованием драйвера экрана (драйвера для аппаратных средств экрана) и отображают посредством экрана. Система из настоящего изобретения создает для операционной системы необходимую информацию для управления созданием изображений на экране.

Переходя теперь к рассмотрению фиг.7В, операционная система (посредством модуля 710 `scrn_update_main` (основного модуля обновления экрана)) сначала производит обновление буфера кадров для экрана, что обозначено стрелкой "1". Это обновление может, но не обязательно, содержать, например, операцию вывода рисунка фона, который может быть отображен в каждой части экрана, в которой не изображена информация, предоставленная системой из настоящего изобретения. Наличие такого фона возможно, но не обязательно, обеспечивает поддержку использования полупрозрачных окон, которые предпочтительно, но не обязательно, могут быть использованы для виртуального существа или агента, а их более подробное описание приведено ниже.

Затем модуль 710 `scrn_update_main` передает запрос на получение обновленных данных в модуль 712 экрана, который является частью системы из настоящего изобретения и который содержит обработчик прерываний для обмена информацией с операционной системой. Затем модуль 712 экрана передает в каждое окно приложения, показанное как агент 714 "окно" (`agentWindow`), причем может, но не обязательно, существовать множество таких окон, запрос для обновления информации о том, какое изображение должно быть выведено на экран. Если произошло такое изменение, что требуется обновление, то агент 714 "окно" (`agentWindow`) уведомляет модуль 712 экрана о том, что требуется обновление. Затем модуль 712 экрана выдает запрос на получение данных о местоположении и размере измененного участка предпочтительно в виде двух отдельных запросов (показанных, соответственно, как стрелки 2.1.2.1 и 2.1.2.2), ответы на которые передает агент 714 "окно" (`agentWindow`).

Модуль 712 экрана возвращает информацию в операционную систему посредством модуля 710 `scrn_update_main` в виде обновленного прямоугольника предпочтительно следующим образом. Модуль 710 `scrn_update_main` отвечает на уведомление о наличии

обновления путем копирования буфера кадров в предварительный буфер (операция 3.1). Затем модуль 712 экрана вводит изменения для каждого окна в предварительный буфер, что обозначено стрелкой 3.2.1. Затем содержимое предварительного буфера копируют в буфер кадров и из него на экран (стрелка 3.3).

На фиг.7С показана архитектура классов для системы из настоящего изобретения, предназначенная для вывода изображений на экран. Показаны оба элемента: модуль 712 экрана и агент 714 "окно" (agentWindow). Класс "агент 714 "окно"" (agentWindow) также осуществляет обмен информацией с тремя другими классами окон, которые предоставляют информацию об обновлении (изменении) окон: класс 716 "окно фона экрана" (BackScreenWindow), класс 718 "буферизованное окно" (BufferedWindow) и класс 720 "окно прямого доступа" (DirectAccessWindow). Класс 718 "буферизованное окно" (BufferedWindow) имеет два дополнительных класса окон, с которыми он осуществляет обмен информацией: класс 722 "окно с транс-буферизацией" (TransBufferedWindow) и класс 724 "окно с предварительной буферизацией" (PreBufferedWindow).

Раздел 2: Система выбора действия

В этом разделе приведено описание предпочтительного варианта осуществления системы выбора действия согласно настоящему изобретению, в том числе описание выбора возможного действия согласно наличию стимула/препятствия (стимулов/препятствий) и т.д. Для помощи в объяснении того, каким образом производят выбор действий интеллектуального агента, сначала приведено объяснение структуры интеллектуального агента и взаимодействий интеллектуального агента с виртуальной окружающей средой, которую предпочтительно обеспечивает система из настоящего изобретения.

На фиг.8 изображен пример структуры интеллектуального агента (фиг.8А), а также приведен пример схемы последовательности операций функционирования интеллектуального агента (фиг.8В). Как показано на фиг.8А, интеллектуальный агент 800 предпочтительно содержит множество классов. Основным классом является класс 802 "виртуальное существо, созданное посредством искусственного интеллекта (ИИ)" (AICreature), который содержит информацию об интеллектуальном агенте, например, о его состоянии, индивидуальных особенностях, целевых установках и т.д., а также информацию о внешнем виде виртуального существа, которое визуально отображает агента, например, о месторасположении, цвете, о том, является ли оно видимым в текущий момент времени, и т.д.

Класс 802 "виртуальное существо, созданное посредством искусственного интеллекта (ИИ)" (AICreature) осуществляет обмен информацией с классом 804 "виртуальный мир" (World), который является базовым классом виртуальной окружающей среды для интеллектуального агента. В свою очередь, класс 804 "виртуальный мир" (World) осуществляет обмен информацией с классами, составляющими виртуальную окружающую среду, из которых показаны лишь некоторые примеры, не являющиеся ограничивающим признаком. Класс 804 "виртуальный мир" (World) предпочтительно осуществляет обмен информацией с различными экземплярами объекта 806 класса "виртуальный мир" (WorldObject), который представляет собой объект, имеющийся в виртуальной среде, и с которым может взаимодействовать интеллектуальный агент. Класс 804 "виртуальный мир" (World) управляет этими различными объектами, а также принимает информацию об их характеристиках, в том числе об их свойствах, например, о местоположении и т.д. Класс 804 "виртуальный мир" (World) также управляет свойствами самой виртуальной

среды, например, размером, видимостью и т.д. Для визуального представления объекта 806 класса "виртуальный мир" (WorldObject) может быть, но не обязательно, использована двумерная или трехмерная графика, либо смесь этих вариантов графики, а также могут быть, но не обязательно, использованы иные возможности

5 мобильного информационного устройства, например, создание звуков и т.д.

Сам объект 806 класса "виртуальный мир" (WorldObject) может, но не обязательно, представлять собой объект, принадлежащий к одному из нескольких классов. Эта абстракция позволяет добавлять различные классы объектов в виртуальную среду или

10 удалять из нее. Например, объектом может, но не обязательно, являться "мяч", который, например, в исходном состоянии может быть представлен как часть меню, а затем "удален" виртуальным существом для того, чтобы играть с ним, что представлено объектом 808 "мяч из меню" (MenuBallObject). Объект 810 "доброе животное" (GoodAnimalObject) предпочтительно также производит обмен

15 информацией с объектом 806 класса "виртуальный мир" (WorldObject); в свою очередь, такие классы, как, например, объект 812 "пища" (FoodObject) (отображающий пищу для виртуального существа), объект 814 "злое животное" (BadAnimalObject) (животное, которое может раздражать виртуальное существо и, например, вызывать драку между

20 ними) и объект 816 "жилище" (HouseObject) (жилище для виртуального существа) предпочтительно производят обмен информацией с объектом 810 "доброе животное" (GoodAnimalObject). Объект 810 "доброе животное" (GoodAnimalObject) содержит такие функциональные возможности, что он способен рисовать объекты на экране и т.д.,

25 поэтому другие классы и объекты предпочтительно производят обмен информацией с объектом 810 "доброе животное" (GoodAnimalObject). Конечно же, в этой системе может существовать множество других классов и объектов, так как, например, для виртуального существа может быть, но не обязательно, предусмотрено наличие других игрушек.

30 Предпочтительно, но не обязательно, объект 806 класса "виртуальный мир" (WorldObject) может быть также связан с состоянием интеллектуального агента, например, за счет обеспечения ранжированного ввода данных в состояние. Эти введенные данные предпочтительно являются ранжированными в том смысле, что они создают либо стимул для интеллектуального агента, либо препятствие для

35 интеллектуального агента; они также могут, но не обязательно, оказывать нейтральное воздействие. Агрегирование множества таких ранжированных введенных данных предпочтительно позволяет определять состояние интеллектуального агента. Как изложено в описании схемы последовательности операций из фиг.8В, а также схем

40 стратегии поиска на графе и стратегии выбора действия, показанных, соответственно, на чертежах фиг.9А и фиг.9В, ранжированные введенные данные предпочтительно агрегируют для того, чтобы дать максимальное поощрение, возвращаемое интеллектуальному агенту из виртуальной среды.

45 Эти ранжированные введенные данные также могут, но не обязательно, содержать введенные пользователем данные в виде одобрительной или неодобрительной ответной реакции для того, чтобы интеллектуальный агент имел, соответственно, стимул или препятствие для продолжения линии поведения, для которой была предоставлена ответная реакция. Вычисление состояния окружающего мира

50 относительно ответной реакции пользователя предпочтительно, но не обязательно, выполняют следующим образом:

$$\text{Grade} = (\text{weighting_factor} * \text{feedback_reward}) + ((1 - \text{weighting_factor}) * \text{world_reward}),$$
 то есть, Ранг = (весовой_коэффициент * поощрение_путем_ответной_реакции) + ((1-

весовой_коэффициент) * поощрение_из_виртуального_мира), при этом feedback_reward (поощрение_путем_ответной_реакции) получено в результате ответной реакции, обеспечиваемой пользователем, а world_reward (поощрение_из_виртуального_мира) представляет собой, как описано выше, агрегированное суммарное поощрение из виртуальной среды; weighting_factor (весовой_коэффициент) предпочтительно, но не обязательно, имеет значение между 0 и 1, указывающее весовой коэффициент ответной реакции пользователя в отличие от ответной реакции виртуальной среды (виртуального мира).

На фиг.8В показана пояснительная схема последовательности операций для приведенного в качестве примера набора взаимодействий между виртуальным миром и интеллектуальным агентом из настоящего изобретения. Последовательность начинается с запроса на обновление состояния интеллектуального агента, посылаемого из модуля 818 виртуального мира в класс 802 "виртуальное существо, созданное посредством искусственного интеллекта (ИИ)" (AICreature). Модуль 818 виртуального мира осуществляет контроль и административное управление всей виртуальной средой, в том числе самим интеллектуальным агентом.

Затем интеллектуальный агент обдумывает действие, которое следует выполнить, что обозначено стрелкой 1.1.1. Действие предпочтительно выбирают путем поиска (стрелка 1.1.1.1) по всем объектам виртуального мира, а затем рекурсивно по всем действиям для каждого объекта, путем взаимодействия с классом 804 "виртуальный мир" (World) и с объектом 806 класса "виртуальный мир" (WorldObject). Производят оценку потенциально возможного поощрения для каждого действия (стрелка 1.1.1.1.1) и его ранжирование (стрелка 1.1.1.1.1.2). Выбирают действие с наиболее высоким поощрением. Затем определяют общий ранг для интеллектуального агента, и класс 802 "виртуальное существо, созданное посредством искусственного интеллекта (ИИ)" (AICreature) выполняет выбранное действие.

Затем модуль 818 виртуального мира (Virtual_world) обновляет местоположение и состояние всех объектов в виртуальном мире, осуществляя обмен информацией с классом 804 "виртуальный мир" (World) и объектом 806 класса "виртуальный мир" (WorldObject).

Поиск по различным потенциально возможным действиям может быть произведен согласно одному или большему количеству нескольких различных способов по выбору. На фиг.9А и фиг.9В показаны два способа выбора действия согласно настоящему изобретению, которые приведены в качестве примеров.

На фиг.9А показан пример способа выбора действия, именуемого здесь "стратегией выбора действия на основе правил". На этапе 1 состояние виртуальной среды определяется состоянием виртуального мира (World state). Происходит событие в виртуальном мире, после которого на этапе 2 вызывают обработчик состояния, соответствующий этому событию. На этапе 3 обработчик состояния предпочтительно выдает запрос в базу знаний. База знаний может быть, но не обязательно, разделена на отдельные разделы и/или на отдельные базы знаний в соответствии с вызванным обработчиком состояния. На этапе 4 в обработчик состояния возвращают ответ.

На этапе 5 выполняют проверку по базе правил, при которой ответ (и, следовательно, предложенное действие, которое, в свою очередь, приводит интеллектуального агента в конкретное состояние) сравнивают с правилами. Если действие является недопустимым, то в последовательности операций возвращаются к этапу 1. Если действие является допустимым, то на этапе 6 осуществляют генерацию действия. Приоритет для действия (более подробное описание которого приведено

ниже при пояснении фиг.9В) предпочтительно определяют затем на этапе 7, причем в более предпочтительном варианте приоритет определяют согласно множеству введенных данных, которыми являются, в том числе, вероятность действия, полезность действия и предпочтение пользователя, но эти примеры не являются

ограничивающим признаком. На этапе 8 действие помещают в очередь для администратора действий. На этапе 9 администратор действий извлекает действие с самым высоким приоритетом, которое затем на этапе 10 выполняется интеллектуальным агентом.

На фиг.9В показан приведенный в качестве примера способ выбора действия согласно стратегии поиска на графе. И вновь, на этапе 1 способ начинают с определения состояния виртуального мира (виртуальной среды), в том числе состояния интеллектуального агента и объектов в виртуальном мире. На этапе 2 посылают запрос интеллектуальному агенту. На этапе 3 интеллектуальный агент получает набор правомочных (разрешенных или возможных) действий для каждого объекта виртуального мира, предпочтительно для каждого объекта виртуального мира посылают запрос, что показано на чертеже.

Теперь способ разветвляется на две части. Первую часть, показанную справа, выполняют для каждого пути к действию. На этапе 4 производят моделирование выполняемого действия. На этапе 5 определяют влияние результата моделирования на виртуальный мир, и предпочтительно это влияние определяют для каждого объекта виртуального мира на этапе 6. На этапе 7 определяют ранг влияния каждого действия.

На этапе 8 определяют состояние объектов и, следовательно, виртуального мира, а также суммарное накопленное поощрение за действие. На этапе 9 производят моделирование влияния действия на интеллектуального агента, предпочтительно на этапе 10 также учитывают взаимное влияние между интеллектуальным агентом и каждым объектом виртуального мира.

Теперь, переходя к рассмотрению левой ветви способа, на этапе 11 всю эту информацию предпочтительно используют для определения пути к действию с наиболее высоким поощрением. На этапе 12 осуществляют генерацию действия. На этапе 13 устанавливают приоритет действия предпочтительно в соответствии с рангом действия или с поощрением. На этапе 14 действие помещают в очередь в администраторе действий, как и на фиг.9А. На этапе 15 администратор действий рассматривает действие согласно приоритету; производят выбор действия с наиболее высоким приоритетом, и на этапе 16 предпочтительно выполняют это действие.

Ниже приведено описание примера способа и структуры выполнения действия. На фиг.10 показана схема последовательности операций способа выполнения действия согласно настоящему изобретению, который приведен в качестве примера. Обработчик 1000 передает целевую установку для действия в модуль 1002 действия, что обозначено стрелкой 1, который предпочтительно содержит интерфейс для базовых действий. Интерфейс для базовых действий дает возможность модулю 1002 действия производить обмен информацией с обработчиком 1000, а также с другими объектами в системе, которые способны осуществлять генерацию действий и откладывать их выполнение для того, чтобы интеллектуальный агент, показанный здесь как приложение 1006 "плавающий агент" (FloatingAgentApp), выполнил их впоследствии. Управление этими действиями осуществляет администратор 1004 действий.

Администратор 1004 действий имеет две очереди, содержащие объекты "действие". Одна очередь представляет собой очередь объектов "действие", готовых к

выполнению, в то время как другая очередь представляет собой очередь объектов "действие", находящихся в процессе рассмотрения для очереди на выполнение.

Последняя очередь может быть использована, например, в том случае, если была осуществлена генерация действия, но внутреннее состояние действия находится в процессе рассмотрения, поэтому действие не готово к выполнению. Когда состояние действия созревает до того, что оно готово к выполнению, действие предпочтительно перемещают в очередь объектов "действие", готовых к выполнению.

Администратор 1008 приложений предпочтительно взаимодействует с приложением 1006 "плавающий агент" (FloatingAgentApp) для выполнения действия, что обозначено стрелкой 2. Затем приложение 1006 "плавающий агент" (FloatingAgentApp) предпочтительно запрашивает следующее действие от администратора 1004 действий (стрелка 2.1), само действие предпочтительно обеспечивает модуль 1002 действия (стрелка 2.2.1). Действия предпочтительно помещают в очередь от обработчика 1000 к администратору 1004 действий (стрелка 3). Целевые установки (и, следовательно, по меньшей мере, часть приоритета) предпочтительно устанавливаются для каждого действия путем обмена информацией между обработчиком 1000 и модулем 1002 действия (стрелка 4). Стрелками 5 и 6 показан способ "харакири()", более подробное описание которого приведено ниже.

Как описано выше, действия предпочтительно поставлены в очередь в порядке приоритетов. Приоритет предпочтительно определяют посредством выдачи запроса в интерфейс модуля 1002 действия администратором 1004 действий. Приоритет действия предпочтительно определяют согласно вычислению, содержащему множество параметров. Например, параметры предпочтительно содержат приоритет, полученный путем вычислений или логических выводов объектом, осуществляющим генерацию, в более предпочтительном варианте полученный на основании прогнозируемой вероятности успеха действия, постоянный приоритет для действия этого типа, который предпочтительно определяют согласно прошлому опыту для действия этого типа (например, согласно приемлемости для пользователя и успеху действия), и приоритет целевой установки, который предпочтительно определяют согласно предпочтениям пользователя.

Одним из необязательных вариантов вычислений для управления вышеуказанными параметрами является следующий:

$$P(\text{общий}) = P(\text{вероятность действия}) * ((P(\text{постоянный приоритет}) + P(\text{целевая установка для действия})/10)/2).$$

В дополнение к выполнению действия на основании приоритета, каждое действие предпочтительно имеет срок "времени жизни", ВЖ (ttl), это значение ВЖ (ttl) означает величину промежутка времени выполнения, прошедшего между моментом, когда отложенное действие было помещено в очередь действий, готовых к выполнению, и временем истечения срока этого действия. Если действие готово, но не получает приоритет для его выполнения до того момента, когда истекло его ВЖ (ttl), администратор 1004 действий предпочтительно вызывает метод "харакири()", который уведомляет действие о том, что оно не будет выполняться. Каждый такой вызов "харакири()" предпочтительно снижает приоритет действия до тех пор, пока не будет достигнуто пороговое значение. После того, как было достигнуто это пороговое значение, постоянный приоритет предпочтительно начинает возрастать. Эта модель функционирует таким образом, что производит обработку действий, которые были предложены или были выполнены, но неудачно, поскольку действие было прервано пользователем. Постоянный приоритет уменьшается за счет того, что в вычисление

приоритета действия вводят прошлый опыт.

Этот способ демонстрирует то, каким образом происходит адаптация предложенных или выполнявшихся действий к неявным предпочтениям конкретного пользователя во время выполнения программы.

Эта модель не является полной без механизма "харакири()", поскольку, если снижается постоянный приоритет действия, то действие не выполняют, поэтому должна быть предоставлена возможность либо аннулировать это действие, либо, возможно, выполнить его снова, например, если изменяются предпочтения пользователя. После нескольких выполнений "харакири()" действие может восстановить приоритет для его выполнения.

В предыдущих разделах предложена инфраструктура, которая дает возможность выполнять различные действия и реализовать различные алгоритмы посредством адаптивной системы из настоящего изобретения. Более подробное описание этих действий и алгоритмов приведено ниже.

Раздел 3: Система формирования эмоций

В этом разделе приведено описание предпочтительного варианта осуществления системы формирования эмоций согласно настоящему изобретению, в том числе описание конкретных эмоций и их силы, которые предпочтительно объединяют для формирования общего настроения. Система формирования эмоций также предпочтительно содержит механизм, позволяющий изменять настроение, а также обеспечивающий возможность по выбору управлять одним или большим количеством аспектов такого изменения, например, скоростью изменения.

На фиг.11А-фиг.11С представлены схемы, описывающие вариант реализации системы формирования эмоций согласно настоящему изобретению, который приведен в качестве иллюстративного примера. На фиг.11А показан пример схемы классов для системы формирования эмоций, а на фиг.11В и фиг.11С показаны примеры схем последовательностей операций для обеспечения функционирования системы формирования эмоций согласно настоящему изобретению.

Как показано на чертеже, в системе 1100 формирования эмоций согласно настоящему изобретению класс целевых установок (целевая установка 1102) отображает абстрактную целевую установку интеллектуального агента. Целевая установка представляет собой то, чего пытается достичь интеллектуальный агент, выполняя действие. Целевая установка 1102 является ответственной за создание эмоций на основании конкретных событий, которые связаны с состоянием целевой установки и с возможностью ее успешного выполнения.

Целевая установка 1102 взаимодействует с классом 802 "виртуальное существо, созданное посредством искусственного интеллекта (ИИ)" (AICreature) (описание которого также было приведено выше при рассмотрении фиг.8). Более подробное описание этих взаимодействий приведено ниже. В кратком изложении, интеллектуальный агент пытается выполнить целевые установки, поэтому для определения того, были ли выполнены целевые установки, требуются взаимодействия между классом 802 "виртуальное существо, созданное посредством искусственного интеллекта (ИИ)" (AICreature), которые, в свою очередь, оказывают воздействие на эмоциональное состояние интеллектуального агента.

Обработку самого эмоционального состояния осуществляет класс 1104 "эмоциональное состояние" (EmotionalState), который, в свою очередь, соединен с классом 1106 "эмоция" (Emotion). Сам класс 1106 "эмоция" (Emotion) предпочтительно соединен с классами конкретных эмоций, например, с классом 1108 "эмоция гнева"

(AngerEmotion) и с классом 1110 "эмоция радости" (JoyEmotion). Класс 1104 "эмоциональное состояние" (EmotionalState) также предпочтительно соединен с классом 1112, определяющим стереотип поведения, который именуют "средством отображения стереотипа поведения" (BehavioralPatternMapper).

Создание эмоции предпочтительно осуществляют посредством системы формирования эмоций тогда, когда увеличивается или уменьшается вероятность успеха, ВУ (LOS), и когда увеличивается или уменьшается вероятность неудачи, ВН (LOF). Когда ВУ (LOS) увеличивается, то предпочтительно осуществляют генерацию эмоции надежды. Когда ВУ (LOS) уменьшается, то предпочтительно осуществляют генерацию эмоции отчаяния. Когда ВН (LOF) увеличивается, то предпочтительно осуществляют генерацию эмоции страха, а когда ВН (LOF) уменьшается, то предпочтительно осуществляют генерацию эмоции радости.

Успех или неудача достижения целевой установки оказывают существенное влияние на состояние целевой установки и на создаваемые эмоции. При неудаче в достижении целевой установки предпочтительно осуществляют генерацию эмоции отчаяния, а если вероятность успеха была высока, то предпочтительно также осуществляют генерацию эмоции разочарования (поскольку надежды на успех были высокими).

При успешном достижении целевой установки предпочтительно осуществляют генерацию эмоции радости, а если надежды на успех и накопленный успех были высоки, то предпочтительно осуществляют генерацию эмоции гордости.

Класс 1106 "эмоция" (Emotion) представляет собой структуру, которая имеет два свойства, которые по своему типу являются главным и второстепенным. Главный тип описывает группу высокого уровня, к которой принадлежит второстепенная эмоция и которая предпочтительно содержит

ПОЛОЖИТЕЛЬНУЮ_ЭМОЦИЮ (POSITIVE_EMOTION) и ОТРИЦАТЕЛЬНУЮ_ЭМОЦИЮ (NEGATIVE_EMOTION). Второстепенные типы предпочтительно содержат следующие эмоции: РАДОСТЬ, НАДЕЖДА, ЗЛОРАДСТВО, ГОРДОСТЬ, СИМПАТИЯ, ГНЕВ, НЕНАВИСТЬ, СТРАХ, РАЗОЧАРОВАНИЕ, ГОРЕ, НЕДОВОЛЬСТВО. Другими свойствами эмоций являются сила эмоции, задаваемая при ее генерации, и политика спада эмоции (то есть скорость изменения эмоции).

Следующий этап после генерации эмоции выполняет класс 1104 "эмоциональное состояние" (EmotionalState), который накапливает эмоции, генерация которых была осуществлена интеллектуальным агентом в течение некоторого времени. Этот класс представляет собой совокупность экземпляров эмоций, которая определяет текущее эмоциональное состояние интеллектуального агента. Текущее эмоциональное состояние предпочтительно определяют путем поддержки иерархии типов эмоций, которые затем обобщают путем агрегирования и корреляции. Например, второстепенные эмоции предпочтительно сводят в количественный показатель для ПОЛОЖИТЕЛЬНОЙ_ЭМОЦИИ (POSITIVE_EMOTION) и количественный показатель для ОТРИЦАТЕЛЬНОЙ_ЭМОЦИИ (NEGATIVE_EMOTION), затем эти две категории предпочтительно соотносят с ХОРОШИМ/ПЛОХИМ НАСТРОЕНИЕМ, которое описывает общее настроение интеллектуального агента.

"Плавающее" приложение интеллектуальный агент опрашивает класс 1104 "эмоциональное состояние" (EmotionalState) всякий раз, когда изменяется доминирующий стереотип поведения (в вышеописанной модели - вследствие генерации, угасания и обобщения эмоций), интеллектуальный агент предпочтительно выражает свое эмоциональное состояние, и его поведение соответствует этому

стереотипу поведения. Интеллектуальный агент предпочтительно, но не обязательно, выражает свое эмоциональное состояние с использованием одного или большего количества средств передачи текстовой информации (описано более подробно ниже), трехмерной анимации, выражения лица, двумерных анимационных эффектов и звуков.

На фиг.11В показан пример схемы последовательности операций для генерации эмоции системой формирования эмоций согласно настоящему изобретению. Как показано на чертеже, администратор 502 приложений (более подробное описание которого приведено при пояснении фиг.5) передает в приложение 1006 "плавающий агент" (FloatingAgentApp) (более подробное описание которого приведено при пояснении фиг.10) команду "выполнить шаг программы" (step), что обозначено стрелкой 1. Затем приложение 1006 "плавающий агент" (FloatingAgentApp) определяет ВН (LOF) (вероятность неудачного исхода) путем опроса класса 1102 целевых установок, что обозначено стрелкой 1.1. Затем целевая установка 1102 определяет ВН (LOF), если новая ВН (LOF) является более высокой, чем предварительно определенная ВН (LOF), то предпочтительно осуществляют генерацию эмоции страха путем выдачи запроса в класс 1106 "эмоция" (Emotion), что обозначено стрелкой 1.1.1.1. Эмоцию страха также добавляют к эмоциональному состоянию путем обмена информацией с классом 1104 "эмоциональное состояние" (EmotionalState), что обозначено стрелкой 1.1.1.2.

Затем администратор 502 приложений передает команду "выполнить другой шаг программы" (стрелка 2) в приложение 1006 "плавающий агент" (FloatingAgentApp), которое определяет ВУ (LOS) (вероятность успеха) опять-таки путем опроса класса 1102 целевых установок, что обозначено стрелкой 2.1. Затем целевая установка 1102 определяет вероятность успеха, ВУ (LOS), если новая ВУ (LOS) является более высокой, чем предварительно определенная ВУ (LOS), то предпочтительно осуществляют генерацию эмоции надежды путем выдачи запроса в класс 1106 "эмоция" (Emotion), что обозначено стрелкой 2.1.1.1. Эмоцию надежды также добавляют к эмоциональному состоянию путем обмена информацией с классом 1104 "эмоциональное состояние" (EmotionalState), что обозначено стрелкой 2.1.1.2.

Стрелкой 3 показана передача администратором 502 приложений команды "выполнить еще один шаг программы" в приложение 1006 "плавающий агент" (FloatingAgentApp), которое выдает запрос на определение эмоции согласно фактическому результату действия. Если действие было неудачным, а последняя ВУ (LOS) была более высокой, чем некоторый коэффициент, равный, например, 0,5, который указывал, что ожидался успешный результат, то приложение 1006 "плавающий агент" (FloatingAgentApp) обуславливает то, что целевая установка 1102 обеспечивает генерацию эмоции отчаяния классом 1106 "эмоция" (Emotion), что обозначено стрелкой 3.1.1.1. Эмоцию отчаяния также добавляют к эмоциональному состоянию путем обмена информацией с классом 1104 "эмоциональное состояние" (EmotionalState), что обозначено стрелкой 3.1.1.2. К тому же, если действие является неудачным (предпочтительно вне зависимости от ожидания успешного результата), класс 1106 "эмоция" (Emotion) предпочтительно осуществляет генерацию эмоции горя, что обозначено стрелкой 3.1.2. Эмоцию горя также добавляют к эмоциональному состоянию путем обмена информацией с классом 1104 "эмоциональное состояние" (EmotionalState), что обозначено стрелкой 3.1.3.

Затем администратор 502 приложений передает команду "выполнить еще один шаг программы" (стрелка 4) в приложение 1006 "плавающий агент" (FloatingAgentApp),

которое обновляет эмоции на основании фактического успешного результата путем передачи сообщения в целевую установку 1102, что обозначено стрелкой 4.1. Затем целевая установка 1102 предпочтительно вызывает генерацию предпочтительно эмоции радости путем выдачи запроса в класс 1106 "эмоция" (Emotion), что обозначено стрелкой 4.1.1. Эмоцию радости также добавляют к эмоциональному состоянию путем обмена информацией с классом 1104 "эмоциональное состояние" (EmotionalState), что обозначено стрелкой 4.1.2.

Если фактический успех является большим, чем спрогнозированный, то целевая установка 1102 предпочтительно вызывает генерацию эмоции гордости путем выдачи запроса в класс 1106 "эмоция" (Emotion), что обозначено стрелкой 4.1.3.1. Эмоцию гордости также добавляют к эмоциональному состоянию путем обмена информацией с классом 1104 "эмоциональное состояние" (EmotionalState), что обозначено стрелкой 4.1.3.2.

На фиг.11С показан пример схемы последовательности операций для выражения эмоции системой формирования эмоций согласно настоящему изобретению. Такое выражение эмоций предпочтительно обусловлено предпочтениями пользователя. Администратор 502 приложений инициирует выражение эмоции путем передачи команды "выполнить шаг программы" (стрелка 1) в приложение 1006 "плавающий агент" (FloatingAgentApp), которое выдает в модуль 1108 отображения стереотипа поведения (bp_mapper) запрос о стереотипе поведения интеллектуального агента, что обозначено стрелкой 1.1. Если доминирующее поведение изменилось, то приложение 1006 "плавающий агент" (FloatingAgentApp) передает в модуль 1110 воспроизведения стереотипа поведения (bp_display) запрос для задания стереотипа поведения (стрелка 1.2.1). Модуль 1110 воспроизведения стереотипа поведения (bp_display) управляет фактическим отображением на экране эмоции. Затем приложение 1006 "плавающий агент" (FloatingAgentApp) выдает запрос на помещение действия в очередь в сообщении, передаваемом в администратор 1004 действий (стрелка 1.2.2).

Администратор 502 приложений передает команду "выполнить другой шаг программы" (стрелка 2) в приложение 1006 "плавающий агент" (FloatingAgentApp), которое выдает в администратор 1004 действий запрос на удаление действия из очереди (стрелка 2.1) и на выполнение действия модулем 1110 воспроизведения стереотипа поведения (bp_display).

Раздел 4: Обмен информацией с пользователем

В этом разделе описан предпочтительный вариант осуществления системы обмена информацией для обмена информацией с пользователем согласно настоящему изобретению, в том числе для обмена текстовой информацией, звуковой информацией и графической информацией, но эти примеры не являются ограничивающим признаком. В качестве примера этих типов обмена информацией приведено описание обмена текстовой информацией, при этом подразумевают, что этот пример приведен только лишь в описательных целях и никоим образом не является ограничивающим признаком.

На фиг.12 показан пример схемы последовательности операций для обмена текстовой информацией согласно настоящему изобретению. Средство 1200 генерации текста является ответственным за генерацию текста, соответствующего некоторому событию, и интеллектуальный агент может осуществлять обмен информацией с ним. Средство 1200 генерации текста предпочтительно содержит средство генерации предложений или коротких фраз на естественном языке согласно шаблонам, которые

являются заранее заданными и содержат зарезервированные места для символов-заполнителей. Комбинирование шаблонов и символов-заполнителей друг с другом дает возможность средству 1200 генерации текста осуществлять генерацию большого количества фраз, соответствующих событию, к которому относится шаблон.

Эта структура может быть, но не обязательно, расширяемой для множества новых и/или изменяющихся событий или тем, поскольку также могут быть добавлены дополнительные шаблоны, а также дополнительные символы-заполнители.

Как показано на фиг.12, приложение 1006 "плавающий агент" (FloatingAgentApp) осуществляет обмен информацией со средством 1200 генерации текста следующим образом: сначала передают запрос на генерацию текста предпочтительно для конкретного события (стрелка 1). Средство 1200 генерации текста предпочтительно выбирает шаблон предпочтительно из множества шаблонов, которые являются соответствующими этому событию (стрелка 1.1). Средство 1200 генерации текста также предпочтительно выбирает символ-заполнитель для шаблона предпочтительно из множества символов-заполнителей, которые являются соответствующими этому событию (стрелка 1.2.1). Заполненный шаблон затем возвращают в приложение 1006 "плавающий агент" (FloatingAgentApp).

Ниже приведено описание примера генерации текста для события изменения настроения, которое является таким, что интеллектуальный агент в текущий момент находится в радостном состоянии, при помощи некоторых шаблонов и символов-заполнителей, которые приведены в качестве примера, не являющегося ограничивающим признаком. Шаблонами могут, но не обязательно являться следующие:

Первый шаблон радости: "%noun1 is %happy_adj2" ("%существительное1 %прилагательное2_радостный")

Второй шаблон радости: "%self_f_pronoun %happy_adj 1" ("%местоимение %прилагательное1_радостный")

Символами-заполнителями могут, но не обязательно являться следующие:

%noun1 = {"the world", "everything", "life", "this day", "the spirit"} ({"окружающий мир", "всё", "жизнь", "этот день", "душевное состояние"})

%happy_adj1 = {"happy", "joyful", "glad", "pleased", "cheerful", "in high spirits", "blissful", "exultant", "delighted", "cheery", "jovial", "on cloud nine"} ({"счастливый", "радостный", "веселый", "довольный", "жизнерадостный", "находящийся в хорошем настроении", "испытывающий блаженство", "ликующий", "получающий наслаждение", "в радостном расположении духа", "в веселом расположении духа", "находящийся на седьмом небе"})

%happy_adj2 = {"nice", "beautiful", "great", "happy", "joyful", "good", "fun"} ({"приятный", "красивый", "прекрасный", "счастливый", "радостный", "хороший", "забавный"})

%self_f_pronoun = {"I am", "I'm", "your intelligent agent", "your agent friend"} ({"Я", "Я", "ваш интеллектуальный агент", "друг вашего агента"})

Примерами некоторых фраз для обмена текстовой информацией, полученных в результате комбинаций шаблонов и символов-заполнителей, являются следующие:

I'm cheerful (я радостен)

the spirit is joyful (нахожусь в радостном расположении духа)

I am exultant (я ликую)

life is beautiful (жизнь прекрасна)

life is good (жизнь хороша)

I'm pleased (я доволен)

I'm jovial (я нахожусь в веселом расположении духа)

I am joyful (я весел)

the world is joyful (окружающий мир приносит радость)

I'm glad (я доволен)

the spirit is joyful (нахожусь в веселом расположении духа)

the spirit is happy (нахожусь в радостном расположении духа)

the world is nice (окружающий мир приятен)

I am happy (я счастлив)

В качестве другого примера, не являющегося ограничивающим, приведен шаблон пропущенного телефонного вызова, который может быть, но не обязательно, создан следующим образом:

%user missed a call from %missed %reaction

(%абонент пропустил телефонный вызов от %пропущен %реакция)

В этом примере в качестве поля %user (%абонент) используется имя абонента; в поле %missed (%пропущен) вводится имя или иной идентификатор (например, телефонный номер); поле %reaction (%реакция) является необязательным, и его используют для реакции интеллектуального агента, например, для выражения разочарования (например, "I'm sad" ("я опечален")).

Как показано на этих примерах, средство 1200 генерации текста может осуществлять генерацию соответствующих предложений для многих событий, от событий, связанных с пропущенным телефонным вызовом, до событий, связанных с разрядкой аккумулятора, что обеспечивает более широкие возможности взаимодействия пользователя с мобильным информационным устройством и делает его более понятным.

Раздел 5: Адаптивная система для телефонных вызовов и SMS-сообщений

В этом разделе приведено описание предпочтительного варианта осуществления адаптивной системы для адаптивной обработки телефонных вызовов и SMS-сообщений согласно настоящему изобретению. Это описание начинается с общего описания некоторых предпочтительных алгоритмов для функционирования системы согласно настоящему изобретению, а затем в нем описан класс обработки телефонных вызовов и обработки SMS-сообщений и схемы последовательностей операций.

Интеллектуальный набор альтернативного номера (SAN)

Алгоритм интеллектуального набора альтернативного номера (SAN) предназначен для обучения набору альтернативного номера, который наиболее вероятно следует набрать после неудачной попытки телефонного вызова. Алгоритм обучает создавать эти пары, и способен затем динамически адаптироваться к новому поведению пользователя. Эти пары из поставленных в соответствие друг другу телефонных номеров используют для того, чтобы предложить телефонный номер, по которому следует позвонить после произведенной пользователем неудачной попытки телефонного вызова.

Этот алгоритм может быть, но не обязательно, реализован следующим образом: вводят наиболее часто используемые элементы данных в первый уровень (возможно, но не обязательно, операции ввода производят после того, как частота использования элемента данных является большей, чем заранее заданное пороговое значение); предлагают пару из поставленных в соответствие друг другу телефонных номеров в качестве телефонного номера, предпочтительно согласно частоте использования пары

по перечню; определяют факт успешного или неуспешного выполнения телефонного вызова; и сохраняют окно предыстории определенных пар для каждого телефонного номера таким образом, чтобы можно было, но не обязательно, удалить наиболее старую запись.

База знаний для этого алгоритма может быть, но не обязательно, отображена в виде дерева для каждого телефонного номера исходящего вызова, при этом она содержит перечень альтернативных/повторных телефонных вызовов и/или иных предпринимаемых действий. Для каждого исходящего телефонного вызова повторный телефонный вызов предпочтительно рассматривают следующим образом: был ли первый телефонный вызов неудачным, и был ли второй телефонный вызов произведен в пределах заранее заданного промежутка времени. Телефонный номер повторного телефонного вызова добавляют в перечень телефонных номеров таких повторных телефонных вызовов для первого телефонного номера. Когда перечень заполнен, то предпочтительно забывают наиболее старый телефонный номер.

Администратор для интеллектуального управления телефонной книгой (SPBM)

Система интеллектуального управления телефонной книгой (SPBM) является не ограничивающим примером системы интеллектуальной телефонной книги, которая использует статистику использования мобильного информационного устройства и статистику телефонных вызовов для обучения возможным связям между элементами списка контактов и их взаимным свойствам. Эта система обеспечивает несколько новых отличительных особенностей телефонной книги, в том числе автоматизированное создание групп контактов и автоматизированное добавление/удаление контактов, но эти примеры не являются ограничивающим признаком.

Например, автоматизированный алгоритм управления группами предпочтительно является способным автоматически группировать телефонные номера контактных лиц в соответствии с их использованием пользователем.

Автоматизированное выявление состояния (ASD)

Система автоматизированного выявления состояния (ASD) предпочтительно дает возможность мобильному информационному устройству определять текущее состояние его использования (например, "совещание", "пользователь ушел") пользователем и предлагать такие изменения для систем интерфейса пользователя (ИП), звукового сопровождения и поведения с искусственным интеллектом (ИИ), которые удовлетворяют текущему состоянию (например, активизацию тихого режима для входящих телефонных вызовов и/или автоматическую передачу ответного сообщения "нахожусь на совещании" в ответ на SMS-сообщения). Предпочтительно, но не обязательно, эта система производит обмен информацией с одним или с большим количеством биологических датчиков, которые предпочтительно, но не обязательно, могут регистрировать биологическое состояние пользователя и/или регистрировать движение пользователя и т.д. Эти дополнительные датчики предпочтительно предоставляют информацию, которая дает возможность адаптивной системе определять правильное состояние мобильного информационного устройства, не получая конкретных данных, введенных пользователем, и/или не запрашивая пользователя о текущем состоянии пользователя. Для этого также могут быть, но не обязательно, использованы изображения, зарегистрированные камерой, которой снабжено устройство.

Другой возможный, но необязательный тип датчика позволяет устройству идентифицировать конкретного пользователя, например, путем анализа отпечатка

пальца и/или биометрической информации иных типов. Такая информация также может быть, но не обязательно, использована для обеспечения безопасности.

Алгоритм выдачи советов о переходе в режим "совещание" предназначен для того, чтобы помочь пользователю управлять режимом "не беспокоить" (do-not-disturb).

Алгоритм имеет базу правил, которая указывает вероятность того, что пользователь находится в режиме совещания и не желает, чтобы его беспокоили, в отличие от вероятности того, что пользователь не изменил режим, но готов принимать телефонные вызовы. Задача алгоритма состоит в том, чтобы помочь управлению этими переходами из одного режима в другой.

Алгоритм предпочтительно функционирует посредством вышеописанных обработчиков состояния системы искусственного интеллекта (ИИ), определяя состояние виртуального мира телефона, а также определяя тот момент, когда база правил указывает, что нужно предложить режим "совещание" (например, пользователь оборвал звонок текущего телефонного вызова и не ответил на телефонный вызов и т.д.). Обработчики состояния (StateHandlers) предпочтительно также отслеживают события противоположного типа, которые могут указывать на то, что режим "совещание" следует отменить.

На фиг.13А и фиг.13В показаны, соответственно, пример схемы классов и пример схемы последовательности операций для обработки телефонного вызова согласно настоящему изобретению.

Как показано на фиг.13А, схема 1300 классов для обработки телефонного вызова содержит обработчик 1302 состояния телефонного вызова (CallStateHandler), который является ответственным за генерацию действий "предложить телефонный вызов" (SuggestCall) классом 1304 "предложить телефонный вызов" (SuggestCall).

Обработчик 1302 состояния телефонного вызова (CallStateHandler) предпочтительно представляет собой алгоритм на основе правил, который отслеживает события, связанные с телефонным вызовом, например, событие 1306 "телефонный вызов начат" (CallStartedEvent), событие 1308 "телефонный вызов завершен" (CallEndedEvent) и событие 1310 "телефонный вызов является неудачным" (CallFailedEvent), каждое из этих событий, в свою очередь, осуществляет обмен информацией с классом 1312 "событие, связанное с телефонным вызовом" (CallEvent). Обработчик 1302 состояния телефонного вызова (CallStateHandler) предпочтительно также поддерживает базу правил, которая является ответственной за две главные функции: обучение машины, которое поддерживает базу знаний о связях между телефонными вызовами; и вывод умозаключений, осуществляемый системой искусственного интеллекта (ИИ) на основании вероятности, о том, следует ли предложить пользователю телефонный номер для телефонного вызова (обработку этих предложений предпочтительно осуществляют посредством класса 1314 "предложить повторный телефонный вызов" (SuggestFollowingCall) или класса 1316 "предложить альтернативный телефонный вызов" (SuggestAlternativeCall)).

Генерацию объектов "события, связанные с телефонным вызовом" осуществляют с использованием модели события описанным выше способом, опять-таки с использованием функции обработчика прерываний в операционной системе мобильного информационного устройства. Если это возможно, то предпочтительно данные о телефонном вызове заполняют информацией о сгенерированном событии (которой является телефонный номер, имя контактного лица, время начала вызова, длительность и т.д.).

Классы "предложить телефонный вызов" (номера позиций 1304, 1314 и 1316)

реализуют интерфейс для базовых действий, описанный в адаптивной модели действия. Обязанность этих классов состоит в том, чтобы предложить пользователю телефонный номер для того, чтобы он сделал повторный телефонный вызов после завершения телефонного вызова, или для того, чтобы он сделал альтернативный телефонный вызов после того, как телефонный вызов оказался неудачным.

Обработчик 1302 состояния телефонного вызова (CallStateHandler) отслеживает события, связанные с телефонным вызовом, и классифицирует события согласно его базе правил (с использованием стратегии выбора действия на основе правил).

Иллюстративный пример возможной, но необязательной, базы правил предложений в отношении телефонного вызова имеет следующий вид:

1. Если телефонный вызов начат, и предыдущий телефонный вызов не отмечен → телефонный вызов отмечают как начатый.

2. Если телефонный вызов завершен, и телефонный вызов был отмечен как начатый → его отмечают как первый телефонный вызов.

3. Если телефонный вызов был начат, предыдущий телефонный вызов был отмечен как первый телефонный вызов, и время между телефонными вызовами является меньшим (<), чем пороговое значение для повторного телефонного вызова → его отмечают как повторный телефонный вызов.

4. Если телефонный вызов отмечен как повторный → обновляют базу знаний.

5. Если телефонный вызов отмечен как повторный → этот телефонный вызов отмечают как первый телефонный вызов (это устанавливает текущий первый телефонный вызов в исходное положение для сравнения со следующим телефонным вызовом).

6. Если телефонный вызов был неудачным, и предыдущий телефонный вызов отмечен как начатый телефонный вызов → этот телефонный вызов отмечают как первый неудачный.

7. Если телефонный вызов начат, предыдущий телефонный вызов отмечен как первый неудачный, и время между телефонными вызовами является меньшим (<), чем пороговое значение для неудачного телефонного вызова → его отмечают как альтернативный телефонный вызов.

8. Если телефонный вызов отмечен как альтернативный → обновляют базу знаний.

9. Если телефонный вызов завершен, время, прошедшее с момента завершения телефонного вызова, является меньшим (<), чем пороговое значение для предложенного повторного телефонного вызова, и сделано умозаключение о повторном телефонном вызове → осуществляют генерацию действия, связанного с предложенным повторным телефонным вызовом (например, действия, которое будет предпринято после телефонного вызова).

10. Если телефонный вызов был неудачным, время, прошедшее с момента неудачного телефонного вызова, является меньшим (<), чем пороговое значение предложенного альтернативного телефонного вызова, и сделано умозаключение о повторном телефонном вызове → осуществляют генерацию действия, связанного с предложенным альтернативным телефонным вызовом.

11. Если время, прошедшее с момента последней отметки является большим (>), чем пороговое значение → снимают отметку со всех телефонных вызовов.

База знаний для предложений в отношении телефонного вызова предпочтительно, но не обязательно, выполнена в виде окна предыстории, в котором связи добавляют в виде событий, произошедших в предыстории объекта. В этом случае соответствующим объектом для альтернативного или повторного телефонного

вызова для определенного контактного лица является само контактное лицо, и все связанные с ним телефонные вызовы предпочтительно расположены в порядке частоты возникновения этих событий в окне предыстории альтернативных телефонных вызовов или повторных телефонных вызовов.

Например, для телефонного номера контактного лица, которым является номер 054-545191, связи между телефонными вызовами в окне предыстории могут быть, но не обязательно, представлены следующим образом:

054-545191 →

052-552211	052-552212	051-546213	052-552211	051-546213	052-552211	051-897555	051-897555	052-552211	052-552211
------------	------------	------------	------------	------------	------------	------------	------------	------------	------------

Размер окна предыстории предпочтительно определен алгоритмом как количество связей (случаев возникновения события), которыми алгоритм должен управлять (или которые должен помнить). Если окно предыстории заполнено, то новое событие предпочтительно добавляют таким образом, чтобы оно было первым, а последний элемент затем удаляют так, чтобы размер окна не превысил его заданный размер. Эта база знаний способна адаптироваться к изменениям шаблонов пользователя, поскольку старые связи удаляют (забывают), заменяя их более новыми связями.

База знаний является недостаточной для того, чтобы предложить альтернативные или повторные телефонные вызовы, поскольку хорошее предложение должно быть получено из базы знаний путем логических умозаключений. Алгоритм логических умозаключений предпочтительно представляет собой простой алгоритм логических умозаключений на основе вероятности для определения наиболее вероятной цели связей в соответствии с базой знаний. С учетом следующих параметров:

C_0 - контактное лицо

$H_0(C_i)$ - количество раз, которое контактное лицо C_i встречается в окне предыстории для контактного лица C_0

H_{size} - размер окна предыстории

Способ предпочтительно предлагает контактное лицо i из условия, что:

$$P(C_i) = \text{Max}(H_0(C_i)/H_{size})$$

В приведенном выше примере для $C_0 = 054-545191$:

максимальным является $P(052-552211)$ и = 0,6.

Процесс вывода логического умозаключения считают успешным только в том случае, если в нем может быть сделано логическое умозаключение о том, что вероятность является большей, чем 50%, и предпочтительно также и в том случае, если заполнено окно предыстории.

На фиг.13В показан пример схемы последовательности операций для обработки телефонного вызова. Как показано, диспетчер 604 событий (EventDispatcher) (описанный более подробно на фиг.6) передает уведомление о событии, связанном с телефонным вызовом, в обработчик 1302 состояния телефонного вызова (CallStateHandler) (стрелка 1), который затем производит оценку базы правил (стрелка 1.1). Выдают запрос на добавление связи в окно 1318 предыстории (HistoryWindow) (стрелка 1.1.1).

Для события завершения телефонного вызова или неудачного телефонного вызова диспетчер 604 событий (EventDispatcher) передает уведомление в обработчик 1302 состояния телефонного вызова (CallStateHandler) (стрелка 2), который затем производит оценку базы правил (стрелка 2.1). Выдают запрос на добавление связи в окно 1318 предыстории (HistoryWindow) (стрелка 2.1.1). Выдают запрос на получение

вероятной связи из окна 1318 предыстории (HistoryWindow) (стрелка 2.2). Эта вероятная связь представляет собой, например, телефонный номер, по которому следует звонить, и ее передают из обработчика 1302 состояния телефонного вызова (CallStateHandler) в класс 1304 "предложить телефонный вызов" (SuggestCall) (стрелка 2.3). Администратор 1008 действий помещает действие в очередь (стрелка 2.4).

Другой необязательный, но предпочтительный алгоритм помогает пользователю управлять пропущенными телефонными вызовами и функцией отложенного звонка. Этот алгоритм нацелен на распознавание важных телефонных вызовов, которые были пропущены (возможно, во время режима отложенного звонка), и предлагают интеллектуальный обратный вызов. Этот обратный вызов предлагают только для тех телефонных номеров, которые были идентифицированы пользователем (или базой знаний) как важные.

База знаний основана на двух возможных (дополнительных) вариантах. Первый вариант является явным, в нем пользователь указывает важность телефонного вызова после того, как он был произведен, и иную информацию в поле расширенной адресной книги. Второй явный вариант определяется частотой телефонного вызова и иными параметрами.

Алгоритм может предложить обратный вызов в том случае, если телефонный номер обратного вызова является важным, и пользователем не был сделан телефонный звонок в течение некоторого промежутка времени, и/или если не был произведен входящий телефонный звонок с искомого телефонного номера.

Логические умозаключения об адресате SMS-сообщения на основании содержимого сообщения (CSAI)

Алгоритм логических умозаключений об адресате SMS-сообщения на основании содержимого сообщения (CSAI) предназначен для того, чтобы предпочтительно, но не обязательно, прогнозировать адресата сообщения по содержимому сообщения. Этот алгоритм предпочтительно обучается распознаванию определенных словарных структур в сообщении и установлению их связи с существующим контактным лицом из адресной книги. После завершения сообщения это контактное лицо предлагают в качестве адресата сообщения.

Этот алгоритм может, но не обязательно, действовать согласно одному или большему количеству правил, которые затем интерпретирует интерпретатор правил. Адаптивная система (например, посредством обучающегося модуля) предпочтительно обучается таблице, например, из 1000 слов. Каждое новое слово, появляющееся в исходящем SMS-сообщении, добавляют в перечень. Для каждого слова имеется запись для каждого контактного лица, которому посылают SMS-сообщения (то есть, для контактных лиц, которым было послано, по меньшей мере, одно SMS-сообщение). Каждая запись слово/контактное лицо содержит то количество раз, которое слово появлялось в SMS-сообщениях, посланных в адрес этого контактного лица, предпочтительно со сведениями о количестве SMS-сообщений, посланных в адрес каждого контактного лица.

Для того чтобы работал алгоритм логических умозаключений предпочтительно для каждого слова W в текущем SMS-сообщении и для каждого контактного лица C вычисляют вероятность того, что $P(C|W)$, на основании значений $P(W|C)$, заданных в таблице, и $P(C)$, также вычисляемых исходя из таблицы. Затем добавляют количество терминов, нормированное на количество слов в текущем SMS-сообщении.

К способу обработки SMS-сообщений обращаются при анализе содержимого SMS-сообщения и логических умозаключениях об адресе "кому послать".

Предпочтительно, но не обязательно, в алгоритме используют изложенную ниже эвристику с конкретными словами-индикаторами, которые неоднократно появляются при передаче сообщения конкретному адресату.

5 Каждое новое слово, появляющееся в исходящем SMS-сообщении, добавляют в перечень. Для каждого слова имеется запись для каждого контактного лица, которому посылают SMS-сообщения (то есть для контактных лиц, которым было послано, по меньшей мере, одно SMS-сообщение). Каждая запись слово/контактное лицо содержит то количество раз, которое слово появлялось в SMS-сообщениях, посланных в адрес
10 этого контактного лица. К тому же, предпочтительно запоминают количество SMS-сообщений, посланных в адрес каждого контактного лица. Обучение предпочтительно происходит путем обновления таблицы слов после синтаксического анализа вновь переданных SMS-сообщений.

15 Способ логических умозаключений искусственного интеллекта (ИИ) предпочтительно функционирует с простой вероятностью, как изложено в приведенном выше описании.

На фиг.14А и фиг.14В изображены, соответственно, иллюстративные, не являющиеся ограничивающими примеры класса обработки SMS-сообщений и схемы
20 последовательности операций обработки SMS-сообщений согласно настоящему изобретению.

На фиг.14А показан пример схемы 1400 классов обработки SMS-сообщений согласно настоящему изобретению. Показаны следующие классы: класс 1402 "обработчик состояния SMS-сообщения" (SMSstateHandler) и класс 1404 "предложить
25 адресата для передачи SMS-сообщения" (SuggestSMStoSend). Обработчик 1402 состояния SMS-сообщения (SMSstateHandler) является ответственным за получение информации о состоянии передачи SMS-сообщения, затем взаимодействует с классом 1404 "предложить адресата для передачи SMS-сообщения" (SuggestSMStoSend)
30 для того, чтобы предложить адрес (телефонный номер), по которому должно быть передано SMS-сообщение.

На фиг.14В показан пример схемы последовательности операций для выдачи такого предложения. Диспетчер 604 событий (EventDispatcher) (см. фиг.6 для более
35 подробного объяснения) передает в обработчик 1402 состояния SMS-сообщения (SMSstateHandler) уведомление о событии, связанном с SMS-сообщением (стрелка 1). Обработчик 1402 состояния SMS-сообщения (SMSstateHandler) начинает действовать, анализируя базу знаний (стрелка 1.1.1), затем в класс 1406 "данные об SMS-сообщениях" (SMSdata) передают
40 запрос на получение информации о контактных лицах (стрелка 1.1.1.1). SMS-сообщение предпочтительно снабжают метками (например, подвергают синтаксическому анализу), что обозначено стрелкой 1.1.1.2, и выдают запрос на получение предложенного адреса контактного лица из класса 1406 "данные об SMS-сообщениях" (SMSdata) (стрелка 1.1.1.3).

45 Если предложенный адрес контактного лица получен, то предпочтительно обработчик 1402 состояния SMS-сообщения (SMSstateHandler) осуществляет генерацию действия, которое передают в класс 1404 "предложить адресата для передачи SMS-сообщения" (SuggestSMStoSend) (стрелка 1.1.2.1.1), после чего задают целевую
50 установку для этого действия (стрелка 1.1.2.1.2) и помещают действие в очередь (стрелка 1.1.2.1.3) путем передачи в администратор 1008 действий (см. фиг.10 для более подробного объяснения).

После того, как SMS-сообщение послано, передают уведомление из диспетчера 604

событий (EventDispatcher) в обработчик 1402 состояния SMS-сообщения (SMSstateHandler) (стрелка 2), который обрабатывает это состояние (стрелка 2.1), при этом обработка предпочтительно содержит операцию обновления базы знаний (стрелка 2.1.1) и введения в нее данных о новом SMS-сообщении (стрелка 2.1.1.1), при которой производят обмен информацией с классом 1406 "данные об SMS-сообщениях" (SMSdata).

Раздел 6: Адаптивная система для меню

В этом разделе описан предпочтительный вариант осуществления адаптивной системы для адаптивной обработки меню согласно настоящему изобретению. Сначала приведено общее описание алгоритма для построения, компоновки и перекомпоновки меню, после которого приведено описание примера схемы классов обработки меню (фиг.15).

Адаптивная система меню основана на способности производить индивидуальную настройку системы меню или интерфейса пользователя, которым снабжена операционная система мобильного информационного устройства, для человека с использованием автоматических логических умозаключений. Все операционные системы с графическим интерфейсом пользователя имеют систему меню, окон или эквивалентную систему интерфейса пользователя. Многие операционные системы имеют возможность индивидуальной настройки системы меню или системы окон для конкретного пользователя, осуществляемой вручную или путем административного управления. Описанная система обеспечивает возможность автоматизированной индивидуальной настройки интерфейса пользователя. Описанная система осуществляет генерацию автоматических действий (возможно с одобрения пользователя или автоматически). Система использует структуру системы меню и создает необходимые абстракции и необходимую базу знаний для вывода умозаключений о правильном действии по индивидуальной настройке и для обеспечения способности автоматического использования опций индивидуальной настройки, предусмотренных в операционной системе.

Интеллектуальный компоновщик меню (ИМА)

Алгоритм интеллектуального компоновщика меню (ИМА) предназначен для создания меню интерфейса пользователя (ИП) в динамическом режиме на основании конкретных предпочтений пользователя и использования мобильного информационного устройства. Алгоритм предпочтительно идентифицирует характеристики использования телефонного аппарата и на основании этих характеристик создает особое личное меню.

Этот алгоритм, в свою очередь, может, но не обязательно, содержать два других алгоритма построения меню. К алгоритму автоматического создания меню быстрого вызова обращаются при генерации автоматических ярлыков для излюбленных и наиболее часто используемых приложений и субприложений. Главной задачей этого алгоритма является усовершенствование не используемого ранее большинством пользователей способа создания вручную их личного контекстного меню. Средство отображения виртуального мира телефонного аппарата (PhoneWorldMapper) накапливает сведения о выполнении приложений и субприложений для базы знаний и обучения и использует эти знания для вывода умозаключений о том, какому приложению/субприложению следует предоставить ярлык меню в пункте личного меню и создать его для пользователя.

Логические рассуждения относительно ярлыков основаны на следующей функции полезности: частота использования приложения, умноженная на весовой

коэффициент, равный количеству необходимых щелчков, сэкономленному за счет ярлыка (щелчки в обычном меню минус щелчки в ярлыке). Предложенные ярлыки и их состав обеспечивают для пользователя наиболее полезное приложение/субприложение/изображение на экране.

Другой такой алгоритм создания меню может, но не обязательно, содержать алгоритм автоматического переупорядочивания меню. К этому алгоритму обращаются при недостаточной индивидуализации в системе меню. Множество пользователей используют интерфейс пользователя телефонного аппарата различными способами, но они имеют одинаковую систему меню и интерфейс. Этот алгоритм обучает использованию, ориентированному на конкретного пользователя, и соответственным образом переупорядочивает систему меню, обеспечивая в более предпочтительном варианте полностью адаптивную систему меню.

Средство отображения виртуального мира телефонного аппарата (PhoneWorldMapper) накапливает сведения о выполнении приложений и субприложений для базы знаний, а также запоминает количество щелчков, необходимое для достижения конкретного целевого объекта. Средство отображения виртуального мира телефонного аппарата (PhoneWorldMapper) дает иерархическое представление, используемое при добавлении элементов в то же самое меню. Внутри меню элементы упорядочены по их полезности.

Предпочтительно, производят периодическую оценку того, является ли система меню оптимальной для пользователя (по описанным выше параметрам), после чего возможно, но не обязательно, производят переупорядочивание согласно наилучшему варианту, полученному путем логических умозаключений.

На фиг.15 показана схема 1500 классов адаптивной системы меню. Эта схема классов обеспечивает необходимую абстракцию посредством класса 1502 "средство отображения виртуального мира телефонного аппарата" (PhoneWorldMapper) и класса 1504 "узел виртуального мира телефонного аппарата" (PhoneWorldNode). Класс 1502 "средство отображения виртуального мира телефонного аппарата" (PhoneWorldMapper) является ответственным за отображение системы меню и системы интерфейса пользователя. Это отображение выполнено посредством класса 1504 "узел виртуального мира телефонного аппарата" (PhoneWorldNode). Класс 1504 "узел виртуального мира телефонного аппарата" (PhoneWorldNode) представляет меню, подменю или пункт меню в структуре графа.

Средство 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper) предпочтительно содержит граф объектов класса 1504 "узел виртуального мира телефонного аппарата" (PhoneWorldNode), ребра представляют собой переходы меню между узлами, а вершины представляют собой отображенные меню и пункты меню. Всякий раз, когда пользователь перемещается в системе меню, средство 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper) отслеживает граф объектов из объектов класса 1504 "узел виртуального мира телефонного аппарата" (PhoneWorldNode) и наводит его на правильное местоположение пользователя. Всякий раз, когда пользователь активизирует некоторый пункт меню, текущий узел предпочтительно записывает это действие и подсчитывает количество случаев активизации. Средство 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper) обеспечивает для настоящего изобретения способность вычислять расстояние (в щелчках) между каждым пунктом меню, а также расстояние от корневого каталога до этого пункта меню, что является возможным вследствие представления системы меню в виде графа.

Таким образом, средство 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper) обеспечивает абстракцию для структуры меню, навигации по меню, активизации меню и расстояния между пунктами меню.

Диаграмма 1500 классов предпочтительно также содержит класс 1506 "событие, связанное с меню" (MenuEvent), предназначенный для обработки событий, связанных с меню, и класс 1508 "предложить ярлык" (SuggestShortcut), предназначенный для выдачи предложений относительно создания ярлыков посредством меню. Класс 1502 "средство отображения виртуального мира телефонного аппарата" (PhoneWorldMapper)

предпочтительно осуществляет обмен информацией с классом 1510 "данные моего меню" (MyMenuData), предназначенным для описания личных шаблонов использования для пользователя, связанных с меню, и с классом 1512 "узел меню виртуального мира телефонного аппарата" (PhoneWorldMenuNode), обеспечивающим узлы меню для вышеописанного графа. Обмен информацией с классом 1504 "узел виртуального мира телефонного аппарата" (PhoneWorldNode) также осуществляет класс 1514 "концевой узел древовидной схемы виртуального мира телефонного аппарата" (PhoneWorldLeafNode), обеспечивающий поддержку вышеописанного графа.

Описанная система обеспечивает три уровня алгоритмов адаптивного интерфейса пользователя. Первый уровень индивидуальной настройки предпочтительно содержит алгоритм предложения ярлыка активации пункта меню. Этот алгоритм осуществляет мониторинг активизации пунктов меню с использованием средства 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper). Алгоритм осуществляет текущий контроль среднего количества случаев активизации пункта меню. Когда количество случаев активизации некоторого пункта меню превышает пороговое значение (которым, возможно, но не обязательно, является вышеупомянутое среднее значение), а расстояние при активизации ярлыка является меньшим, чем требуемое для активизации самого пункта меню, то предпочтительно предлагают ярлык. Пользователь получает преимущество от наличия автоматического ярлыка, так как это уменьшает количество операций, выполняемых пользователем для активизации желаемой функции. Построенная на основании этого процесса генерация действия представляет собой стратегию, основанную на правилах, которая использует средство 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper) в качестве ее базы знаний. Этот алгоритм автоматически осуществляет индивидуальную настройку ярлыков, ориентированных на конкретного пользователя.

Второй уровень индивидуальной настройки предпочтительно содержит переупорядочение пунктов меню. Алгоритм осуществляет мониторинг случаев активизации пунктов меню с использованием средства 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper) и переупорядочивает пункты меню в конкретном меню согласно количеству случаев активизации таким образом, что наиболее часто используемые элементы оказываются расположенными первыми по порядку. Этот алгоритм осуществляет индивидуальную настройку порядка расположения пунктов меню, адаптируясь к их использованию конкретным пользователем. Этот алгоритм для переупорядочения пунктов меню предпочтительно использует ту же самую базу знаний о случаях активизации, что и предыдущий алгоритм.

Третий уровень индивидуальной настройки предпочтительно содержит построение меню. Алгоритм осуществляет мониторинг использования пунктов меню и меню и выбирает наиболее часто используемые пункты меню. Для этих пунктов меню

алгоритм выбирает первый общий узел в графе средства 1502 отображения виртуального мира телефонного аппарата (PhoneWorldMapper). Этот узел становится самым меню, и наиболее используемые пункты меню становятся пунктами меню этого узла. Это меню предпочтительно расположено в системе меню первым по порядку.

5 Это также приводит к изменению графа средства отображения виртуального мира телефонного аппарата (PhoneWorldMapper) на новый граф, представляющий изменение в системе меню. Предпочтительно, этот алгоритм выполняет итерации и создает меню в убывающем порядке по частоте активизации пунктов меню.

10 Раздел 7: Адаптивная система для игр

В этом разделе описан предпочтительный вариант осуществления адаптивной системы для игр согласно настоящему изобретению. На фиг.16 показан пример схемы классов игр согласно настоящему изобретению.

15 Некоторые из целевых установок интеллектуального агента предпочтительно, но не обязательно, предназначены для развлечения пользователя. Интеллектуальный агент также может, но не обязательно, иметь личные целевые установки, например, для обмена информацией.

Для управления состояниями этих целевых установок система предпочтительно имеет в схеме 1600 классов игр, которая показана на фиг.16, два класса: класс 1602 "обработчик состояния "пользователь скучает"" (UserBoredStateHandler) и класс 1604 "обработчик состояния виртуального существа" (CreatureStateHandler). Оба класса предпочтительно осуществляют генерацию действий согласно стратегии на основе правил. Правила, поддержку которых обеспечивают вышеупомянутые классы,

20 относятся к отображаемым ими целевым установкам. Оба класса используют модель событий в качестве способа ввода данных для оценки правил и изменения состояния (то есть, оба класса являются обработчиками событий).

В качестве действия при простое (Idle action) (то есть, если он не задействован каким-либо иным способом) интеллектуальный агент предпочтительно выбирает действие "перемещение" (MoveAction) (на чертеже не показано), которое также может быть адаптировано к предпочтениям пользователя, например, относящимся к анимации, звукам и т.д.

35 Сначала действие "перемещение" (MoveAction) предпочтительно производит выбор между состоянием "ПЕРЕМЕЩЕНИЕ" (MOVE) или состоянием "ПОКОЙ" (REST). Выбор основан на вероятности. В каждом состоянии действие "перемещение" (MoveAction) производит выбор надлежащей анимации для состояния также на основании вероятности. Исходное состояние значений вероятности для каждого варианта выбора устанавливаются равным 50%.

40 Ввод данных пользователем оказывает воздействие на вероятность пары (состояние, анимация), выбранной в текущий момент. Если пользователь вводит неправильные данные, то вероятность состояния и вероятность текущей анимации уменьшается, а для правильно введенных данных она увеличивается. Значения вероятности предпочтительно имеют определенные минимальное и максимальное пороговые значения для предотвращения возможности того, что определенное состояние или определенная анимация никогда не смогут быть выбраны.

Также показано действие 1606 "обмен информацией" (CommAction). Этим действием управляет целевая установка, инициируя обмен информацией, и возможно, но не обязательно, его генерацию осуществляет обработчик 1604 состояния виртуального существа (CreatureStateHandler) в зависимости от предпочтений пользователя относительно выразительности и обмена информацией и от состояния обмена

50

информацией, в котором находится интеллектуальный агент. Например, если интеллектуальный агент не производил обмен информацией с пользователем в течение некоторого времени, а текущий момент времени является подходящим моментом времени для того, чтобы предпринять попытку обмена информацией с пользователем (согласно базе правил обработчика состояния), то предпочтительно осуществляют генерацию действия обмена информацией. Это действие может вызвать вибрацию и/или звук, а также, когда это возможно, то может быть использован необязательный обмен текстовой информацией.

Управление действием отображения поведения предпочтительно, но не обязательно, осуществляет модель формирования эмоций; всякий раз, когда эмоциональное состояние изменяется, интеллектуальный агент предпочтительно выражает новое эмоциональное состояние, возможно, но не обязательно, с использованием текста, звука, двумерной и трехмерной анимации.

Действие 1608 "игра" (GameAction) предпочтительно производит запуск игры в пространстве "плавающего" приложения. Это действие возможно, но не обязательно, выбирает один или большее количество объектов из приложения "виртуальный мир искусственного интеллекта". Интеллектуальный агент исследует объект и действует на него. Например, может быть выбран объект "мяч", после чего интеллектуальный агент может переместиться и ударить по мячу, пользователь может переместить мяч в новое место и т.д. Некоторые из объектов могут, но не обязательно, представлять собой объекты, "упакованные" в интерфейс пользователя (пояснение которых приведено в описании приложения "виртуальный мир системы искусственного интеллекта"). Это действие "игра" предпочтительно отличается тем, что только интеллектуальный агент принимает решение о выборе возможного действия без его подкрепления пользователем.

Действие 1610 "игра в прятки" (HideAndSeek) использует способность средства отображения виртуального мира телефонного аппарата (PhoneWorldMapper) отслеживать местоположение пользователя в системе меню и в различных изображениях, выводимых на главный экран. Интеллектуальный агент предпочтительно выбирает местоположение в дереве меню и прячется, после чего пользователь просматривает систему меню до тех пор, пока пользователь не найдет интеллектуального агента или не завершится время, отведенное на поиск. После того, как интеллектуальный агент найден (или не найден) пользователем, предпочтительно выводят сообщение, в котором пользователю сообщают некоторые сведения о текущем местоположении в системе меню и/или некоторые полезные сведения о текущем изображении, выведенном на экран. Таким образом, пользователь может узнать об особенностях и об иных возможностях, доступных в главной платформе. Полезные подсказки предпочтительно являются доступными для интеллектуального агента посредством класса 1504 "узел виртуального мира телефонного аппарата" (PhoneWorldNode), который содержит подсказки, соответствующие конкретному узлу, описанному экземпляром объекта из этого класса.

Класс 1612 "предложить обучающую машину на основе игры-викторины" (SuggestTmTrivia) может, но не обязательно, предоставлять пользователю игру-викторину (trivia game), предпочтительно по той теме, к которой пользователь проявил интерес.

Раздел 8: Обучающая система

В этом разделе описан предпочтительный вариант осуществления обучающей системы согласно настоящему изобретению, в том числе предпочтительный вариант

осуществления настоящего изобретения для обучения пользователя знаниям о предмете, который не является непосредственно связанным с работой самого устройства. Приведено общее описание обучающей машины, после которого
 5 обучающей машины согласно фиг.17А (пример схемы классов обучающей машины) и фиг.17В (пример схемы последовательности операций, выполняемых обучающей машиной).

Вышеописанный уровень приложений предпочтительно использует
 10 инфраструктуру обучающей системы для создания различных обучающих приложений в рамках настоящего изобретения.

Обучающая машина предпочтительно является способной осуществлять обработку и/или обеспечивать поддержку таких аспектов обучения и получения знаний, как содержание, логика обучения, запоминание, обновления, взаимодействия с
 15 интеллектуальным агентом (при его наличии), построение занятий, произношение (если нужно произносить или понимать слова, представленные в звуковой форме). Последний вопрос является особенно важным для обучения языкам, поскольку для каждого языка необходимо запоминать следующие данные: языковые определения (название, алфавит, гласные и т.д.); правила (грамматика, синтаксис) и словарь.
 20 Правило предпочтительно представляет собой простой элемент языка, которому можно обучить на примерах и который также может быть легко проверен. Словарь предпочтительно задан в виде наборов слов, в которых каждое заданное слово предпочтительно имеет некоторый уровень и также может быть, но не обязательно, распределено по категориям согласно различным критериям (например, слова, связанные с работой, слова, связанные с путешествиями, простые разговорные слова и т.д.). Одним из других важных аспектов является, в том числе, контекст, при этом для
 25 каждого слова w в словаре должно иметься, по меньшей мере, 3 контекста и взаимосвязи, поэтому для каждого из слов w_1 , w_2 в словаре должен существовать максимальный набор взаимосвязей. Взаимосвязь предпочтительно определяют как набор из 4 слов " $w_1:w_2$ тождественны $w_3:w_4$ ".
 30

Архитектура обучающей машины высокого уровня предпочтительно содержит класс, именуемый "язык обучающей машины" (TMLanguage), который обеспечивает
 35 абстракцию для текущего языка обучающей машины, ОМ (ТМ), предоставляет возможность расширения для всей инфраструктуры ОМ. Предпочтительно также имеется класс, определенный как "занятие с обучающей машиной" (TMLesson), предназначенный для организации отдельных занятий, например, в соответствии со
 40 словами из набора, с правилами, с контрольными тестами или вопросами о практическом применении и т.д.

Периодичность занятий возможно, но не обязательно, определяют как еженедельную. Занятие составлено из: набора слов, который представляет собой
 45 текущий словарь для этого занятия, набора правил, который может содержать одно или большее количество правил, изучаемых на этом занятии, практических занятий для того, чтобы предоставить пользователю возможность практического применения изучаемого материала, и, возможно, но не обязательно, контрольного теста.

На фиг.17А показан пример схемы 1700 классов обучающей машины для
 50 инфраструктуры обучающей машины, которая выполнена таким образом, что обеспечивает расширяемую структуру для типовых и адаптивных обучающих приложений. Класс 1702 приложений, именуемый "приложение "обучающая машина"" (TeachingMachineApp), является ответственным за обеспечение времени выполнения и

интерфейса пользователя для приложения, основанного на контрольном тесте. В приложении предпочтительно встроено средство 1704 "обучающая машина" (TMEngine), которое является ответственным за формирование профиля пользователя (модели пользователя) в анализируемом поле. Например, если общим полем является словарь английского языка, то средство 1704 "обучающая машина" (TMEngine) предпочтительно производит сбор данных о проценте успешных результатов, достигнутых пользователем, в различных подполях английского словаря с точки зрения взаимосвязей между словами, отрицания, функции, темы и т.д.

После анализа успеваемости пользователя в различных подполях общего поля, обучение которому осуществляет приложение, средство 1704 "обучающая машина" (TMEngine) предпочтительно управляет приложением таким образом, чтобы оно произвело проверку и улучшение знаний пользователя по тем темам и под полям, в которых успеваемость была более слабой. Средство 1704 "обучающая машина" (TMEngine) предпочтительно запускает циклы оценки знаний пользователя, сопровождаемые обучением и адаптацией к успеваемости пользователя, для генерации вопросов контрольного теста, соответствующих новому состоянию пользователя.

Средство 1704 "обучающая машина" (TMEngine) также собирает сведения об успеваемости пользователя с течением времени и может, но не обязательно, снабжать приложение 1702 "обучающая машина" (TeachingMachineApp) статистикой, относящейся к проценту успешных результатов, достигнутых пользователем.

Расширяемая структура контрольного теста предпочтительно обеспечена за счет использования уровней абстракции и интерфейсов. Средство 1704 "обучающая машина" (TMEngine) предпочтительно представляет собой хранилище контрольных вопросников; контрольные тесты могут быть, но не обязательно, реализованы незаметно для пользователя, так как все контрольные тесты предпочтительно реализуют стандартный интерфейс контрольного вопросника 1706 обучающей машины (TMQuiz). Каждый контрольный вопросник может осуществлять доступ к соответствующей ему базе данных с вопросами, ответами и значениями процентов успешных результатов, достигнутых пользователем, и запоминание этой базы данных с использованием класса 1708 "доступ к данным обучающей машины" (TMDataAccess). Контрольные вопросники и аспекты обучения по конкретной теме, имеющиеся в обучающей машине, предпочтительно являются разделенными, что обеспечивает возможность работы адаптивного обучающего приложения с множеством тем различных типов и высокую степень его расширения.

Примерами некоторых контрольных тестов различных типов являются, в том числе, контрольный тест 1710 на знание сети виртуального мира обучающей машины (TMWorldNet), контрольный тест 1712, проводимый обучающей машиной в виде игры-викторины (TMTriviaQuiz), и контрольный тест 1714 на знание связей, проводимый обучающей машиной (TMRelationQuiz).

На фиг.17В показан пример схемы последовательности операций обучения согласно настоящему изобретению. Администратор 502 приложений (более подробное описание которого приведено при пояснении фиг.5) передает команду "выполнить шаг программы" в приложение 1702 "обучающая машина" (TeachingMachineApp) (стрелка 1). Затем приложение 1702 "обучающая машина" (TeachingMachineApp) передает в средство 1704 "обучающая машина" (TMEngine) запрос на подготовку следующего этапа обучения (стрелка 1.1). Эту подготовку предпочтительно начинают путем выдачи запроса на получение следующего вопроса (стрелки 1.2.1 и 1.2.1.1) из контрольного вопросника 1706 обучающей машины (TMQuiz). Получают ответ от

пользователя и производят его оценку (стрелка 1.2.2) средством 1704 "обучающая машина" (TMEngine) и контрольным вопросником 1706 обучающей машины (TMQuiz). Если ответ правильный, то контрольный вопросник 1706 обучающей машины (TMQuiz) обновляет отсчет правильных ответов (стрелка 1.2.2.1.1.1), в противном случае он обновляет отсчет неправильных ответов (стрелка 1.2.2.1.2.1), при этом также обновляют общий процент успешных результатов. Средство 1704 "обучающая машина" (TMEngine) предпочтительно запоминает статистику контрольного теста. Если был выбран неправильный ответ, то предпочтительно, но не обязательно, отображают правильный ответ для пользователя.

Следующая часть последовательности может быть, но не обязательно, выполнена в том случае, если ранее уже было проведено, по меньшей мере, однократное тестирование пользователя. Администратор 502 приложений снова передает команду "выполнить шаг программы" (стрелка 2). Приложение 1702 "обучающая машина" (TeachingMachineApp) передает запрос на подготовку этапа обучения (стрелка 2.1). Определяют тему, которую пользователь знает хуже всего (стрелка 2.1.1), и также предпочтительно определяют самый простой контрольный тест для пользователя (стрелка 2.1.1.2). Для каждого вопроса на данном этапе обучения приложение 1702 "обучающая машина" (TeachingMachineApp) предпочтительно получает следующий вопрос вышеописанным способом и, как описано выше, производит оценку ответа пользователя.

Эта архитектура предпочтительно является расширяемой для новых тем, а также для новых структур контрольных тестов. Новые темы предпочтительно содержат общую тему (например, "английский язык") и тип содержания (например, "американский сленг" или "слова, связанные с путешествиями"). Предпочтительно, тема содержит данные для этой темы, а также структуру контрольных тестов, вследствие чего обучающая машина может автоматически объединять данные со структурой контрольных тестов. Каждый контрольный тест предпочтительно основан на шаблоне контрольного теста, при этом имеются инструкции в отношении данных, которые могут быть, но не обязательно, помещены в конкретное место (в конкретные места) внутри шаблона.

ПРИМЕР 3: ЭВОЛЮЦИОННАЯ СИСТЕМА ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА

В этом примере описан предпочтительный вариант осуществления эволюционной системы согласно настоящему изобретению, в том числе, описание ДНК для виртуального существа или виртуального персонажа согласно предпочтительному варианту осуществления настоящего изобретения, а также описание необязательной "мастерской генов" (gene studio) согласно настоящему изобретению. Эволюционная система предпочтительно, но не обязательно, дает возможность виртуальному существу или виртуальному персонажу "развиваться", то есть изменять, по меньшей мере, один аспект поведения и/или внешнего вида виртуального существа. Этот пример описан как предпочтительно, но необязательно, действующий совместно с интеллектуальным агентом, описанным в Примере 2, при этом подразумевают, что это описание приведено только лишь в иллюстративных целях и никоим образом не является ограничивающим признаком.

Эволюция (изменение) интеллектуального агента описана здесь как относящаяся к обоим характеристикам агента: к осязаемым характеристикам агента, отображаемым посредством виртуального персонажа или виртуального существа, и к неосязаемым характеристикам агента, оказывающим воздействие на поведение виртуального

персонажа или виртуального существа.

На фиг.18А показан пример схемы 1800 классов эволюции. Генетическая модель, описанная в диаграмме классов, позволяет изменять различные свойства интеллектуального агента, которые предпочтительно содержат визуальные, а также функциональные свойства. Модель содержит класс 1802 "ДНК виртуального существа" (CreatureDNA), который отображает структуру ДНК. Структура ДНК представляет собой вектор имеющихся генов и предпочтительно может быть расширена таким образом, чтобы включать в себя новые гены. Ген представляет собой параметр, имеющий диапазон возможных значений (то есть, генотип). Система согласно настоящему изобретению интерпретирует ген таким образом, что выражение данных в гене представляет собой его генотип. Например, в качестве первого гена в ДНК расположен ген головы, и его значение выражено в виде визуальной структуры головы виртуального существа, хотя предпочтительно цвет головы закодирован в другом гене.

Для обеспечения такого развития интеллектуального агента, которое приводит к получению конкретного экземпляра ДНК, нравящегося пользователю, в генетической модели согласно настоящему изобретению предпочтительно реализованы генетические операции гибридизации и мутации, изменяющие ДНК. Класс 1804 "посредник виртуального существа" (CreatureProxy) является ответственным за обеспечение интерфейса для ДНК и генетических операций для системных классов. Класс 1804 "посредник виртуального существа" (CreatureProxy) предпочтительно также хранит другую негенетическую информацию об интеллектуальном агенте (то есть имя, дату рождения и т.д.).

Класс 1806 "администратор эволюции" (EvolutionMGR) предпочтительно управляет эволюцией (развитием) интеллектуального агента и обеспечивает интерфейс для класса 1804 "посредник виртуального существа" (CreatureProxy) интеллектуального агента и его генетические операции с приложениями.

Класс 1808 "двигатель эволюции" (EvolutionEngine) отслеживает события, связанные с эволюцией, генерация которых может производиться время от времени, для указания того, что должна быть вызвана и выполнена определенная генетическая операция над ДНК интеллектуального агента. Структура ДНК приведена ниже.

Класс 1802 "ДНК виртуального существа" (CreatureDNA) предпочтительно отслеживает такие события, связанные с эволюцией, из класса 1810 "событие, связанное с эволюцией" (EvolutionEvent).

Структура ДНК

```
#ifndef __CREATURE_DNA_
#define __CREATURE_DNA_
#include "CreatureDefs.h"
#include "CommSerializable.h"
#define GENE_COUNT 19
#define BASE_COLOR_GENE 8
typedef struct internal_dna
{
    unsigned char head;
    unsigned char head_color;
    unsigned char head_scale;
    unsigned char body;
    unsigned char body_color;
```

```

unsigned char body_scale;
unsigned char hand;
unsigned char hand_color;
unsigned char hand_scale;
5 unsigned char tail;
unsigned char tail_color;
unsigned char tail_scale;
unsigned char leg;
10 unsigned char leg_color;
unsigned char leg_scale;
unsigned char dexterity;
unsigned char efficiency;
unsigned char interactive;
15 unsigned char base_color;
    } internal_dna;
typedef internal_dna p_internalDna;
    /**
20  * Этот класс отображает структуру ДНК виртуального существа.
    * ДНК хранит все данные о частях тела виртуального существа и
    * о некоторых индивидуальных особенностях и функциональных
    * свойствах
    */
25 class CreatureDNA /*: public CommSerializable*/
    {
    public:
    static const int gene_count;
    /**
30  * конструктор по умолчанию, устанавливают исходное
    * состояние ДНК, равное нулю
    */
    CreatureDNA();
35  /**
    * Копирование конструктора
    * иной параметр @param - ДНК для копирования
    */
    CreatureDNA(const CreatureDNA & other);
40  /**
    * Функция инициализации, ее следует вызвать, если
    * не был вызван конструктор.
    */
45  void init();
    /**
    * Рандомизация данных ДНК
    *
    */
50  void randomizeDna();
    /**
    * Фактические данные ДНК

```

```

*/
union {
internal_dna genes;
5  unsigned char data[GENE_COUNT];
};
/**
* Интервал значений для гена "тип"
*/
10 static const int TYPE_RANGE;
/**
* Интервал значений для гена "цвет"
*/
15 static const int COLOR_RANGE;
/**
* Интервал значений для гена "масштаб"
*/
20 static const int SCALE_RANGE;
/**
* Интервал значений для гена "характер"
*/
25 static const int CHARECTER_RANGE;
static const int BASE_COLOR_RANGE;
private:
/**
* Местоположение гена "масштаб" в
* триплете "тип, цвет, масштаб"
*/
30 static const int SCALE_LOCATION;
};
#endif /* __CREATURE_DNA__ */

```

35 Конструирование ДНК интеллектуального агента предпочтительно выполняют следующим образом. При создании варианта "живого" мобильного телефона ДНК предпочтительно составлена из гена для каждого элементарного блока интеллектуального агента. Элементарный блок может, но не обязательно, являться видимой частью агента, предпочтительно содержащей цвет или масштаб (размер элементарного блока), а также предпочтительно содержащим свойство, не

40 воспринимаемое зрением, которое относится к функциональным возможностям и к поведению интеллектуального агента. Эта модель структуры ДНК может быть расширена по мере возможного добавления большего количества элементарных блоков и увеличения количества уровней выражения каждого элементарного блока.

45 Построение интеллектуального агента из структуры ДНК предпочтительно выполняют применительно к каждому гену и его значению. Значение (уровень выражения) каждого гена (элементарного блока) описывает различный генотип, выраженный в скомпонованном агенте. Основные элементарные блоки агента, видимого зрителем, смоделированы как прототипы, следовательно, интервал значений каждого гена воспринимаемых зрением свойств определяется количеством прототипов. Также возможно осуществлять генерацию в значениях времени

50 выполнения выраженных генов, не основываясь на прототипах, например, для

получения генотипов, которые являются независимыми от заранее заданных прототипов, уровни выражения гена "цвет" могут быть вычислены как индексы в таблице цветов главной платформы, или масштаб может быть также вычислен относительно размера главного экрана. Модели прототипа предпочтительно подвергают декомпозиции, а затем осуществляют повторную компоновку агента, не являющегося прототипом, предпочтительно согласно значениям гена каждого элементарного блока.

Приведенный ниже пример обеспечивает иллюстративное объяснение этого процесса, не являющееся ограничивающим признаком. Для простоты и доходчивости он не содержит генов цвета и масштаба и иных генов свойств, не воспринимаемых зрением, но тот же самый процесс также применим и для этих генов.

Вариант ДНК из 16 прототипов и 5 элементарных блоков может быть, но не обязательно, задан следующим образом:

$$\text{ДНК}_0 = \{[\text{голова}, 0:15], [\text{тело}, 0:15], [\text{ноги}, 0:15], [\text{руки}, 0:15], [\text{хвост}, 0:15]\}$$

Каждый из пяти элементарных блоков имеет 16 различных возможных генотипов согласно значениям гена элементарного блока, которые выведены из количества моделей прототипа. При построении интеллектуального агента правильный элементарный блок берут согласно значению этого элементарного блока в ДНК, которое представляет собой значение соответствующего ему гена.

Например, конкретным экземпляром для вышеописанной схемы ДНК может являться:

$$\text{ДНК} = \{[3], [5], [10], [13], [0]\}$$

Количество разнообразных возможных вариантов построения интеллектуального агента в этой простой версии ДНК равно:

$$V_0 = (16) \cdot (16) \cdot (16) \cdot (16) \cdot (16) = (16)^5 = 1048576$$

Если добавлен ген основного цвета, предназначенный для описания общего цвета интеллектуального агента (то есть, зеленого, синего и т.д.), с уровнем выражения из 16 возможных основных цветов, то получают следующее множество:

$$\text{ДНК}_1 =$$

$$\{[\text{голова}, 0:15], [\text{тело}, 0:15], [\text{ноги}, 0:15], [\text{руки}, 0:15], [\text{хвост}, 0:15], [\text{основной_цвет (bs_color)}, 0:15]\}$$

Количество разнообразных возможных вариантов в этом случае становится равным:

$$V_1 = V_0 \cdot 16 = (16)^6 = 16777216$$

Если к этой версии ДНК добавлен ген интенсивности для гена основного цвета (то есть, от светлого цвета до темного цвета), с уровнем выражения из 16 возможных интенсивностей основного цвета, то предпочтительно получают следующее множество:

$$\text{ДНК}_2 =$$

$$\{[\text{голова}, 0:15], [\text{тело}, 0:15], [\text{ноги}, 0:15], [\text{руки}, 0:15], [\text{хвост}, 0:15], [\text{основной_цвет (bs_color)}, 0:15], [\text{интенсивность}, 0:15]\}$$

Вычисленное количество разнообразных возможных вариантов в этом случае равно:

$$V_2 = V_1 \cdot 16 = (16)^7 = 268435456$$

Как описано со ссылкой на фиг.18В и фиг.18С, на которых показаны, соответственно, схема последовательности операций мутации и схема последовательности операций гибридизации, с ДНК может быть, но не обязательно,

выполнено множество генетических операций.

Как показано на фиг.18В, основная операция мутации предпочтительно выбирает по случайному закону из набора генов, который может, но не обязательно, являться всей ДНК, тот ген, который может быть видоизменен, и затем изменяет значение
5 выбранного гена в пределах возможного интервала значений этого гена (уровней выражения). Основная операция может быть, но не обязательно, выполнена многократно.

Приложение 1812 "мутация" посылает в класс 1806 "администратор эволюции" (EvolutionMGR) (стрелка 1.1) запрос на создание мутанта. Класс 1806 "администратор эволюции" (EvolutionMGR) передает этот запрос посреднику 1804 виртуального
10 существа (CreatureProxy), возможно, но не обязательно, для создания нескольких мутантов (это значение может быть задано в вызове функции; стрелка 1.1.1). Для каждого такого мутанта посредник 1804 виртуального существа (CreatureProxy)
15 предпочтительно выбирает случайный ген (стрелка 1.1.1.1) и изменяет его на значение, которое все же находится в пределах интервала значений гена (стрелка 1.1.1.2). Мутанта (мутантов) затем возвращают в приложение 1812 "мутация" и предпочтительно отображают для пользователя, а более подробное
20 описание этого приведено ниже в описании Примера 4.

Если пользователь одобряет мутанта, то затем приложение 1812 "мутация" передает в класс 1806 "администратор эволюции" (EvolutionMGR) команду на замену существующей реализации агента новым мутантом (стрелка 2.1). Затем класс 1806
25 "администратор эволюции" (EvolutionMGR) устанавливает ДНК для виртуального существа в посреднике 1804 виртуального существа (CreatureProxy) (стрелка 2.1.1), который затем предпочтительно обновляет предысторию агента в "предыстории_агента" (agent_history) 1814 (стрелка 2.1.1.1).

На фиг.18С показан пример схемы последовательности операций для основной
30 операции гибридизации (или операции скрещивания), которая происходит тогда, когда две ДНК претендентов совмещены одна с другой. Предпочтительно, выбирают одну или большее количество точек пересечения, расположенных на векторе ДНК (количество точек пересечения может изменяться от 1 до количества генов в ДНК; это количество может, но не обязательно, быть выбрано по случайному закону).
35 Операцию выбора точек пересечения именуют get_cut_index ("получить_индекс_вырезки"). В каждой точке пересечения выбирают значение для ДНК из одного из существующих значений ДНК. Это может быть, но не обязательно, выполнено по случайному закону или согласно отсчету, именуемому
40 "индекс_вырезки" (cutting_index). Результатом является смешение между двумя ДНК претендентов. Основная операция гибридизации может быть, но не обязательно, выполнена многократно с многочисленными претендентами.

Как показано на чертеже, приложение 1816 "гибридизация" (HybridApp) передает в класс 1806 "администратор эволюции" (EvolutionMGR) команду начать процесс
45 гибридизации. Процесс возможно, но не обязательно, выполняют до тех пор, пока пользователь не одобрит гибридного агента или не прервет процесс. Класс 1806 "администратор эволюции" (EvolutionMGR) начинает гибридизацию путем передачи команды на получение целевой ДНК (стрелка 2.1.1) из класса 1804 "посредник
50 виртуального существа" (CreatureProxy), при этом выполняют несколько скрещиваний (гибридизаций). Как показано на чертеже, "индекс_вырезки" (cutting_index) сохраняют для указания того, когда следует произвести скрещивание между значениями двух ДНК.

Гибридного агента возвращают, и если пользователь одобряет его, то тогда текущего агента заменяют гибридным агентом, как описано выше относительно процесса создания мутанта. В конце предпочтительно производят обновление предыстории агента в "предыстории_агента" (agent_history) 1814.

Гибридизация может быть предпочтительно, но не обязательно, выполнена с ДНК агента, переданной из источника, являющегося внешним относительно мобильного информационного устройства, например, в SMS-сообщении, посредством связи в инфракрасном диапазоне, технологии Bluetooth или сети Интернет, или из любого иного источника. Этот процесс проиллюстрирован применительно к получению такой гибридной ДНК посредством SMS-сообщения, при этом подразумевают, что этот пример приведен только лишь в иллюстративных целях и никоим образом не является ограничивающим признаком. SMS-сообщение предпочтительно содержит данные для ДНК, представленные в стандарте MIME (стандарт многоцелевых расширений электронной почты в сети Интернет). В более предпочтительном варианте система из настоящего изобретения имеет обработчик прерываний для данных этого типа, представленных в стандарте MIME, поэтому SMS-сообщение этого типа предпочтительно подвергают автоматическому анализу для гибридизации, что не требует ручного вмешательства пользователя.

На фиг.19 показан пример схемы последовательности операций такого процесса. Как показано на чертеже, пользователь 1 передает запрос на создание гибрида интеллектуального агента пользователя 1 с интеллектуальным агентом пользователя 2 посредством телефонной трубки 1. Пользователь 2 может по своему выбору одобрить или отклонить запрос посредством телефонной трубки 2. Если пользователь 2 одобряет его, то выполняют операцию гибридизации между ДНК от обоих агентов для телефонной трубки 1. Результат возможно, но не обязательно, отображают для стороны, выдавшей запрос (для пользователя 1), которая может сохранить этот гибрид в качестве замены для текущего агента. Если гибрид использован в качестве замены, то затем пользователь 2 получает уведомление и сохраняет гибрида в коллекции результатов гибридизации в телефонной трубке 2.

ПРИМЕР 4: ВЗАИМОДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С НАСТОЯЩИМ ИЗОБРЕТЕНИЕМ

Этот пример описан со ссылкой на множество типичных, не ограничивающих снимков экрана, приведенных в иллюстративных целях для представления того, каким образом необязательный, но предпочтительный вариант осуществления системы из настоящего изобретения взаимодействует с пользователем.

Фиг.20 показывает приведенный в качестве примера снимок экрана с "плавающим агентом", который представляет собой виртуальное существо или виртуальный персонаж (визуальное выражение интеллектуального агента). На фиг.21 показан приведенный в качестве примера снимок экрана с меню, предназначенным для выбора объектов для виртуального мира интеллектуального агента.

На фиг.22 показано приложение "мастер начала работы" (Start Wizard), которое предоставляет пользователю возможность задавать конфигурацию и изменять установочные параметры агента, а также предпочтения пользователя.

Одним из примеров действия, выполняемого с использованием мастера, является установка индивидуальных особенностей (Set Personality) для задания установочных параметров для системы формирования эмоций интеллектуального агента. Здесь пользователь может задавать конфигурацию индивидуальных особенностей виртуального существа и тенденции.

Пользователь может предпочтительно, но не обязательно, задавать установочные параметры виртуального существа путем нажатия на надлежащую клавишу с изображением стрелки для повышения уровня параметра и для выполнения

5
противоположной операции и уменьшения уровня различных параметров, таких как, например, энтузиазм, общительность, антиобщественное поведение, сдержанность (уровень терпения), меланхолия, эгоистичное поведение и т.д.

Пользователь также предпочтительно является способным устанавливать предпочтения пользователя, например, определять то, насколько быстро следует

10
получать справку. Некоторыми другими не ограничивающими примерами этих предпочтений являются, в том числе: обмен информацией (та степень, в которой агент осуществляет обмен информацией); "развлечение_пользователя" (entertain_user) (управляет агентом, играющим с пользователем); "развлечение_самого_себя" (entertain_self) (управляет агентом, который играет сам с собой);

15
"сбережение_энергии_аккумулятора" (preserve_battery) (продляет срок службы аккумулятора); и "уровень_прозрачности" (transparency_level) (уровень прозрачности виртуального существа).

Посредством "мастера начала работы" (Start Wizard) пользователь также

20
предпочтительно устанавливает подробности о пользователе (User Details), предпочтительно содержащие имя пользователя, день рождения (согласно необязательному варианту осуществления настоящего изобретения это значение является важным в SMS-сообщении для гибридизации, так как оно определяет

25
возможность создания гармоничных отношений (конгапа; (konghup)) между пользователями, то есть способность создания гибрида с благоприятным астрологическим прогнозом; опция "конгапа" (konghup) построена согласно соответствующим таблицам гороскопов и дат) и пол, но эти примеры не являются ограничивающим признаком.

Пользователь также может предпочтительно устанавливать подробности о

30
виртуальном существе (Creature Details).

На фиг.23 показан пример меню для выполнения гибридизации посредством вышеописанного приложения "гибридизация".

На фиг.24А показан приведенный в качестве примера снимок экрана для

35
просмотра нового виртуального существа и для возможной, но не обязательной, его повторной генерации путем нажатия на клавишу "генерация" (Generate), которая дает возможность пользователю осуществлять генерацию виртуального существа по случайному закону. На фиг.24В показано полученное в результате этого виртуальное

40
существо на снимке экрана с клавишей "гибрид" (Hybrid): нажатие на эту клавишу подтверждает выбор виртуального существа пользователем и переход к окну предварительного просмотра виртуального существа.

Окно предварительного просмотра предоставляет пользователю возможность

45
видеть вновь созданное виртуальное существо в трех измерениях и, возможно, но не обязательно, воспроизводить анимацию виртуального существа с использованием следующих клавиш выбора:

1. Навигационная клавиша "вверх" (UP): увеличивает масштаб и уменьшает размер виртуального существа.
- 50
2. Навигационная клавиша "вниз" (DOWN): уменьшает масштаб и увеличивает размер виртуального существа.
3. Навигационная клавиша "влево" (LEFT): переключение между клавишей "подтверждение" ("Ok") и клавишей "назад" ("Back").

4. Навигационная клавиша "вправо" (RIGHT): переключение между клавишей "подтверждение" ("Ok") и клавишей "назад" ("Back").

5. Клавиша "подтверждение" (Ok): подтверждение выбора.

6. Клавиша "сброс" (CLR): выход из окна предварительного просмотра виртуального существа к меню "живого" мобильного устройства (Living Mobile Menu).

7. Клавиша "конец" (End): выход из окна предварительного просмотра виртуального существа в главное меню.

8. Клавиша '0': операция, делающая виртуальное существо более светлым или более темным.

9. Клавиша '1': поворачивает виртуальное существо влево по кругу в направлении часовой стрелки.

10. Клавиша '2': вращает виртуальное существо в трехмерном представлении.

11. Клавиша '3': поворачивает виртуальное существо вправо по кругу в направлении против часовой стрелки.

12. Клавиша '5': вращает виртуальное существо в трехмерном представлении.

13. Клавиша '6': создает анимацию виртуального существа различными способами. Каждое новое нажатие на эту клавишу изменяет тип анимации.

Варианты анимации, которые может выполнять виртуальное существо, возможно, но не обязательно, содержат следующее: ходьба, сидение, нюхание, летание и прыгание, но эти примеры не являются ограничивающим признаком.

На фиг.25 показан приведенный в качестве примера снимок экрана предыстории гибридов, которая дает возможность пользователю просматривать и исследовать предысторию изменений виртуального существа в поколениях. Пользователь может предпочтительно видеть текущее виртуальное существо и его родителей и, возможно, но не обязательно, также родителей его родителей. Предпочтительно, для каждого виртуального существа могут существовать максимум 2 родителя.

Фиг.26 показывает приведенный в качестве примера снимок экрана "мастерская генов" (Gene studio) с последовательностью ДНК текущего виртуального существа. "Мастерская генов" также предпочтительно дает пользователю возможность изменять и модифицировать последовательность ДНК агента.

ПРИМЕР 5: ИНТЕЛЛЕКТУАЛЬНЫЙ АГЕНТ ДЛЯ СЕТЕВОГО МОБИЛЬНОГО ИНФОРМАЦИОННОГО УСТРОЙСТВА

Этот пример относится к использованию интеллектуального агента в сетевом мобильном информационном устройстве, которым предпочтительно является телефонный аппарат сотовой связи. Интеллектуальный агент предпочтительно, но не обязательно, содержит виртуальный персонаж, предназначенный для взаимодействия с пользователем, и агента, предназначенного для взаимодействия с другими компонентами в сети, например, с другими мобильными информационными устройствами, и/или с самой сетью. Следовательно, виртуальный персонаж предпочтительно образует интерфейс пользователя (или его часть) и также имеет внешний вид, который в более предпочтительном варианте является трехмерным. Этот внешний вид может, но не обязательно, быть человекоподобным, но в альтернативном варианте может быть основан на персонаже или на виртуальном существе любого типа, реальном или воображаемом. Следовательно, агент предпочтительно обрабатывает информацию, обмен которой производят между виртуальным персонажем и мобильным информационным устройством и/или другими компонентами в сети и/или другими виртуальными персонажами в других мобильных информационных устройствах. Также следует отметить следующее: несмотря на то,

что этот вариант реализации описан применительно к мобильным информационным устройствам, которыми являются, например, телефонные аппараты сотовой связи, особенность этого варианта реализации, заключающаяся в наличии виртуального персонажа (или даже сам агент), может, но не обязательно, быть реализована с использованием адаптивной системы (Пример 2) и/или интерфейса пользователя с упреждающим действием (Пример 1), описание которых приведено выше.

К интеллектуальному агенту из настоящего изобретения обращаются для создания более сильных эмоциональных переживаний, применяя концепцию "живого устройства" ("Living Device"). Эта концепция предпочтительно содержит оба фактора: фактор уникальности интеллектуального агента, поскольку каждое живущее виртуальное существо является уникальным и особенным по внешности и поведению, и тот фактор, что также обеспечивают разнообразие, например, разнообразие внешнего вида виртуальных персонажей, для усовершенствования взаимодействия пользователя с "живым устройством". Виртуальный персонаж предпочтительно имеет привлекательные визуальные свойства, возможно, но не обязательно, с соответствующими дополнительными объектами и окружающей средой.

Интеллектуальный агент предпочтительно отображает интеллектуальное принятие решения с неожиданным поведением, которое свидетельствует о его существовании как самостоятельного индивида и о независимом обучении. Такое независимое поведение является важной особенностью настоящего изобретения, поскольку это еще не было продемонстрировано ранее ни для интерфейса пользователя какого-либо типа, ни для взаимодействия пользователя с вычислительным устройством какого-либо типа, и, конечно же, не было использовано для интеллектуального агента, предназначенного для мобильного информационного устройства. Также предпочтительно с течением времени происходит эволюция интеллектуального агента, а также всех "живых существ", при этом отображают их визуальное изменение. Это является одним из наиболее важных свойств "живого устройства".

Этап эволюции инициирует возникновение у пользователя эмоциональной реакции удивления и ожидание следующего этапа эволюции.

Эволюция представляет собой визуальное изменение виртуального существа с течением времени. Период времени может быть, но не обязательно, установлен равным, например, одному году, поскольку этот период является сроком службы телефонного аппарата сотовой связи среднего класса на рынке. В течение года периодические изменения предпочтительно происходят за счет эволюции. Ход эволюции (адаптация к окружающей среде) является результатом естественного отбора. Управление естественным отбором может, но не обязательно, осуществлять пользователь (то есть, пользователь принимает решение о том, является ли следующее поколение более совершенным), хотя другим вариантом является заранее заданный процесс естественного отбора, осуществляемый путем создания определенных критериев автоматизированного отбора.

Интеллектуальный агент может быть, но не обязательно, реализован для функционирования в двух "виртуальных мирах" или различных окружающих средах: в виртуальном мире телефонного аппарата и в виртуальном мире виртуального существа. Как описано выше, "виртуальный мир" телефонного аппарата (мобильного информационного устройства) дает возможность интеллектуальному агенту осуществлять управление различными функциями телефонного аппарата и предлагать пользователю варианты выбора различных функций. Интеллектуальный агент предпочтительно является способным функционировать на основании одного или

большого количества процессов использования телефона, которые смоделированы для того, чтобы им следовал агент. Другой важной особенностью виртуального мира телефона являются выражения эмоций, которыми могут являться как выражения, представленные графически, например, временное выключение экрана или создание произвольных рисунков, так и выражения лица и текстовые выражения из одного или двух слов, соответствующие конкретному случаю.

Виртуальный мир предпочтительно представляет собой область для визуального отображения и игр, в которую, возможно, но не обязательно, могут быть введены иные объекты, чем виртуальный персонаж, и пользователь может наблюдать обучение виртуального персонажа и взаимодействие с ними. Объекты, введенные в виртуальный мир, могут, но не обязательно, являться заранее заданными, с различными возможными линиями поведения, возникающими в результате процесса обучения. Пользователь может предпочтительно, но не обязательно, давать поощрения или создавать препятствия и являться частью процесса обучения. В этом отношении, интеллектуальный агент (посредством внешнего вида виртуального персонажа) может, но не обязательно, действовать как какое-либо виртуальное домашнее животное или как компаньон.

Некоторыми предпочтительными особенностями интеллектуального агента являются, в том числе, инфраструктура трехмерной графики (связанная с внешним видом виртуального персонажа); использование механизмов искусственного интеллекта (ИИ) и обучения машины для обеспечения как адаптивного, так и упреждающего поведения; предоставление возможностей проведения игр; способность обеспечения большего удобства использования мобильного информационного устройства, а также конкретной помощи пользователю; и обеспечение уровня абстракций главной платформы, но эти примеры не являются ограничивающим признаком. Эти особенности в совокупности обеспечивают надежную, привлекательную и имеющую новое содержимое платформу, обеспечивающую поддержку множества приложений искусственного интеллекта (ИИ), все из которых, в общем, определены таким образом, что выполняются в мобильном информационном устройстве.

Виртуальный персонаж также предпочтительно имеет несколько важных визуальных особенностей. Например, выводимый клип может, но не обязательно, иметь внешний размер до 60 x 70 элементов изображения, хотя, конечно же, может быть выбрано иное разрешение в соответствии с характеристиками экрана дисплея мобильного информационного устройства. Виртуальный персонаж предпочтительно отображают в виде трехмерного многоугольного объекта, имеющего несколько цветов, но в любом случае он предпочтительно имеет множество различных трехмерных визуальных характеристик, например, оттенки, текстуры, поддержку анимации и т.д. Эти возможности могут быть, но не обязательно, обеспечены посредством заранее созданных визуальных элементарных блоков, хранящихся в мобильном информационном устройстве. Визуально воспринимаемый внешний вид виртуального персонажа предпочтительно формируют во время работы устройства.

Виртуальный персонаж может, но не обязательно, начать "жить" после запуска модуля оперативной помощи, учитывающего предпочтение пользователя (введения пользователя в "живое устройство"). В дополнение к эволюции, виртуальный персонаж может, но не обязательно, отображать незначительные визуальные изменения, отображающие мутации (изменение цвета/перемещение некоторых ключевых вершин со случайным шагом). Визуальный этап эволюции предпочтительно

выполняют путем добавления/замены элементарного блока. Виртуальный персонаж может предпочтительно перемещаться во всех направлениях и вращаться, а в более предпочтительном варианте - он представляет собой полностью анимированный трехмерный персонаж.

Виртуальный персонаж предпочтительно показан как "плавающий" по дисплею мобильного информационного устройства, при этом интерфейс пользователя мобильного информационного устройства находится на заднем плане, но он также может быть, но не обязательно, удален по запросу пользователя. Виртуальный персонаж предпочтительно является способным понимать обычное текущее взаимодействие пользователя с мобильным информационным устройством и пытается свести к минимуму принудительное скрывание/удаление пользователем.

Согласно необязательным, но предпочтительным вариантам осуществления настоящего изобретения, виртуальный персонаж может быть запрограммирован таким образом, что "двигается" по экрану более естественным, физически реалистичным способом. Например, имеются различные типы алгоритмов и параметров, посредством которых пытаются описать физически реалистичное поведение и движение для управления движением роботов. Примеры таких алгоритмов и параметров описаны в публикации "Automatic Generation of Kinematic Models for the Conversion of Human Motion Capture Data into Humanoid Robot Motion", A. Ude et al., *Proc. First IEEE-RAS Int. Conf. Humanoid Robots (Humanoids 2000)*, Cambridge, MA, USA, September 2000 (которая, тем самым, включена сюда в полном объеме путем ссылки). В этой ссылке описаны различные способы регистрации данных о движениях человека и способы автоматического преобразования зарегистрированных данных в кинетические параметры человекоподобного робота. В кратком изложении, производят моделирование движений как человека, так и робота, и используют эти модели для преобразования данных о реальных движениях человека в данные, которые могут быть использованы для управления движениями человекоподобных роботов.

Ссылка этого типа является полезной, поскольку она предоставляет информацию о том, как моделировать движение человекоподобного робота. Хотя настоящее изобретение имеет отношение к реалистичному движению виртуального персонажа (виртуального персонажа, изображаемого в трех измерениях), подобные модели могут быть, но не обязательно, использованы не только для человекоподобного робота, но для виртуального персонажа. Кроме того, также может быть, но не обязательно, создана модель для моделирования движений животных, посредством чего обеспечивают возможность получения более реалистичных движений виртуального персонажа в виде животного или подобного животному. В более общем изложении, система предпочтительно может осуществлять типовую обработку любого заданного набора трехмерных символьных данных.

Эти модели предпочтительно, но не обязательно, также могут быть использованы для обеспечения возможности эволюции движений виртуального персонажа, поскольку в течение эволюционного процесса могут, но не обязательно, изменяться различные параметры модели, изменяя, тем самым, характер движений виртуального персонажа. Такие модели также предпочтительно являются полезными для описания недетерминированных движений виртуального персонажа, а также, возможно, но не обязательно, для обеспечения возможности эволюции недетерминированных движений. Такое недетерминированное поведение также помогает поддерживать интерес пользователя.

Согласно другим предпочтительным вариантам осуществления настоящего изобретения создание поведения виртуального персонажа и управление этим поведением также предпочтительно, но не обязательно, осуществляют согласно недетерминированной модели. Такие модели могут быть, но не обязательно, написаны на известном языке описания поведения, например, на языке ABL (язык описания поведения), как описано в публикации "A Behavior Language for Story-Based Believable Agents", M. Mateas and A. Stern, *Working Notes of Artificial Intelligence and Interactive Entertainment*, AAAI Spring Symposium Series, AAAI Press, USA, 2002 (которая, тем самым, включена сюда в полном объеме путем ссылки). В этой ссылке описан язык описания поведения (ABL), который может быть использован для создания виртуальных персонажей, которые имеют реалистичное поведение. Такие реалистичные линии поведения содержат реагирование на ввод данных, например, в речевой форме, а также движение и/или жесты, все из которых обеспечивают реалистичную связь с человеком-пользователем программного обеспечения. Следует отметить, что под термином "движение" не обязательно подразумевают моделирование движения, которое является реалистичным по внешнему виду, а скорее, подразумевают обеспечение поддержки движения, которое является реалистичным с точки зрения ситуации, в которой оно происходит.

Такой язык, который содержит различные операции ввода и вывода и который может быть использован для моделирования и для обеспечения поддержки реалистичного, недетерминированного интерактивного поведения с человеком-пользователем, может быть, но не обязательно, использован для формирования поведения виртуального персонажа согласно настоящему изобретению. Например, язык описывает "средства выражения элементарных фрагментов драматургического действия" ("beat idioms"), которые являются примерами выразительного поведения. Эти средства выражения элементарных фрагментов драматургического действия разделены на три категории: задачи элементарных фрагментов драматургического действия ("beat goals"), обработчики и перекрестные линии поведения между элементарными фрагментами драматургического действия ("cross-beat behaviors"). Задачи элементарных фрагментов драматургического действия представляют собой линии поведения, которые должны быть выполнены применительно к конкретной ситуации, например, для приветствия человека-пользователя. Обработчики являются ответственными за взаимодействия между человеком-пользователем и виртуальным существом (например, виртуальным персонажем из настоящего изобретения), или за взаимодействия между виртуальными существами. Перекрестные линии поведения между элементарными фрагментами драматургического действия позволяют виртуальному существу перемещаться между наборами линий поведения или средств выражения элементарных фрагментов драматургического действия. Понятно, что такие конструкции в рамках языка могут быть, но не обязательно, использованы для виртуального персонажа согласно настоящему изобретению.

Конечно же, следует отметить, что язык описания поведения (ABL) является только лишь одним не ограничивающим примером возможного языка агента, вместо языка ABL и/или в комбинации с языком ABL могут быть, но не обязательно, использованы языки и/или модели иных типов.

Виртуальный персонаж также предпочтительно имеет несколько выражений эмоций, которые не обязательно должны отображаться как выражение лица, но вместо этого могут быть созданы посредством анимации или текста, таких как, например,

счастливым, грустным, удивленным, огорченным, обиженным или скучающим.

Выражения эмоций могут быть объединены.

Виртуальный персонаж также может быть представлен как изменяющий внешний вид экрана, пишущий текст для пользователя и/или воспроизводящий звуки через телефонный аппарат, это предпочтительно осуществляют за счет функционирования интеллектуального агента. Агент также может, но не обязательно, активизировать режим вибрации, например, в том случае, когда виртуальный персонаж сталкивается с твердыми объектами в виртуальном мире, или в том случае, когда пытается привлечь внимание пользователя. Виртуальный персонаж также может, но не обязательно, иметь такой вид, что он активно манипулирует экранами интерфейса пользователя телефонного аппарата.

Для реализации этих различных функций виртуального персонажа и/или интеллектуального агента интеллектуальный агент предпочтительно, но не обязательно, может быть создан так, как показано на фиг.27-30, хотя следует отметить, что на этих иллюстрациях представлен только лишь один вариант реализации, который приведен в качестве примера, и что возможно множество иных вариантов реализации. И вновь, вариант реализации интеллектуального агента может, но не обязательно, содержать варианты реализации, описанные в приведенных выше Примерах 1 и 2, или быть основанным на них.

На фиг.27 показана блок-схема системы 2700 интеллектуального агента согласно настоящему изобретению. Как показано на чертеже, первый пользователь 2702 управляет первым мобильным информационным устройством 2704, которое в этом примере может, но не обязательно, быть реализовано в виде телефонного аппарата сотовой связи, при этом подразумевают, что этот пример приведен только лишь в описательных целях и никоим образом не является ограничивающим признаком. Второй пользователь 2706 управляет вторым мобильным информационным устройством 2708. Первое мобильное информационное устройство 2704 и второе мобильное информационное устройство 2708 предпочтительно осуществляют обмен информацией через сеть 2710, например, путем обмена сообщениями.

Каждое из устройств: первое мобильное информационное устройство 2704 и второе мобильное информационное устройство 2708 предпочтительно содержат интеллектуальный агент, предназначенный для взаимодействия с соответствующими пользователями 2702 и 2706, а также для взаимодействия с другим интеллектуальным агентом. Следовательно, как показано на чертеже, система 2700 предоставляет возможность сообществу таких интеллектуальных агентов взаимодействовать друг с другом и/или получать информацию для их соответствующих пользователей, например, через сеть 2710.

Взаимодействия пользователей 2702 и 2706 с их соответствующими мобильными информационными устройствами 2704, 2708 предпочтительно содержат обычный режим работы мобильного информационного устройства, но также добавляют и новые увлекательные функциональные возможности "живого мобильного телефона". Эти функциональные возможности предпочтительно содержат интеллектуальный агент, но также и использование виртуального персонажа для обеспечения интерфейса пользователя, а также в более предпочтительном варианте - для обеспечения более глубоких эмоциональных переживаний пользователя.

Интеллектуальный агент предпочтительно содержит структуру "знаний" и интеллектуального программного обеспечения. Внутреннее функционирование такой системы предпочтительно содержит несколько алгоритмических инструментальных

средств, в том числе алгоритмы обучения машины (ОМ) и искусственного интеллекта (ИИ), но эти примеры не являются ограничивающим признаком.

Система 2700 может, но не обязательно, включать в себя взаимодействия между множеством пользователей, как показано на чертеже. Такие взаимодействия обеспечивают большее удобство использования и большее удовольствие от использования мобильного информационного устройства для конечного пользователя.

На фиг.28 показана система интеллектуального агента из фиг.27 в более подробном виде. Как показано на чертеже, первый интеллектуальный агент 2800 предпочтительно, но не обязательно, является способным функционировать согласно данным 2802 сценария (например, описанной выше базы знаний) для того, чтобы он мог выполнять действия, обучаться и принимать решения относительно функционирования мобильного информационного устройства. Процесс обучения и развития первого интеллектуального агента 2800 предпочтительно обеспечивает модуль 2804 эволюции, который, как описано выше, обеспечивает эволюцию. Если первый интеллектуальный агент 2800 производит обмен информацией с пользователем посредством виртуального персонажа согласно предпочтительному варианту осуществления настоящего изобретения, то внешний вид виртуального персонажа предпочтительно обеспечивает анимационный модуль 2806.

Первый интеллектуальный агент 2800 также может, но не обязательно, производить обмен информацией через сеть (на чертеже не показана) с сервером 2808 базы данных и/или с другим сетевым ресурсом, например, с компьютером 2810, например, для получения информации для пользователя.

Первый интеллектуальный агент 2800 также может, но не обязательно, производить обмен информацией со вторым интеллектуальным агентом 2812, как показано на чертеже.

На фиг.29 показана блок-схема примера варианта реализации системы 2900 выбора действия согласно настоящему изобретению, которое обеспечивает инфраструктуру, обеспечивающую возможность выбора действия интеллектуальным агентом.

Система 2900 выбора действия предпочтительно содержит администратор 2902 действий (ActionManager) (см. также фиг.10 для ее описания), который фактически выполняет действие. Интерфейс 2904 для базовых действий (BaseAction) предпочтительно обеспечивает интерфейс для всех действий, выполняемых администратором 2902 действий (ActionManager).

Действия могут, но не обязательно, использовать способности устройств и приложений, обозначенных как администратор 2906 анимации (AnimationManager) и администратор 2908 звукового сопровождения (SoundManager), которые являются необходимыми для выполнения конкретного действия. Каждое действие предпочтительно, но не обязательно, объединяет соответствующие управляющие администраторы для надлежащего и правильного выполнения.

Администратор 2906 анимации (AnimationManager), предпочтительно, но не обязательно, также может осуществлять управление действием 2910 "изменение интерфейса пользователя" (ChangeUIAction), которое изменяет внешний вид визуального отображения интерфейса пользователя. В дополнение к этому или в альтернативном варианте, в том случае, если для отображения интеллектуального агента для пользователя используют виртуальный персонаж, администратор 2906 анимации (AnimationManager) также может предпочтительно, но не обязательно, осуществлять управление действием 2912 "отойти от объекта"

(GoAwayFromObjectAction) и действием 2914 "пойти к объекту" (GoTowardObjectAction), которые дают возможность виртуальному персонажу взаимодействовать с виртуальными объектами в виртуальном мире виртуального персонажа.

На фиг.30А и фиг.30В показаны два иллюстративных, не ограничивающих примера снимков экрана с виртуальным персонажем согласно настоящему изобретению, изображение которого выведено на экран мобильного информационного устройства. На фиг.30А показан приведенный в качестве примера снимок экрана интерфейса пользователя для регулировки громкости тональной мелодии звонка путем взаимодействия с виртуальным персонажем. На фиг.30В показан приведенный в качестве примера снимок экрана интерфейса пользователя для приема сообщений путем взаимодействия с виртуальным персонажем.

Несмотря на то, что было описано лишь ограниченное количество вариантов осуществления настоящего изобретения, понятно, что может быть реализовано множество изменений, модификаций и иных вариантов применения настоящего изобретения.

Формула изобретения

1. Интерфейс пользователя с упреждающим действием для вычислительного устройства, причем вычислительное устройство имеет операционную систему, содержащий:

(а) блок интерфейса, предназначенный для обмена информацией между пользователем и упомянутой операционной системой; и

(б) обучающийся модуль, включающий в себя датчик для выявления шаблона взаимодействия пользователя с блоком интерфейса и для упреждающего изменения, по меньшей мере, одной функции блока интерфейса согласно упомянутому выявленному шаблону.

2. Интерфейс пользователя с упреждающим действием по п.1, в котором упомянутый, по меньшей мере, один шаблон выбран из группы, состоящей из шаблона, определенного согласно, по меньшей мере, одному предыдущему взаимодействию пользователя с блоком интерфейса, и заранее заданного шаблона, или из их комбинации.

3. Интерфейс пользователя с упреждающим действием по п.1 или 2, в котором блок интерфейса содержит графический дисплей, а упомянутое изменение, по меньшей мере, одной функции блока интерфейса содержит изменение, по меньшей мере, участка упомянутого графического дисплея.

4. Интерфейс пользователя с упреждающим действием по п.3, в котором упомянутое изменение, по меньшей мере, участка графического дисплея содержит следующие операции: выбирают меню для отображения на экране дисплея согласно упомянутому выявленному шаблону и отображают упомянутое меню на экране дисплея.

5. Интерфейс пользователя с упреждающим действием по п.4, в котором упомянутая операция выбора меню содержит следующую операцию: осуществляют построение меню из множества пунктов меню.

6. Интерфейс пользователя с упреждающим действием по п.1 или 2, в котором блок интерфейса содержит средство воспроизведения звука, а упомянутое изменение, по меньшей мере, одной функции блока интерфейса содержит изменение, по меньшей мере, одного слышимого звука, воспроизводимого вычислительным устройством.

7. Интерфейс пользователя с упреждающим действием по п.1, в котором

вычислительное устройство выбрано из группы, состоящей из обычного компьютера, банкомата (АТМ), мобильного информационного устройства, которым является в том числе телефонный аппарат сотовой связи, персонального информационного устройства (PDA) или бытового прибора, снабженного операционной системой.

8. Интерфейс пользователя с упреждающим действием по п.1 или 2, в котором обучающийся модуль содержит базу знаний для хранения информации, собранной в результате взаимодействия с пользователем и/или операционной системой.

9. Интерфейс пользователя с упреждающим действием по п.8, в котором база знаний содержит множество объединенных знаний, определенных из поведения пользователя и из заранее запрограммированной информации.

10. Интерфейс пользователя с упреждающим действием по п.8, в котором обучающийся модуль дополнительно содержит множество датчиков, предназначенных для того, чтобы воспринимать состояние операционной системы.

11. Интерфейс пользователя с упреждающим действием по п.10, в котором обучающийся модуль дополнительно содержит блок восприятия, предназначенный для обработки данных, полученных с выхода упомянутых датчиков, для определения состояния операционной системы и состояния блока интерфейса.

12. Интерфейс пользователя с упреждающим действием по п.11, в котором обучающийся модуль дополнительно содержит систему логических рассуждений для обновления базы знаний и для обучения связи между изменением блока интерфейса и состоянием операционной системы.

13. Интерфейс пользователя с упреждающим действием по п.1 или 2, в котором обучающийся модуль дополнительно содержит, по меньшей мере, один из следующих алгоритмов: алгоритм искусственного интеллекта и алгоритм обучения машины.

14. Интерфейс пользователя с упреждающим действием по п.1 или 2, в котором обучающийся модуль максимально увеличивает процент упреждающих изменений, а это приводит к тому, что в результате упомянутого изменения пользователь осуществляет прямой выбор.

15. Интерфейс пользователя с упреждающим действием по п.14, в котором максимальное увеличение реализовано путем обучения с пополнением.

16. Интерфейс пользователя с упреждающим действием по п.15, в котором обучение с пополнением реализовано путем итерационного процесса обучения.

17. Интерфейс пользователя с упреждающим действием по п.16, в котором каждую итерацию процесса обучения выполняют после того, как было выполнено упомянутое изменение.

18. Интерфейс пользователя с упреждающим действием по п.3, в котором упомянутое упреждающее изменение, по меньшей мере, одной функции блока интерфейса содержит активацию дополнительного прикладного программного обеспечения посредством операционной системы.

19. Интерфейс пользователя с упреждающим действием по п.1, дополнительно содержащий интеллектуального агента, способного обмениваться информацией с человеком-пользователем.

20. Интерфейс пользователя с упреждающим действием по п.19, в котором интеллектуальный агент управляет, по меньшей мере, одним взаимодействием вычислительного устройства через сеть.

21. Способ упреждающего взаимодействия между пользователем и вычислительным устройством посредством интерфейса пользователя, при этом вычислительное устройство имеет операционную систему, а способ содержит

следующие операции:

с использованием датчика выявляют шаблон поведения пользователя согласно, по меньшей мере, одному взаимодействию пользователя с интерфейсом пользователя и осуществляют упреждающее изменение, по меньшей мере, одной функции интерфейса пользователя согласно упомянутому шаблону.

22. Способ по п.21, в котором упомянутый, по меньшей мере, один шаблон выбран от группы, состоящей из шаблона, определенного согласно, по меньшей мере, одному предыдущему взаимодействию пользователя с интерфейсом пользователя, и заранее заданного шаблона или из их комбинации.

23. Способ по п.21 или 22, в котором интерфейс пользователя содержит графический дисплей, а упомянутое изменение, по меньшей мере, одной функции интерфейса пользователя содержит изменение, по меньшей мере, участка графического дисплея.

24. Способ по п.23, в котором упомянутое изменение, по меньшей мере, участка графического дисплея содержит следующие операции:

выбирают меню для отображения на экране дисплея согласно упомянутому выявленному шаблону и

отображают упомянутое меню на экране дисплея.

25. Способ по п.24, в котором выбор меню содержит следующую операцию: построение меню из множества пунктов меню.

26. Способ по п.21 или 22, в котором интерфейс пользователя содержит средство воспроизведение звука, а упомянутое изменение, по меньшей мере, одной функции интерфейса пользователя содержит следующую операцию:

изменяют, по меньшей мере, один слышимый звук, воспроизводимый вычислительным устройством.

27. Способ по п.21, в котором вычислительное устройство выбрано из группы, состоящей из обычного компьютера, АТМ (автоматическая кассовая машина), сотового телефона, мобильного устройства обработки информации, PDA (персональный цифровой помощник), бытового устройства, имеющего операционную систему.

28. Способ по п.21 или 22, в котором обучающийся модуль содержит базу знаний, а способ содержит следующую дополнительную операцию: сохраняют информацию, собранную в результате взаимодействий с пользователем и/или с операционной системой, с использованием базы знаний.

29. Способ по п.28, в котором база знаний содержит множество объединенных знаний, определенных из поведения пользователя и из заранее запрограммированной информации.

30. Способ по п.28, в котором обучающийся модуль дополнительно содержит множество датчиков и использует упомянутые датчики для того, чтобы воспринимать состояние операционной системы.

31. Способ по п.30, в котором обучающийся модуль дополнительно содержит блок восприятия и использует блок восприятия для обработки данных с выхода датчиков и для определения состояния операционной системы и состояния интерфейса пользователя.

32. Способ по п.31, в котором обучающийся модуль дополнительно содержит систему логических рассуждений и использует систему логических рассуждений для обновления базы знаний и для обучения связи между изменением интерфейса пользователя и состоянием операционной системы.

33. Способ по п.21 или 22, в котором обучающийся модуль дополнительно содержит, по меньшей мере, один из следующих алгоритмов: алгоритм искусственного интеллекта и алгоритм обучения машины, и этот способ выполняется обучающимся модулем.

5 34. Способ по п.21 или 22, в котором обучающийся модуль максимально увеличивает процент упреждающих изменений, а это приводит к тому, что в результате упомянутого изменения пользователь осуществляет прямой выбор.

35. Способ по п.34, в котором упомянутое максимальное увеличение реализовано
10 путем обучения с пополнением.

36. Способ по п.35, в котором обучение с пополнением реализовано путем итерационного процесса обучения.

37. Способ по п.36, в котором каждую итерацию процесса обучения выполняют после того, как было выполнено упомянутое изменение.

15 38. Способ по п.23, в котором упомянутое упреждающее изменение, по меньшей мере, одной функции интерфейса пользователя содержит активацию дополнительного прикладного программного обеспечения посредством операционной системы.

39. Способ по п.38, в котором способ реализован с использованием
20 интеллектуального агента, способного обмениваться информацией с человеком-пользователем.

40. Способ по п.39, в котором интеллектуальный агент управляет, по меньшей мере, одним взаимодействием вычислительного устройства через сеть.

25

30

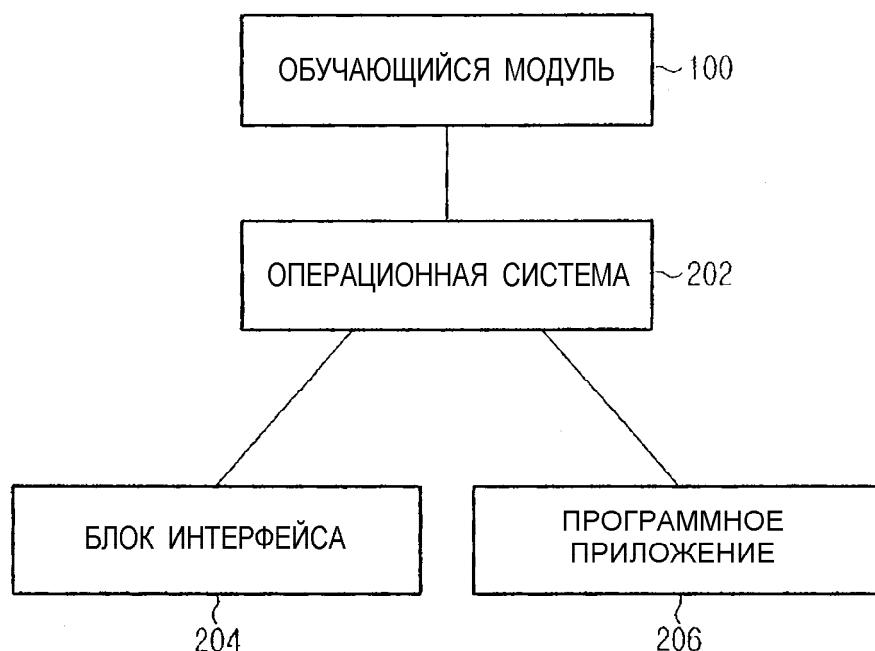
35

40

45

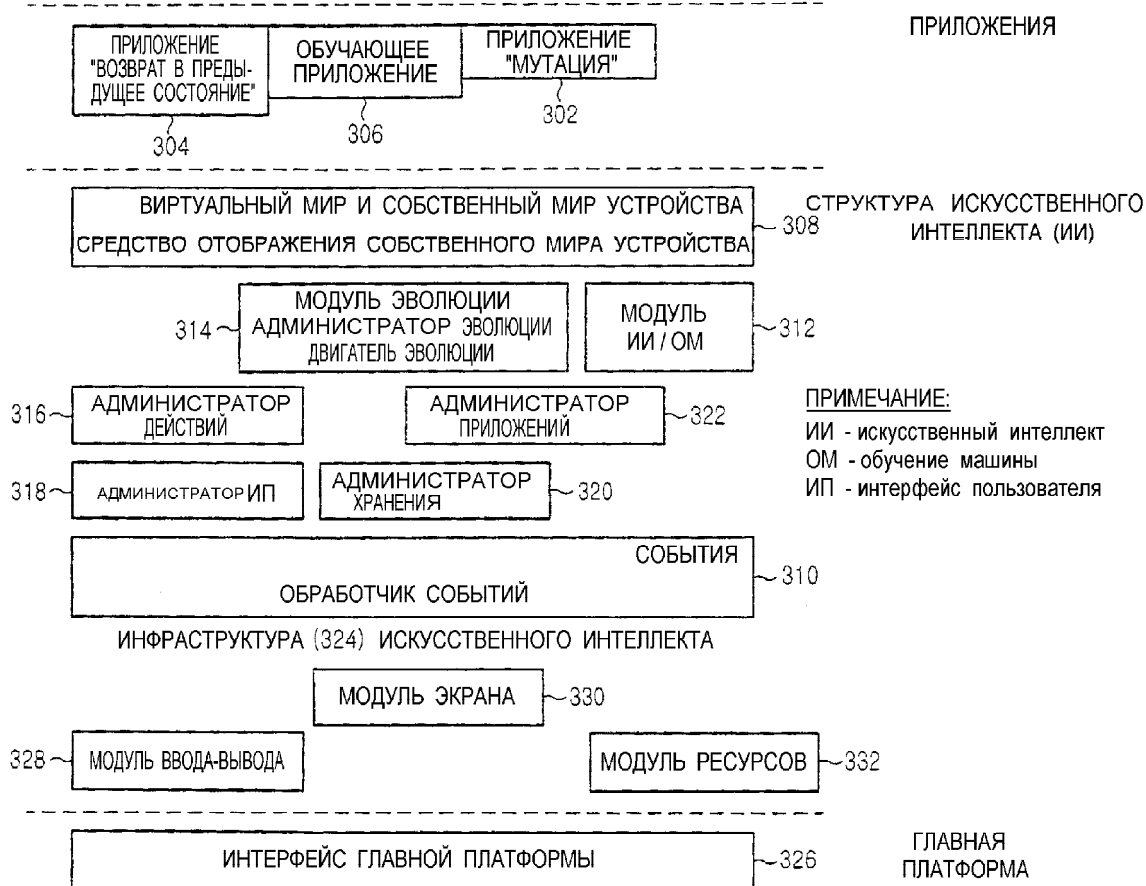
50

200

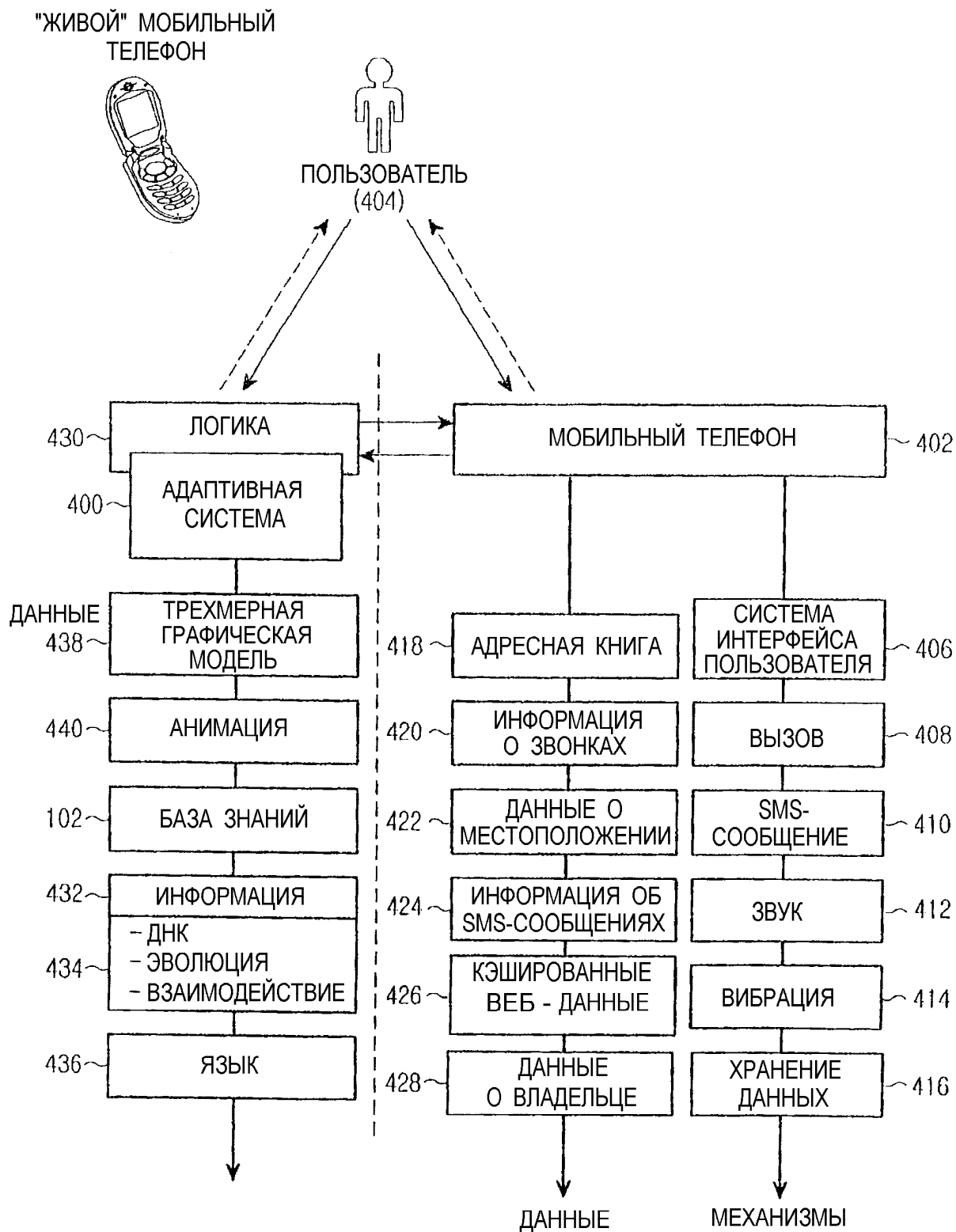


Фиг. 2

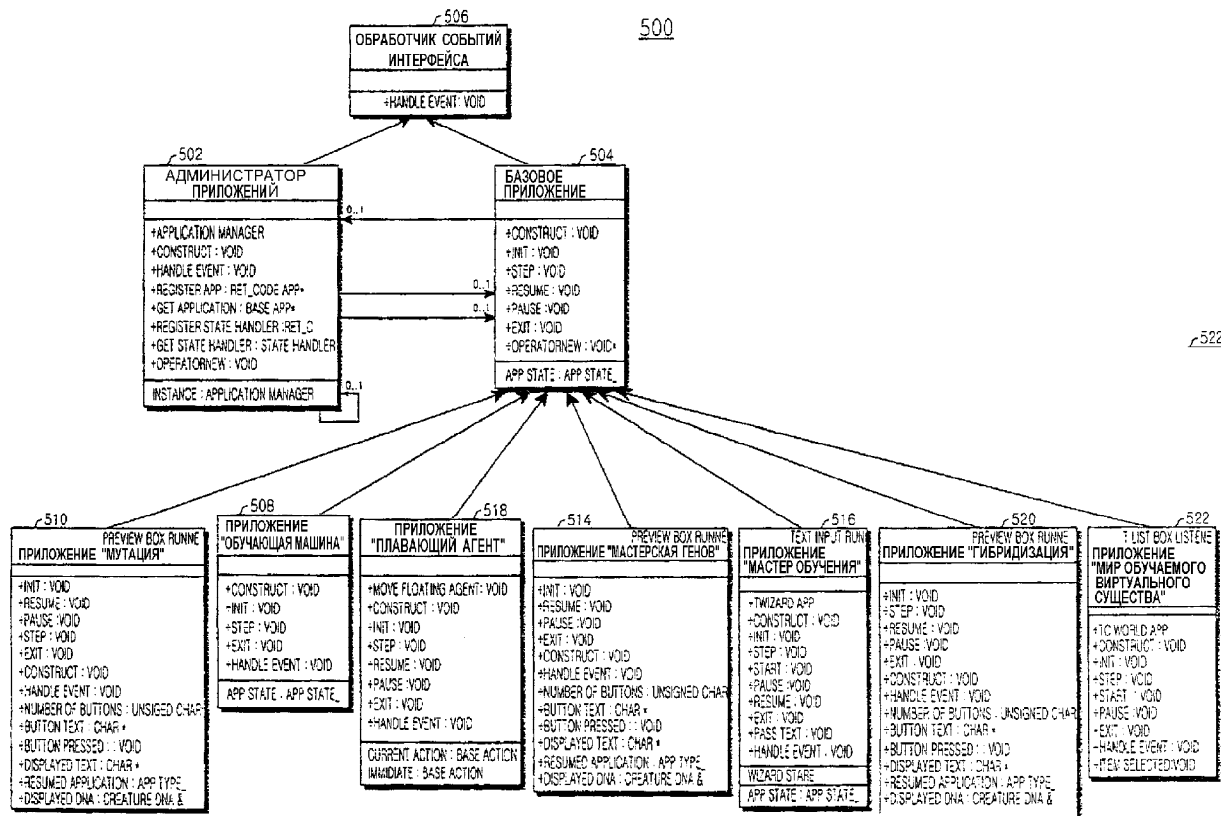
300



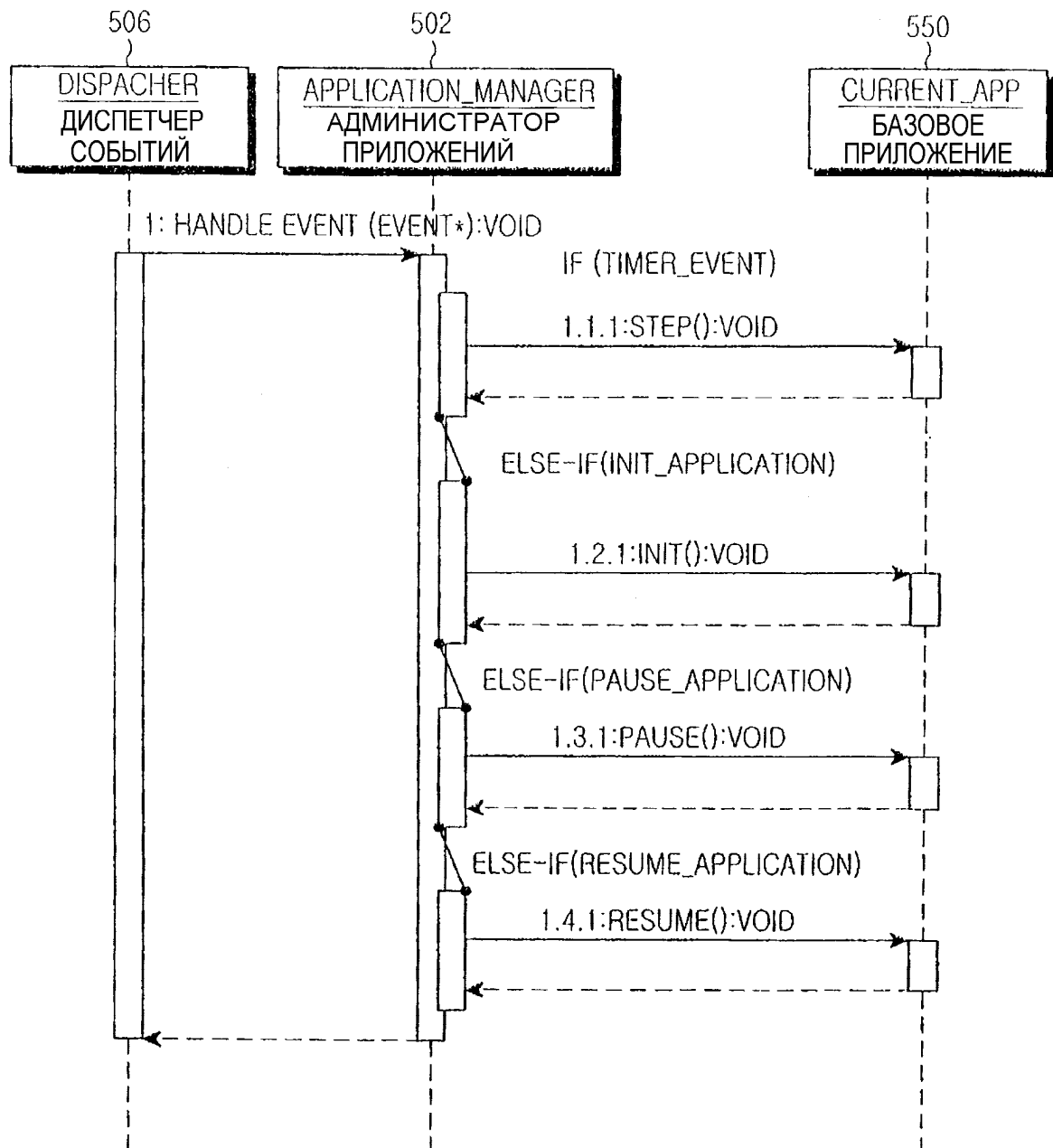
Фиг. 3



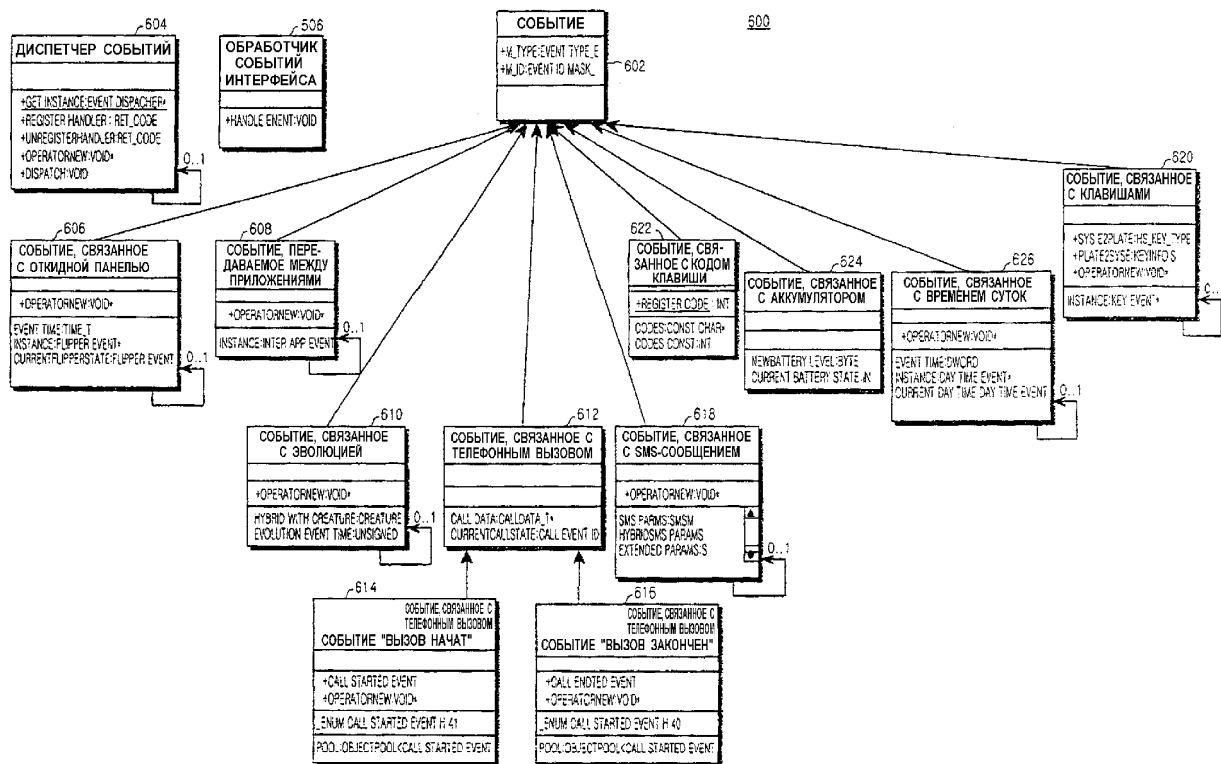
Фиг. 4



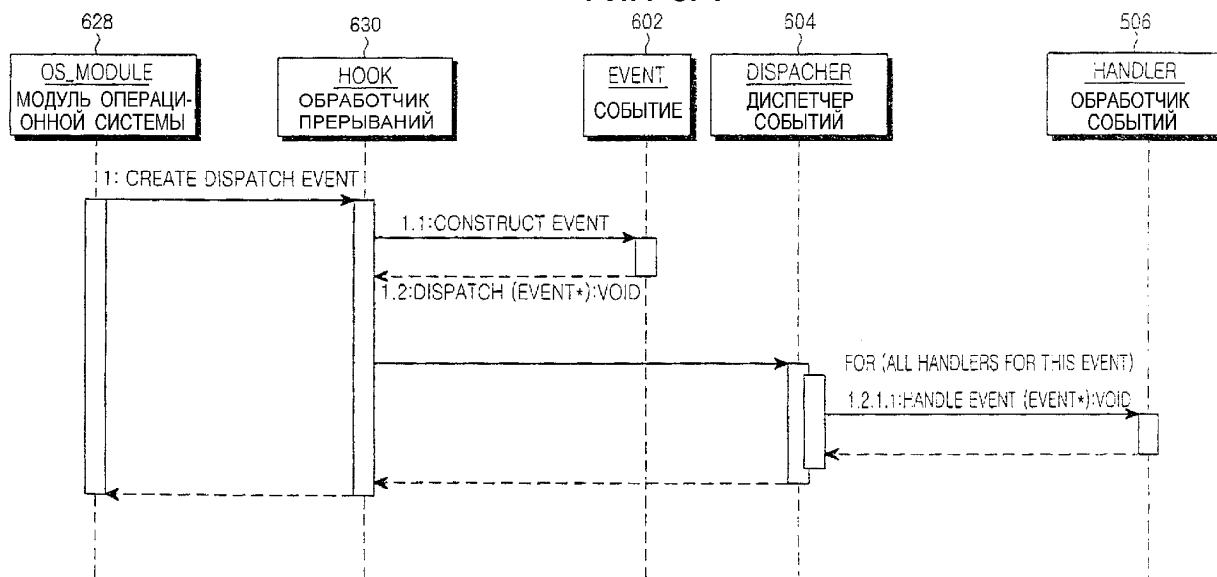
Фиг. 5А



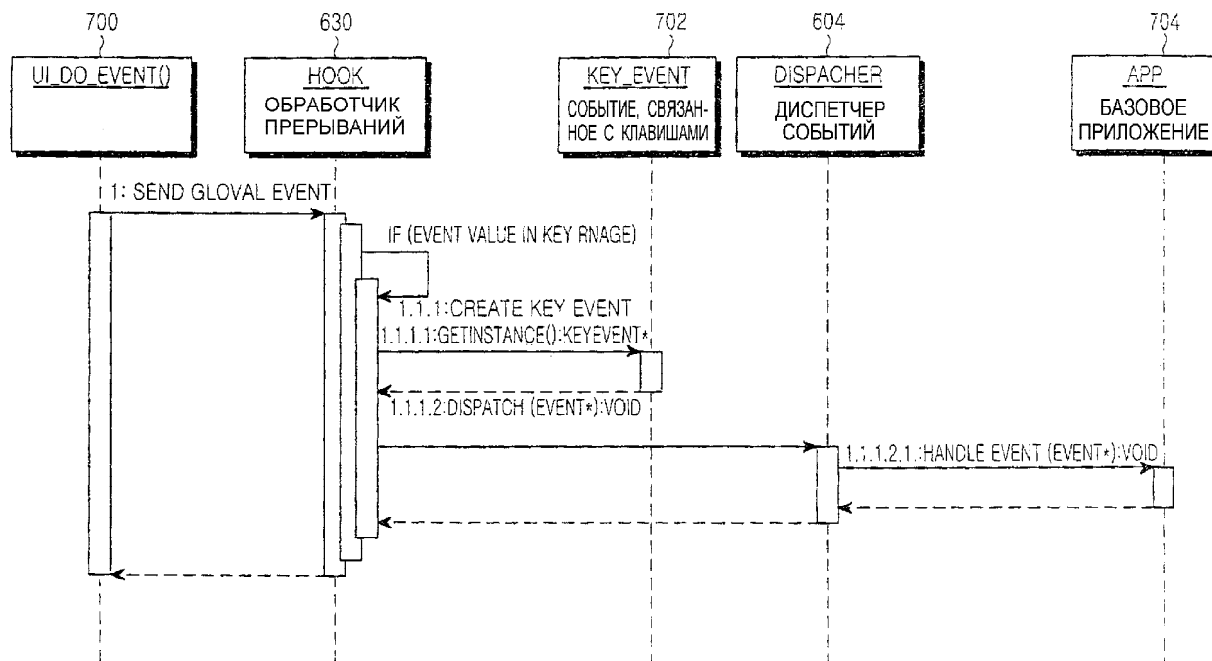
Фиг. 5В



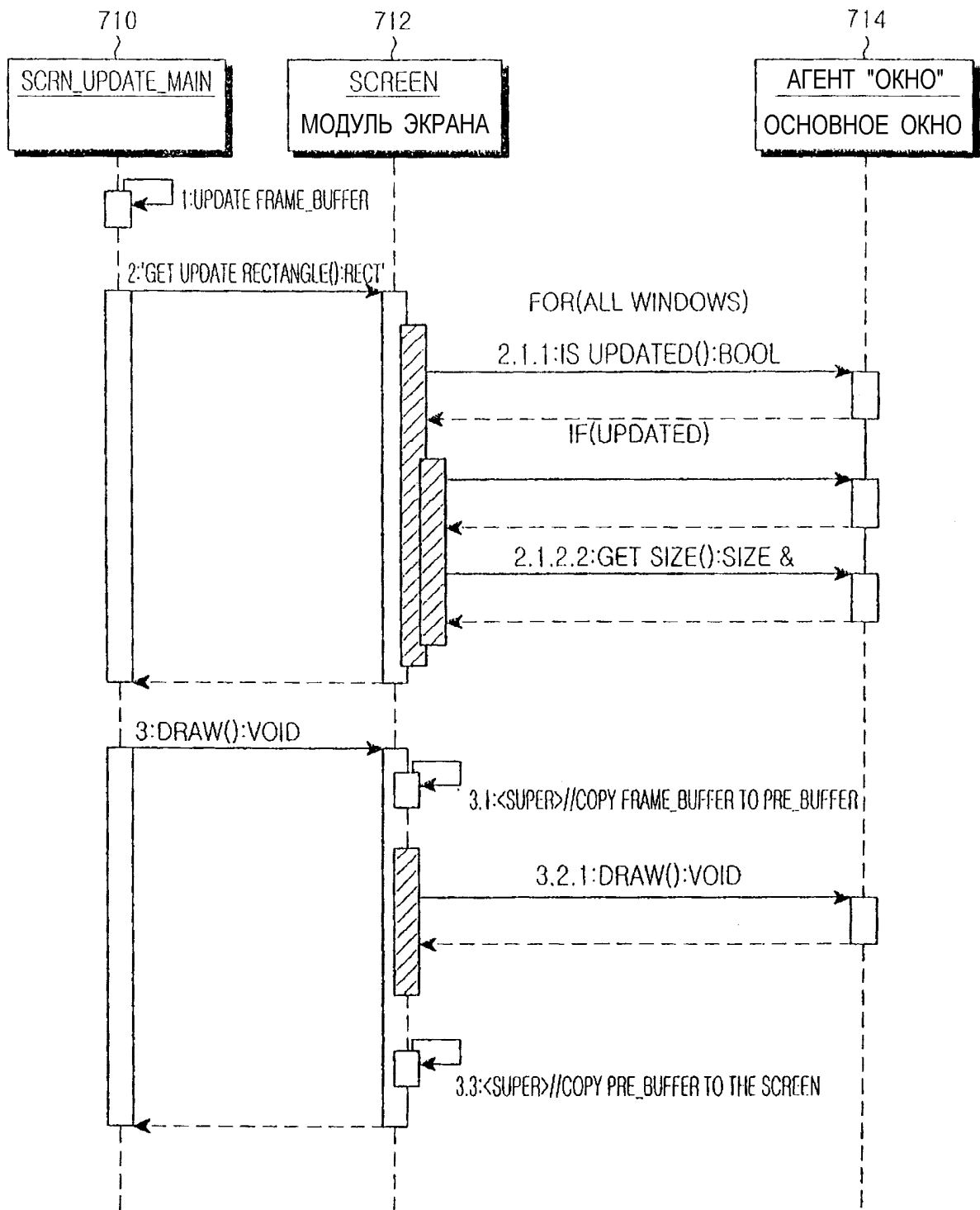
Фиг. 6А



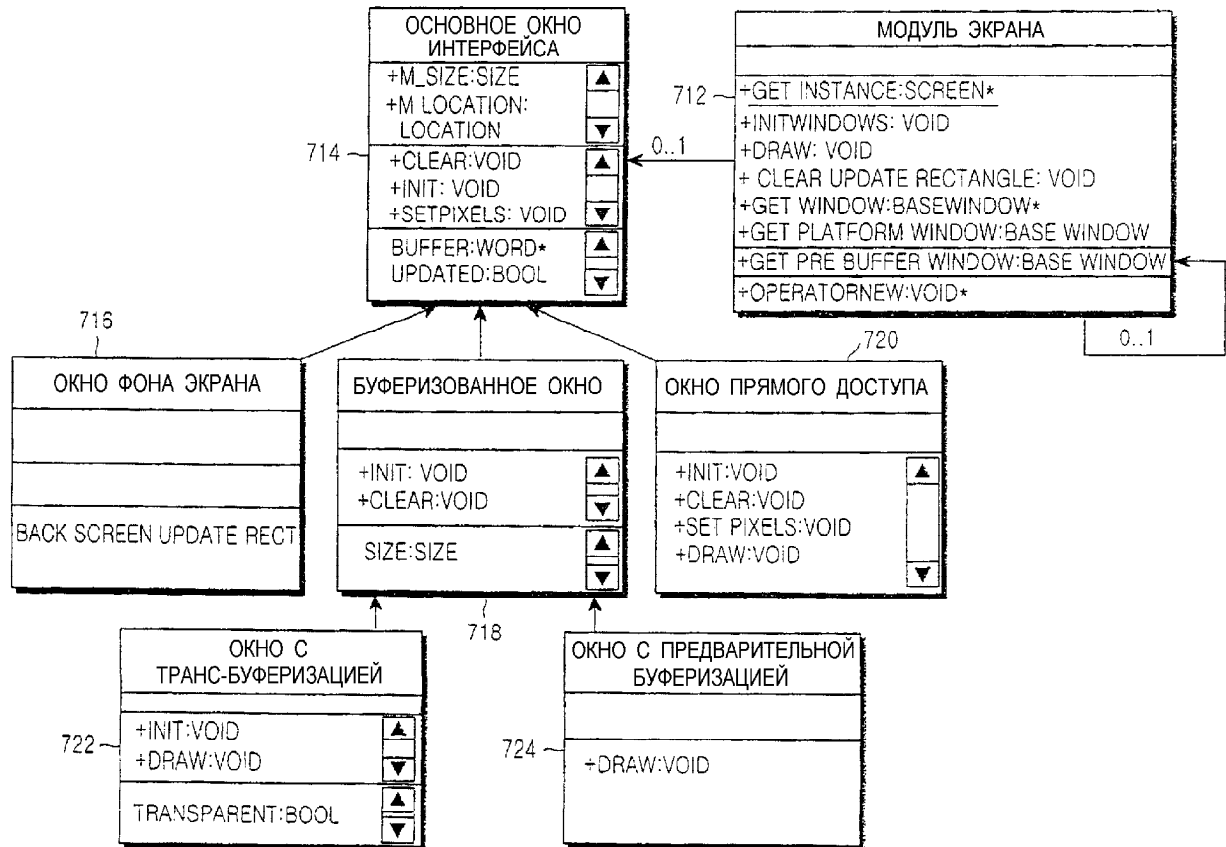
Фиг. 6В



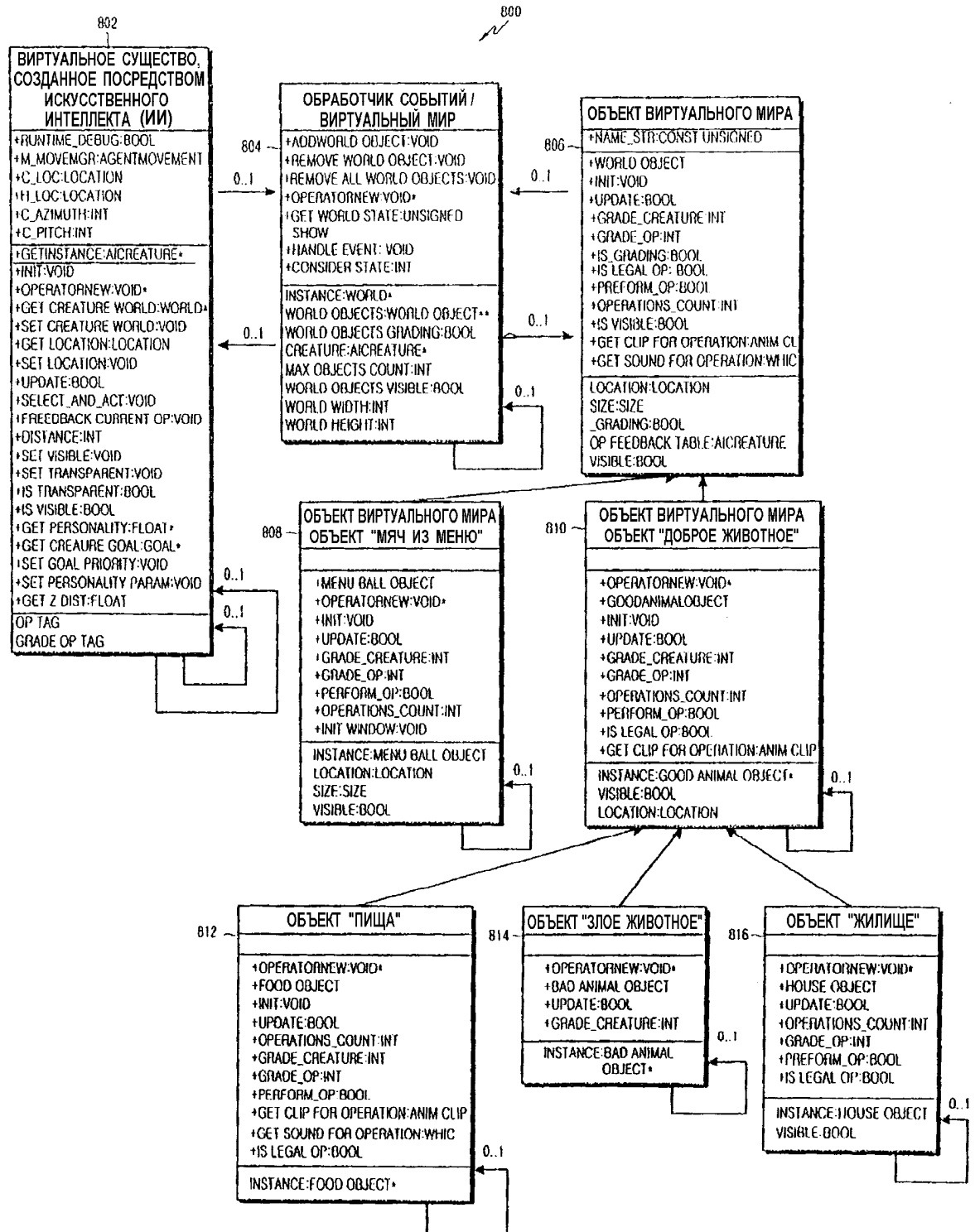
Фиг. 7А



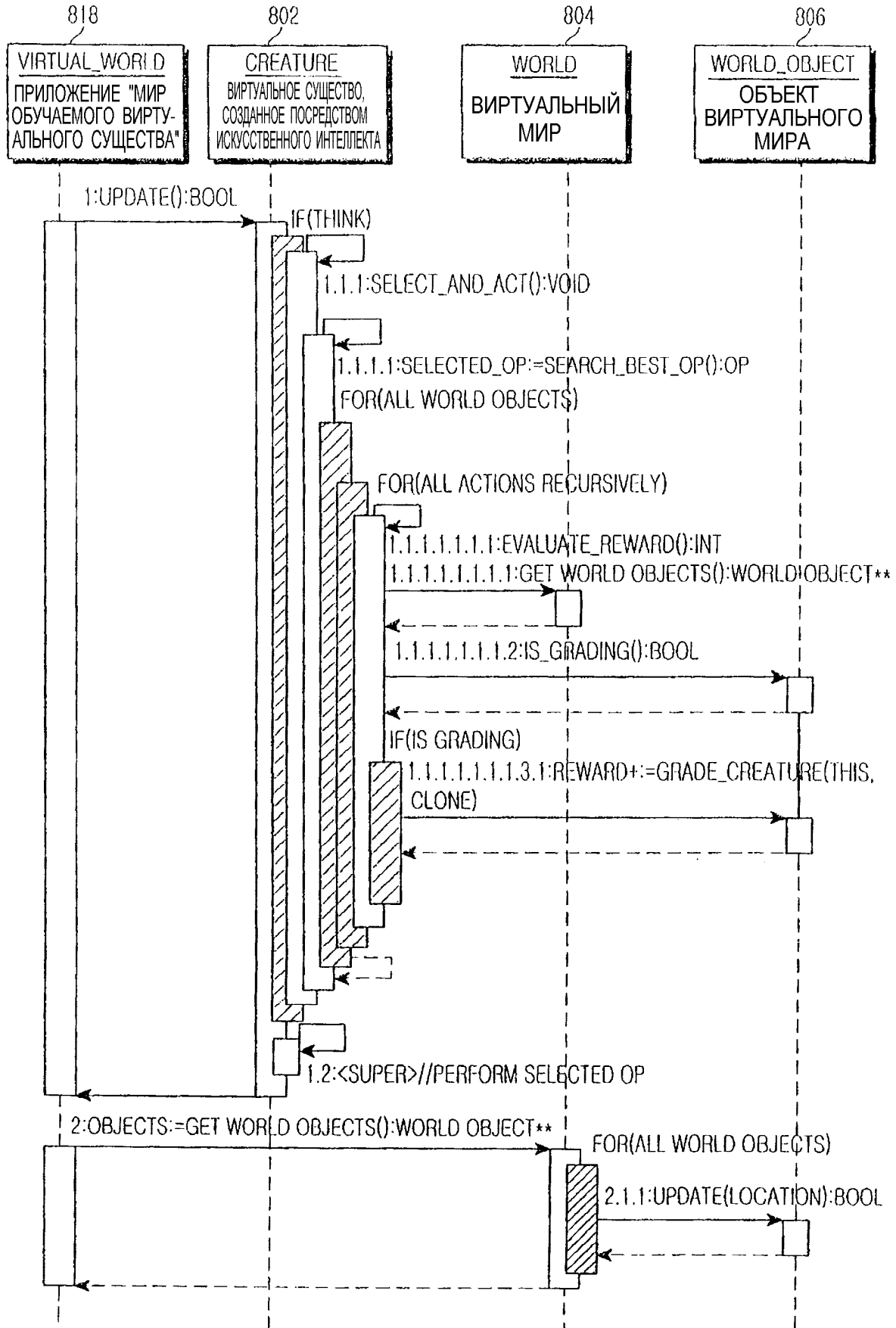
Фиг. 7 В



Фиг. 7С

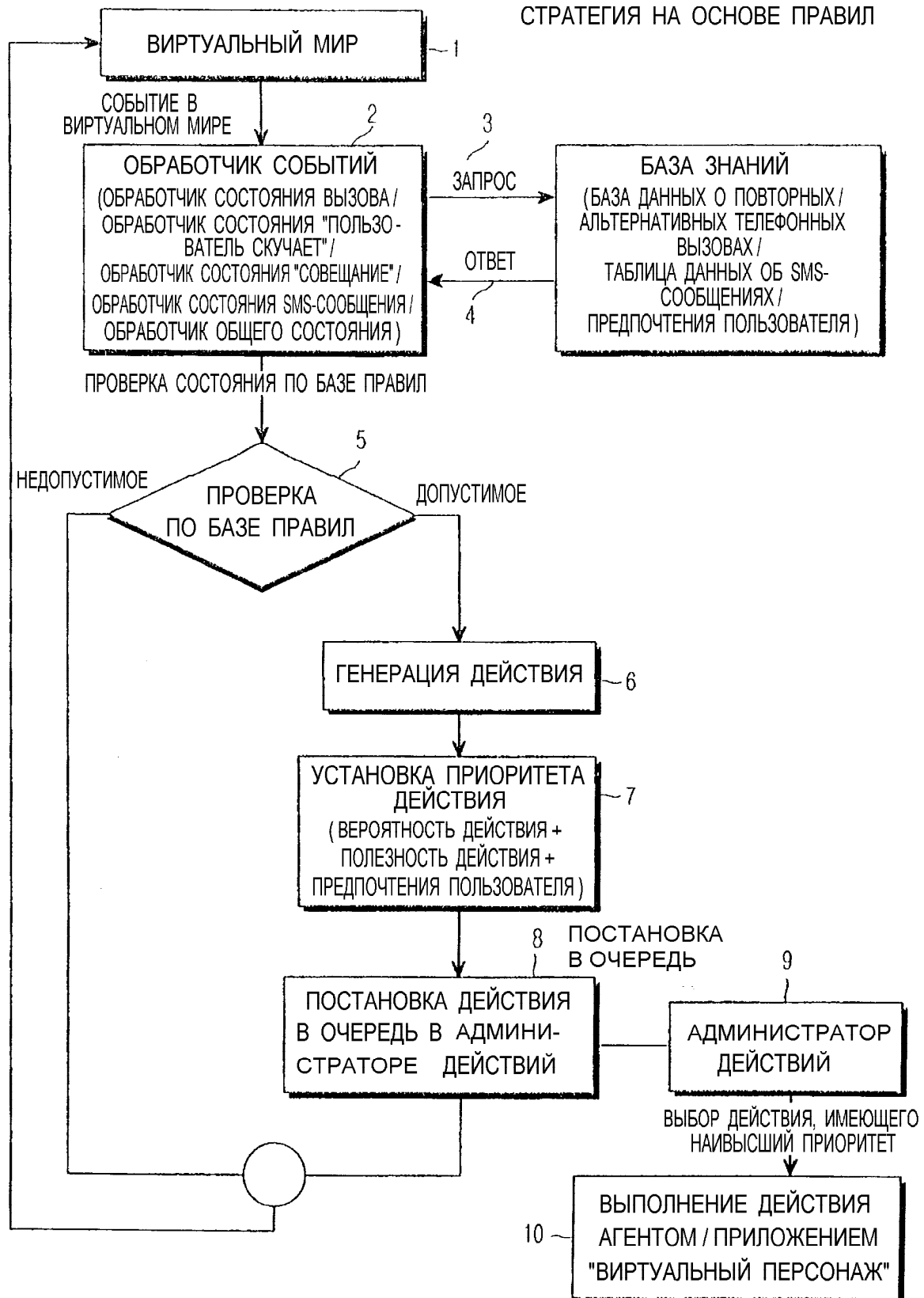


Фиг. 8А

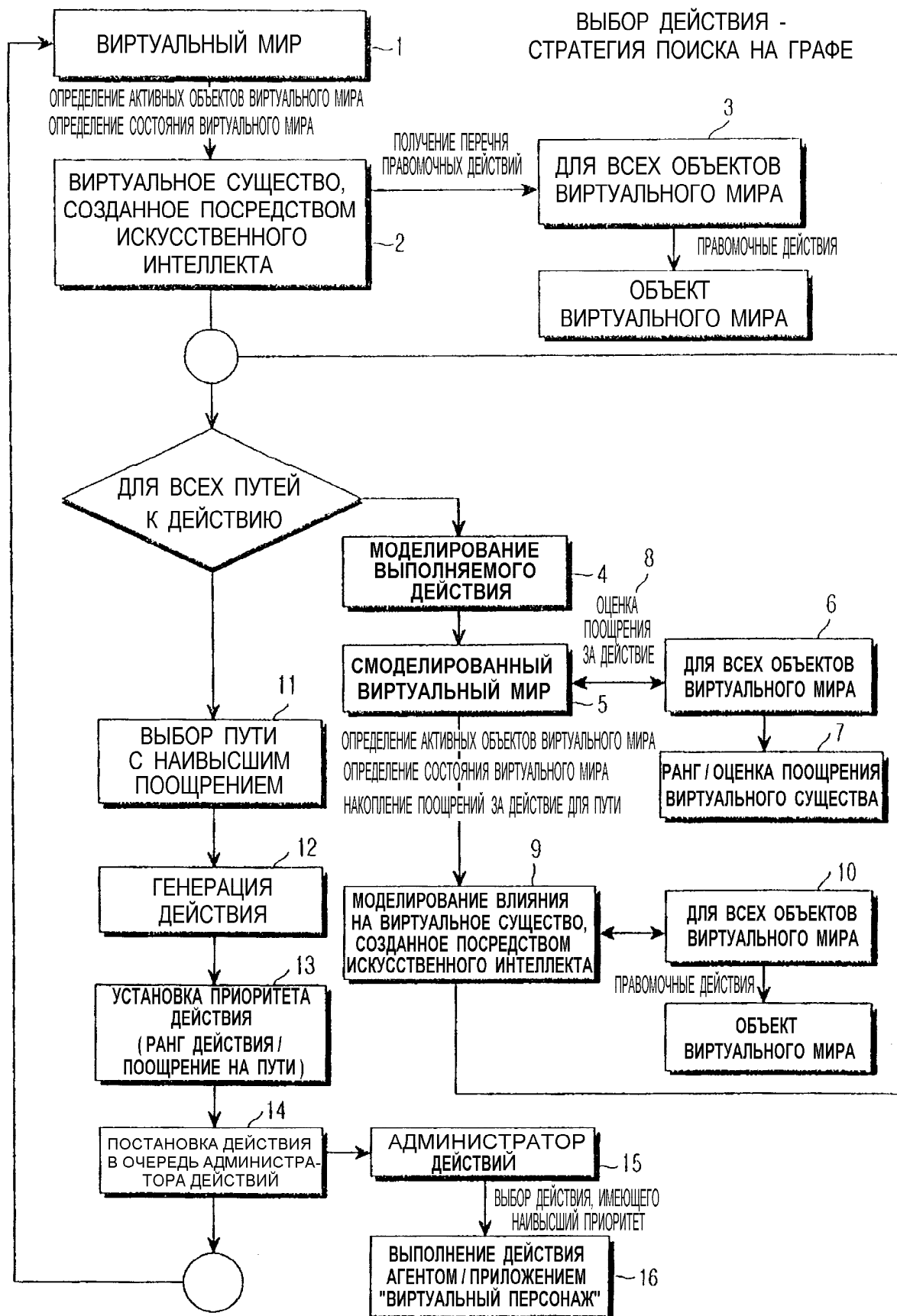


Фиг. 8В

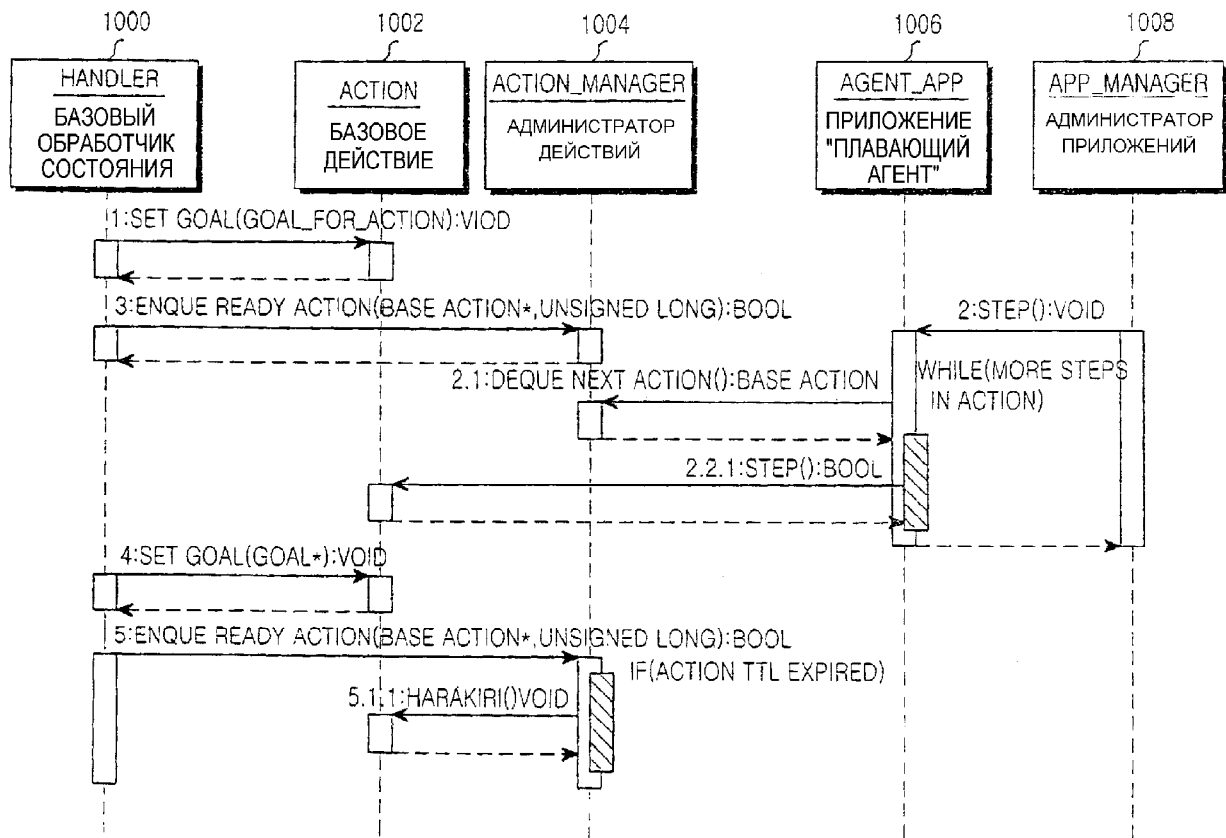
ВЫБОР ДЕЙСТВИЯ - СТРАТЕГИЯ НА ОСНОВЕ ПРАВИЛ



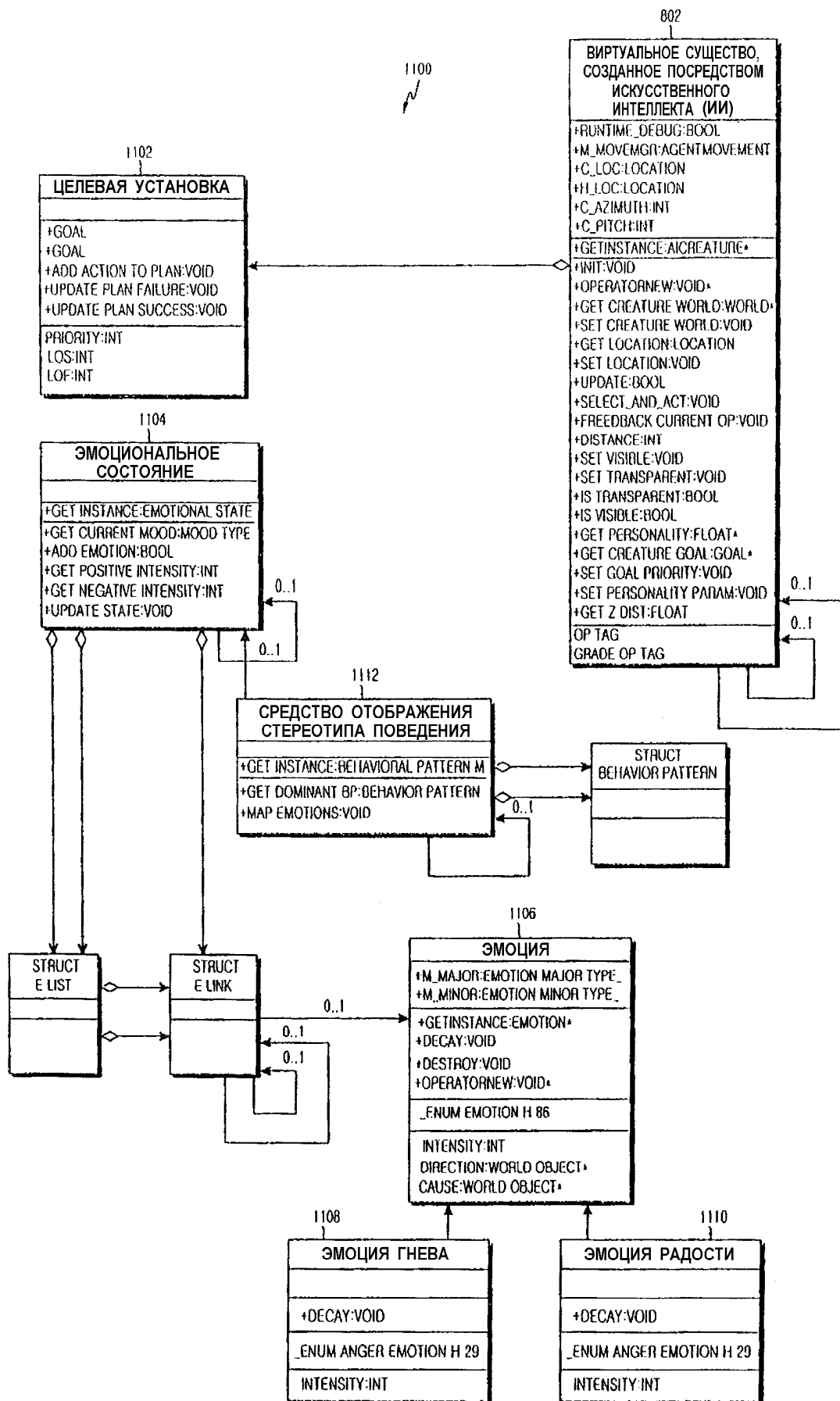
Фиг. 9А



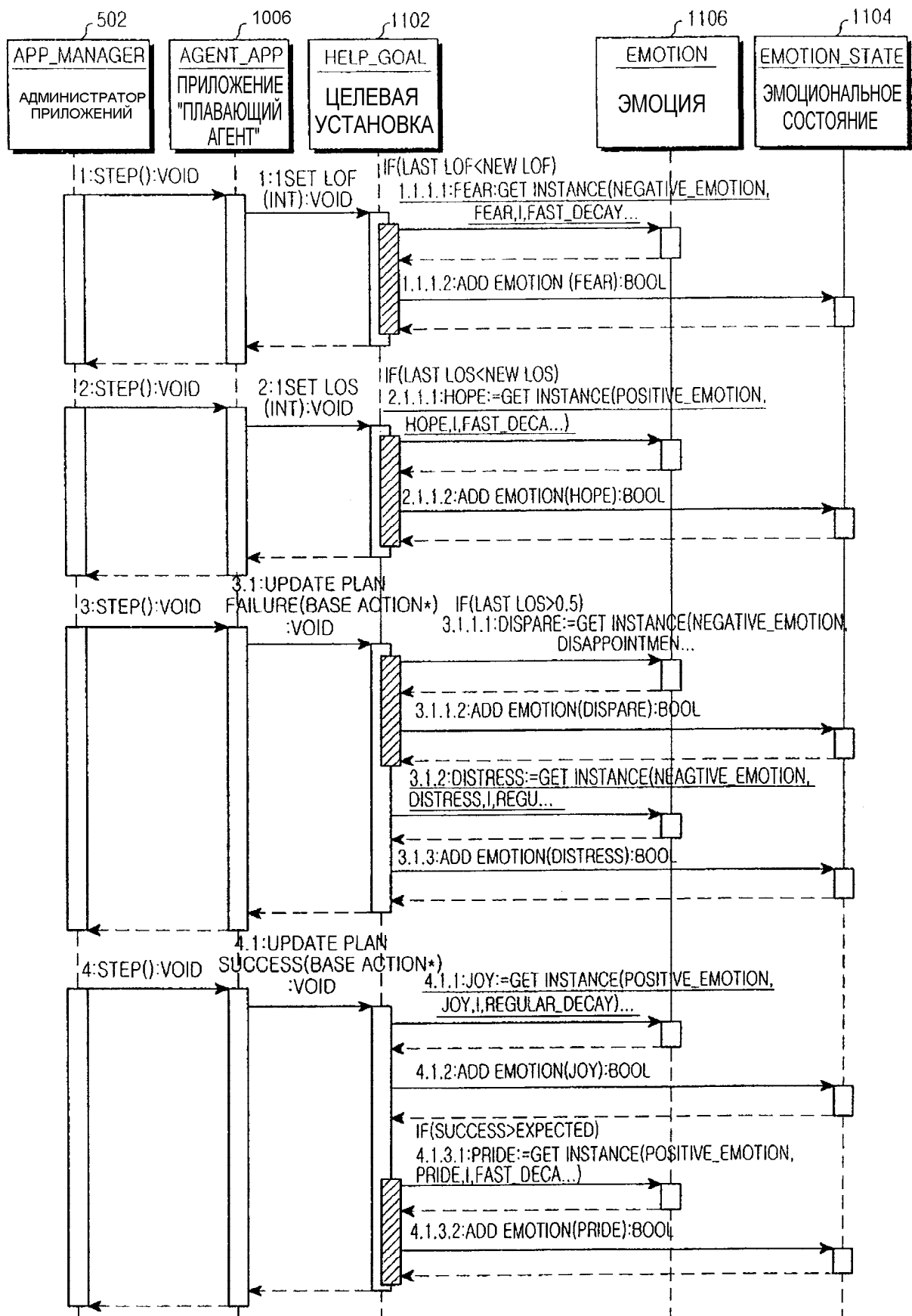
Фиг. 9В



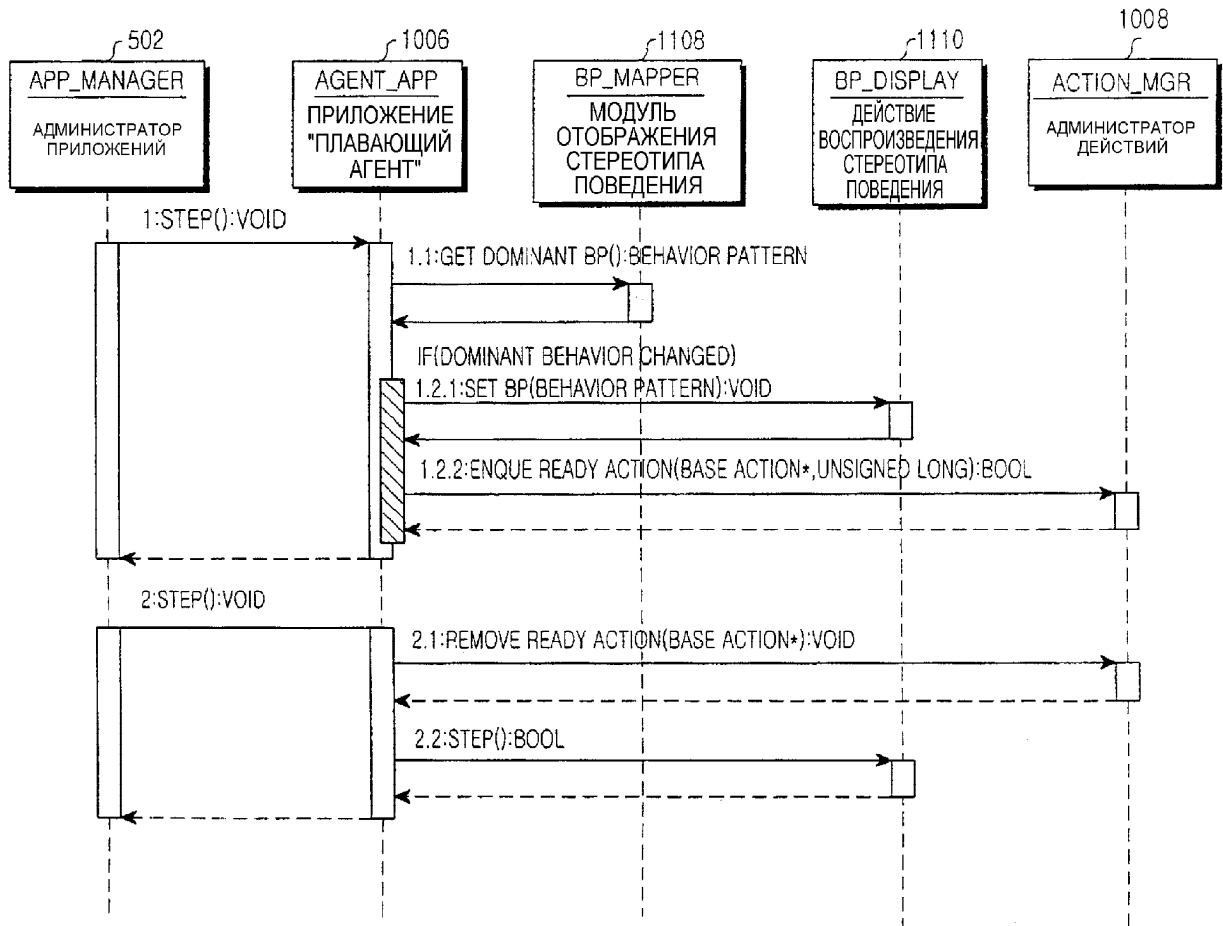
Фиг. 10



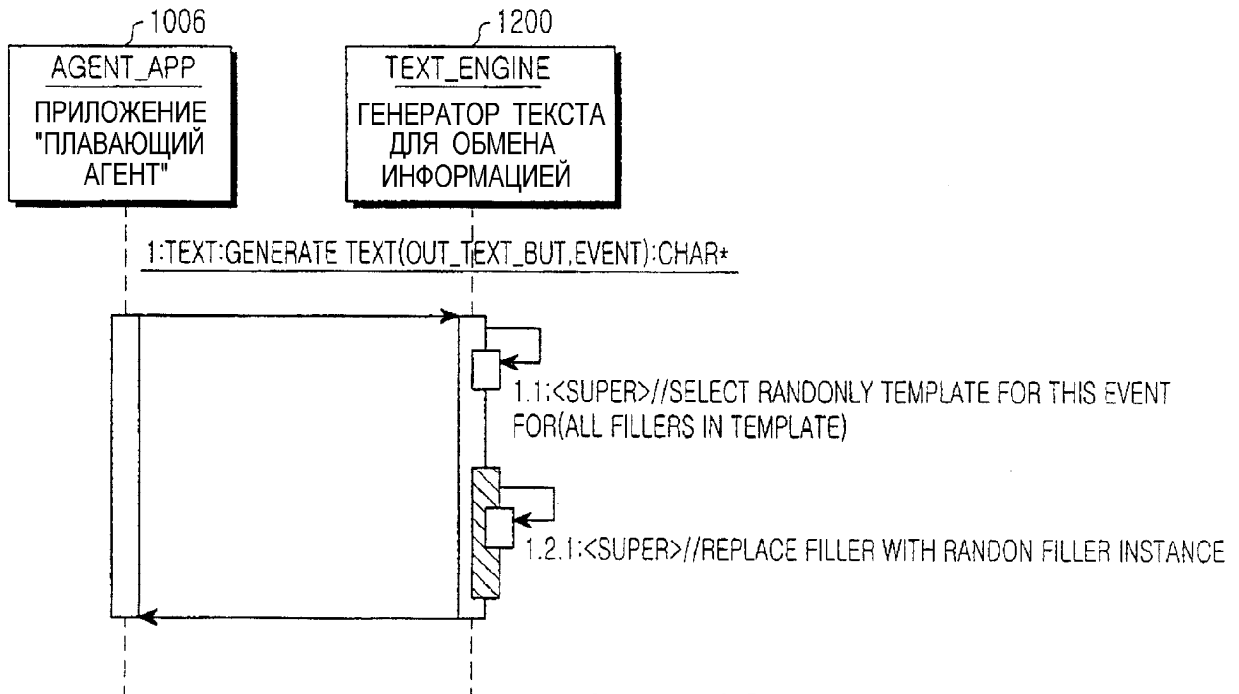
Фиг. 11А



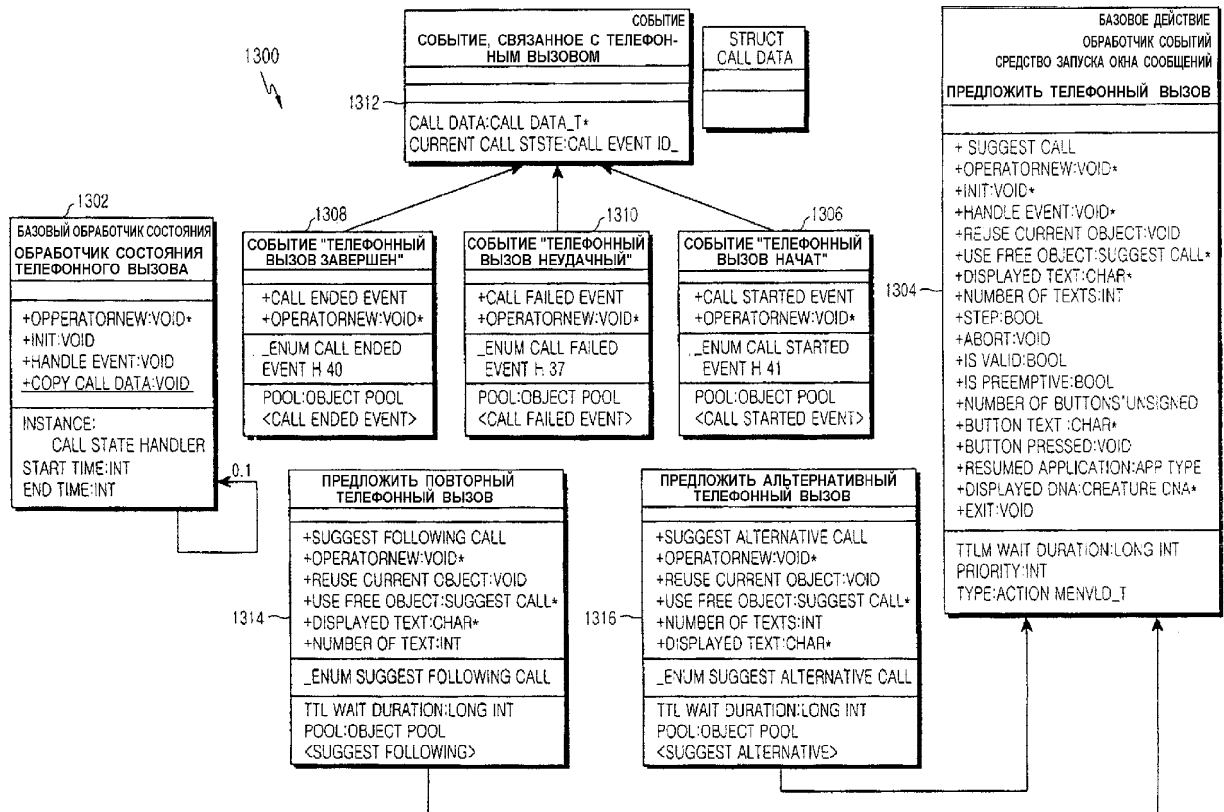
Фиг. 11В



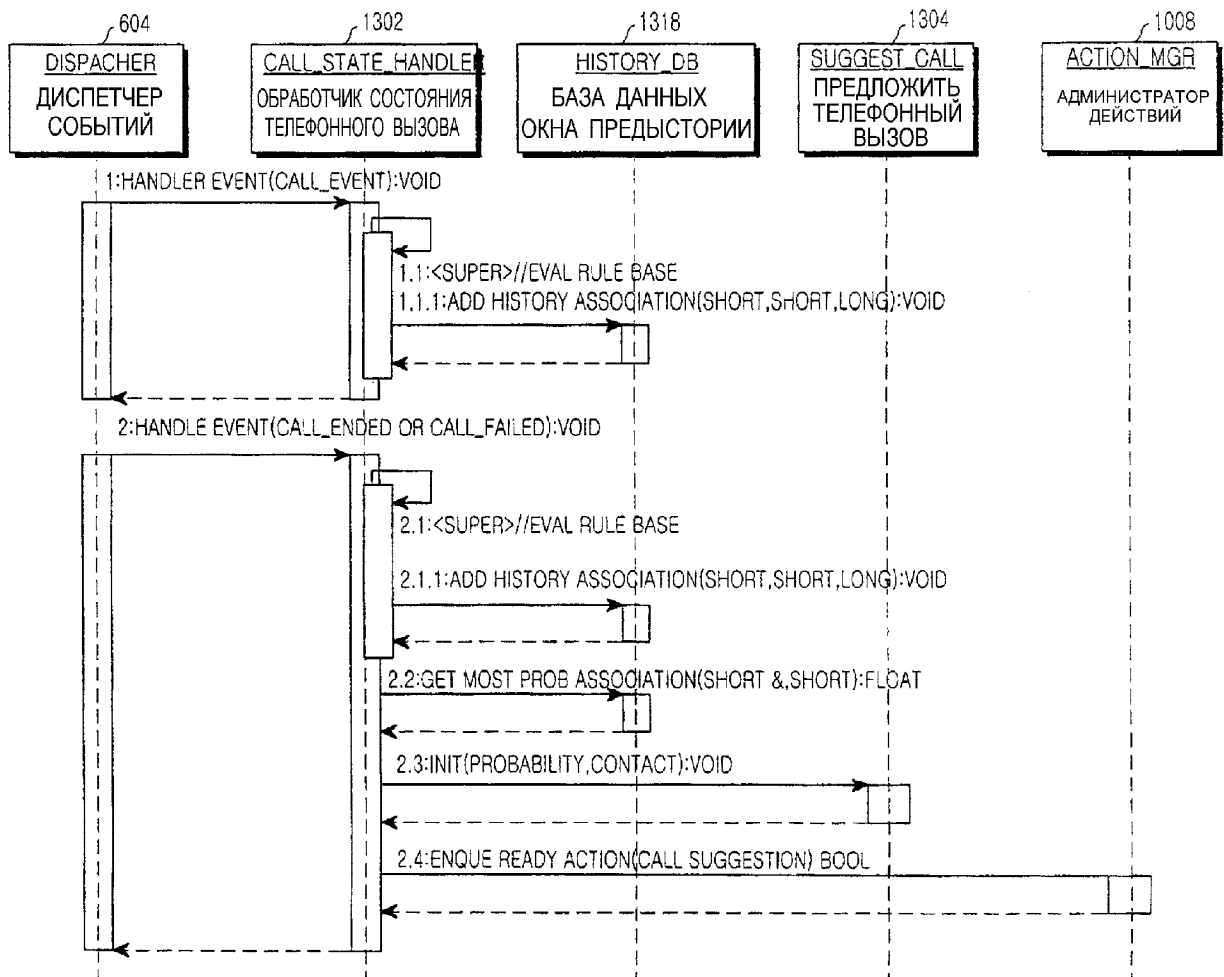
Фиг. 11С



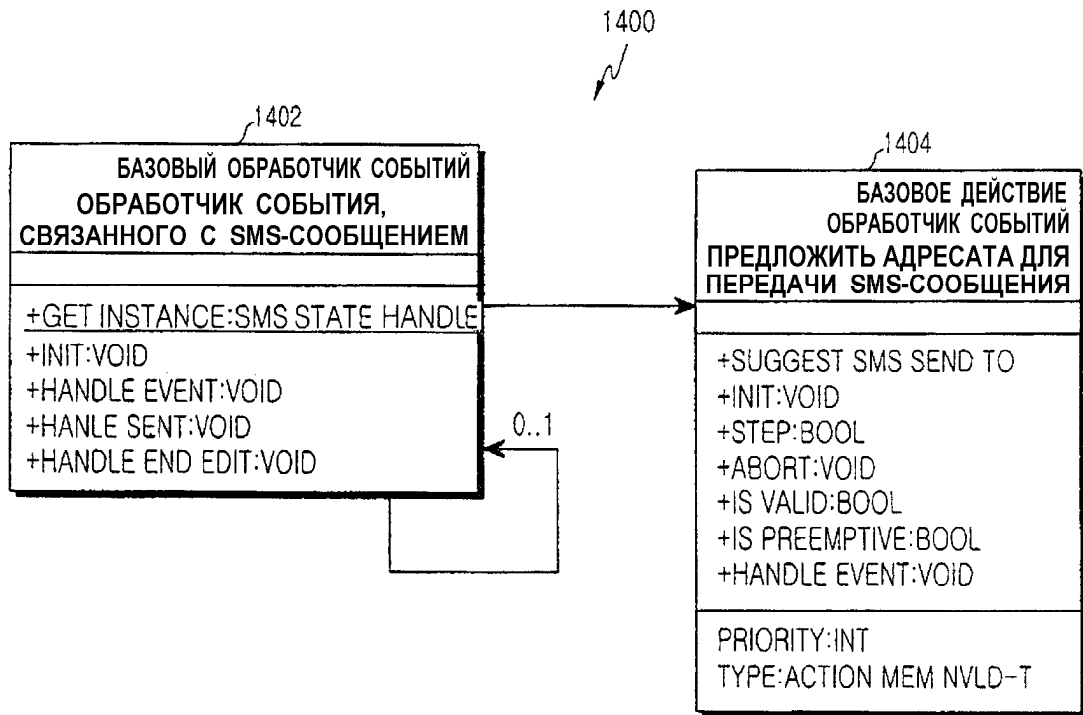
Фиг. 12



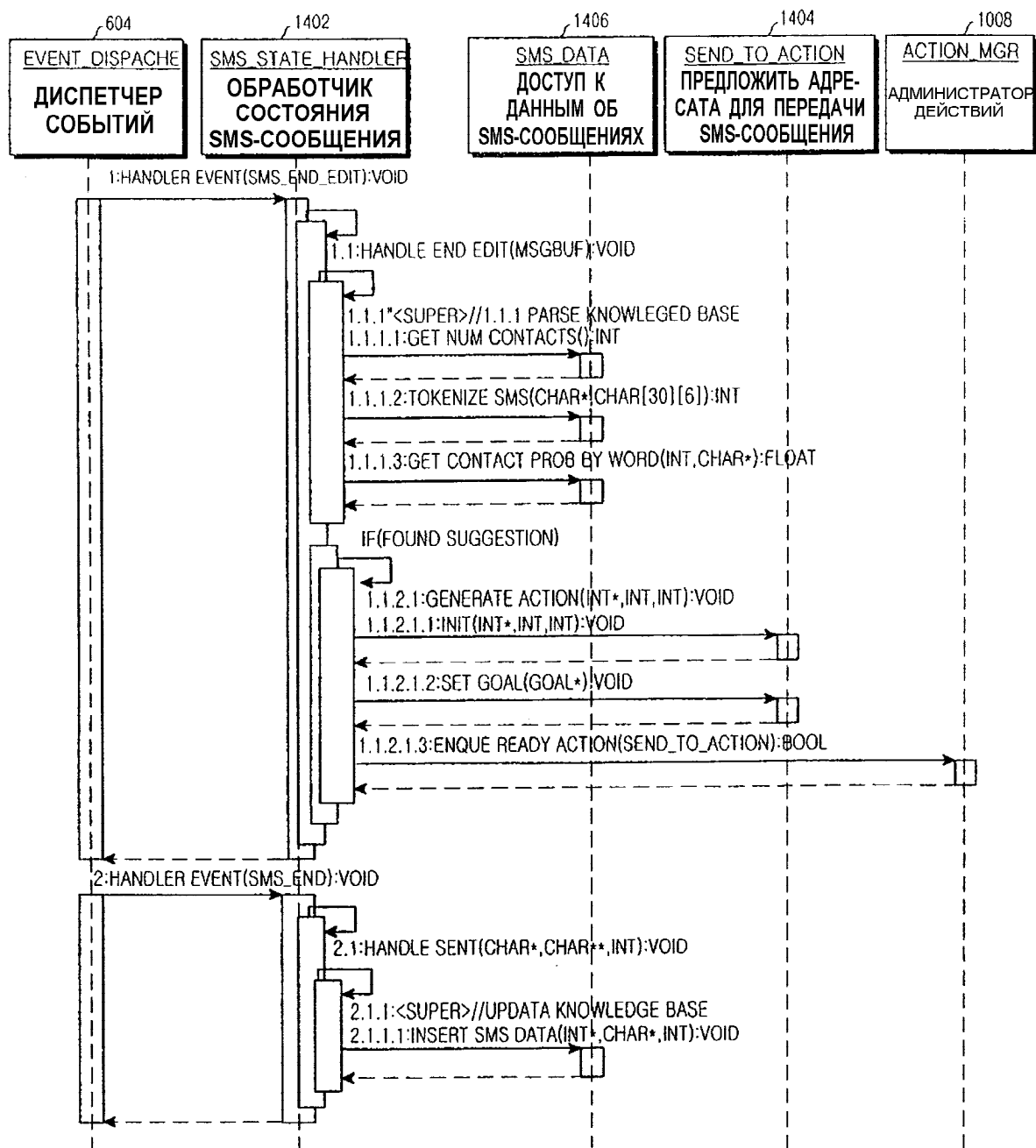
Фиг. 13А



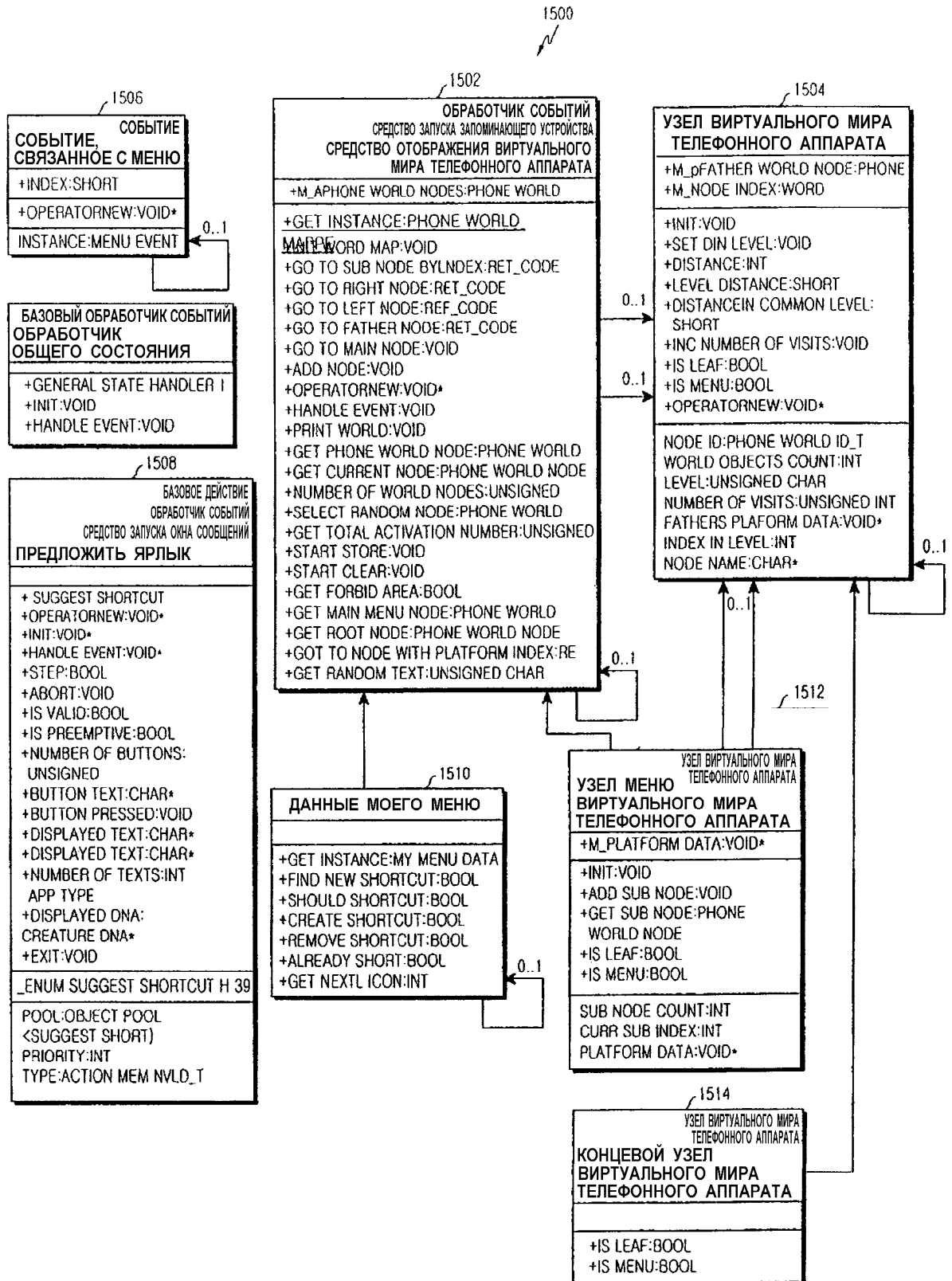
Фиг. 13В



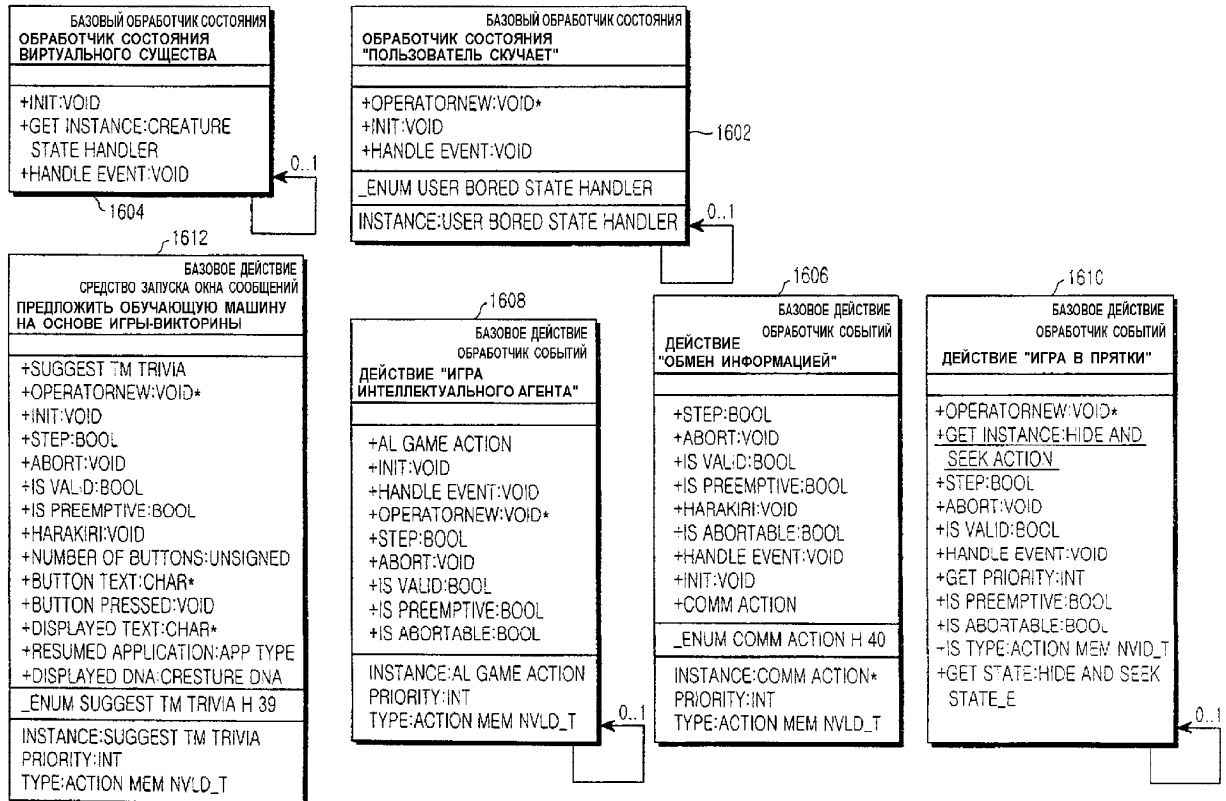
Фиг. 14А



Фиг. 14 В

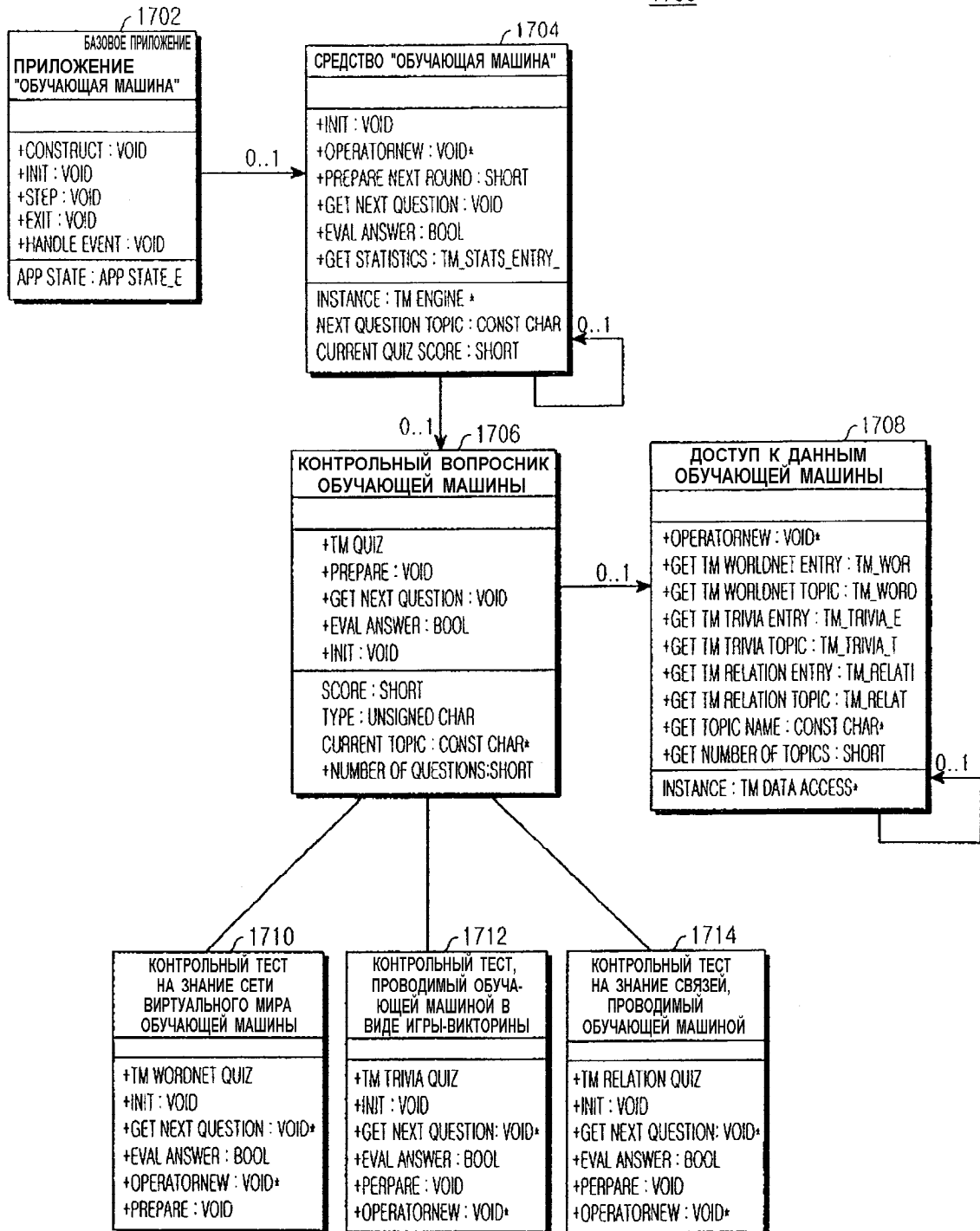


Фиг. 15

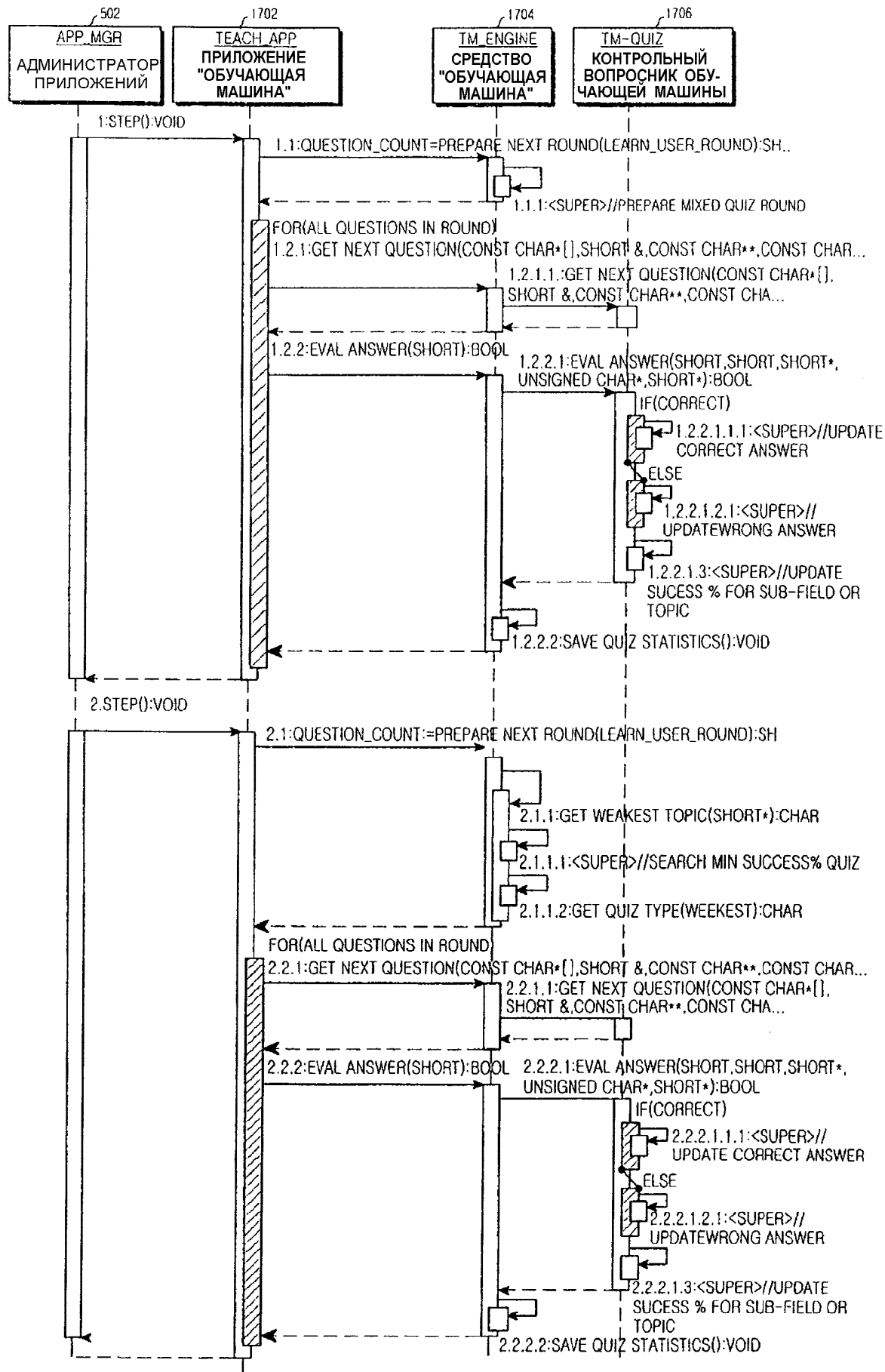


Фиг. 16

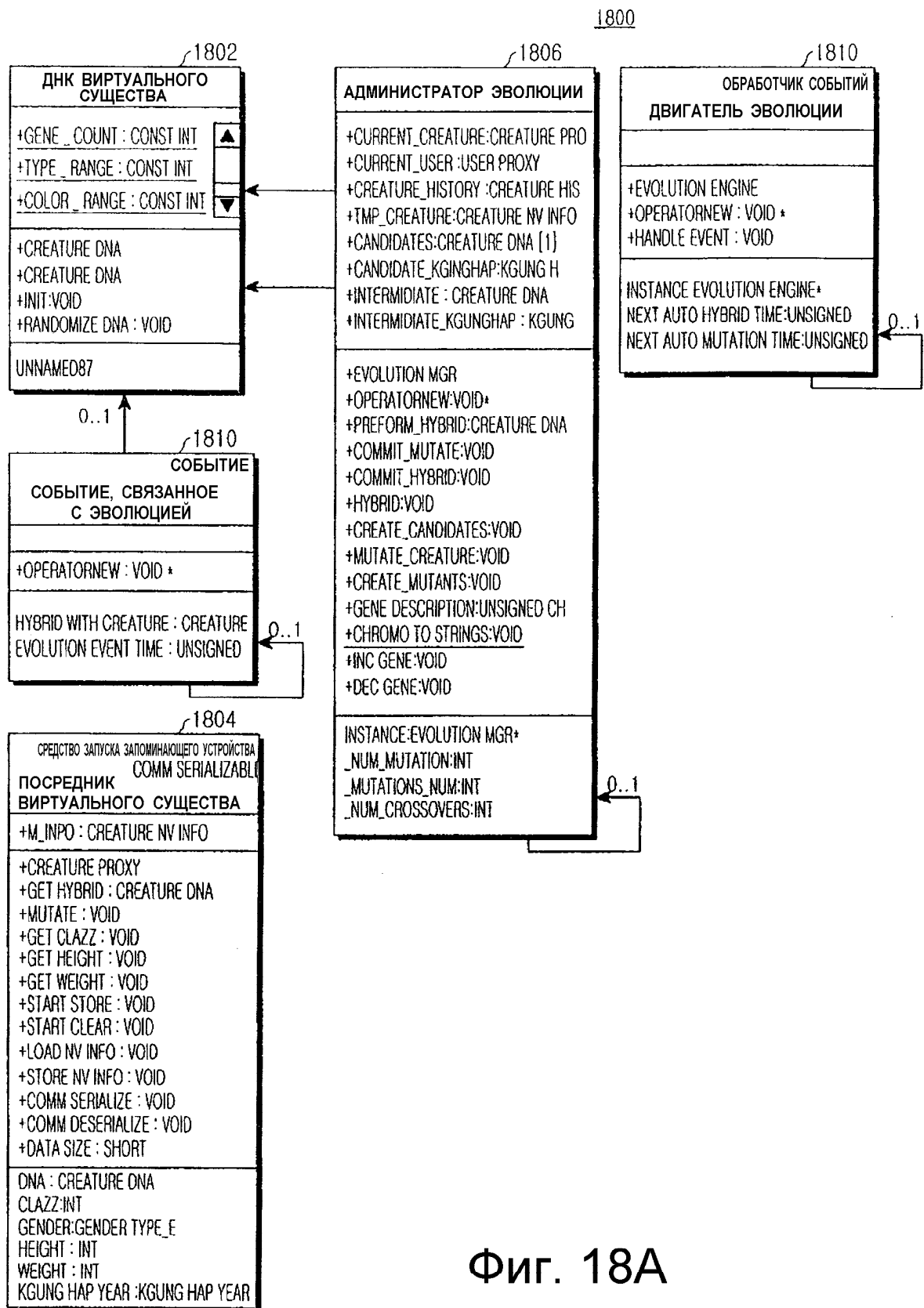
1700



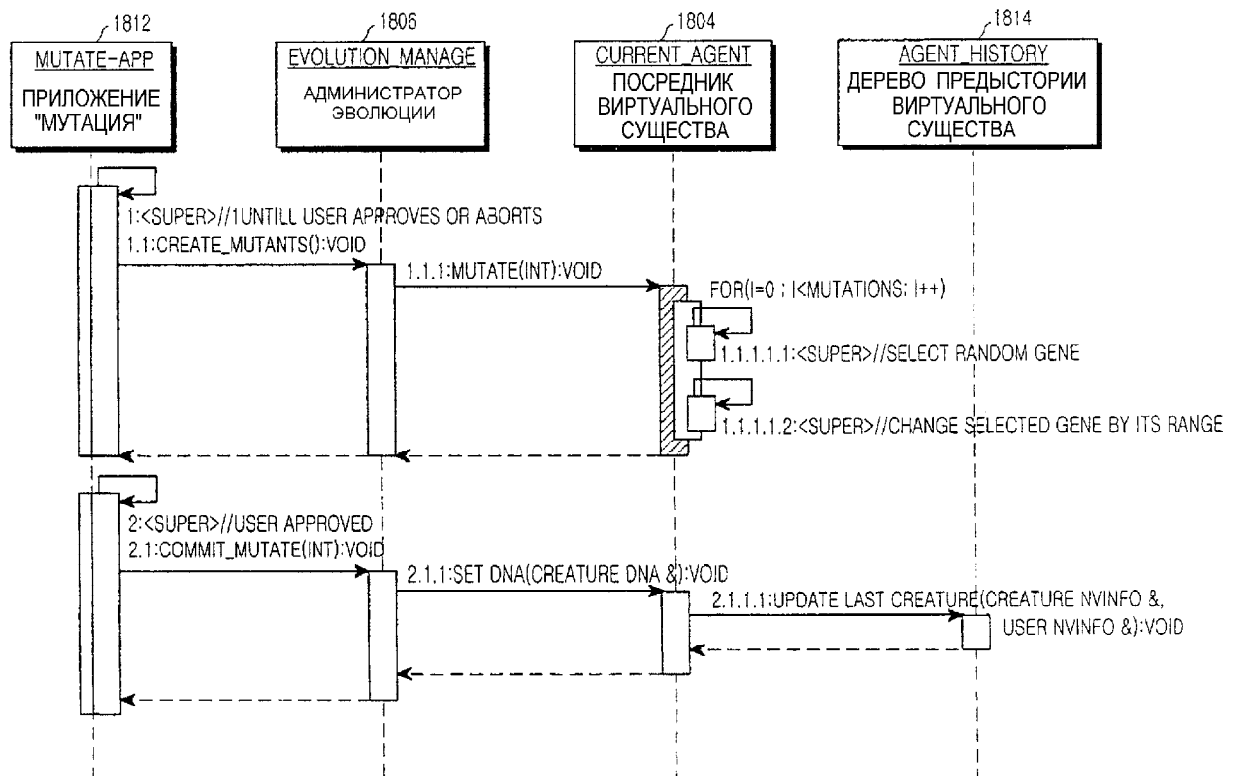
Фиг. 17А



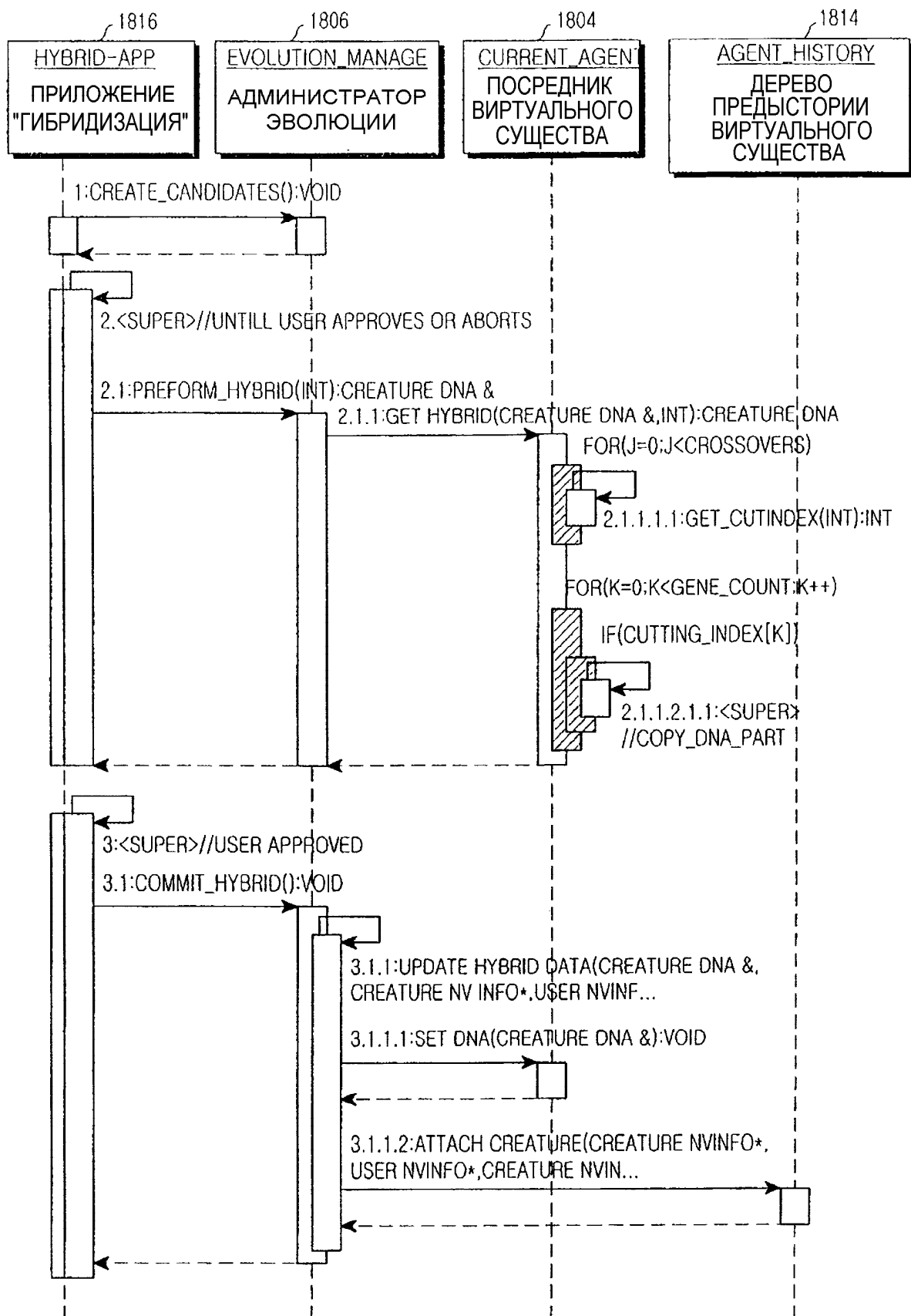
Фиг. 17В



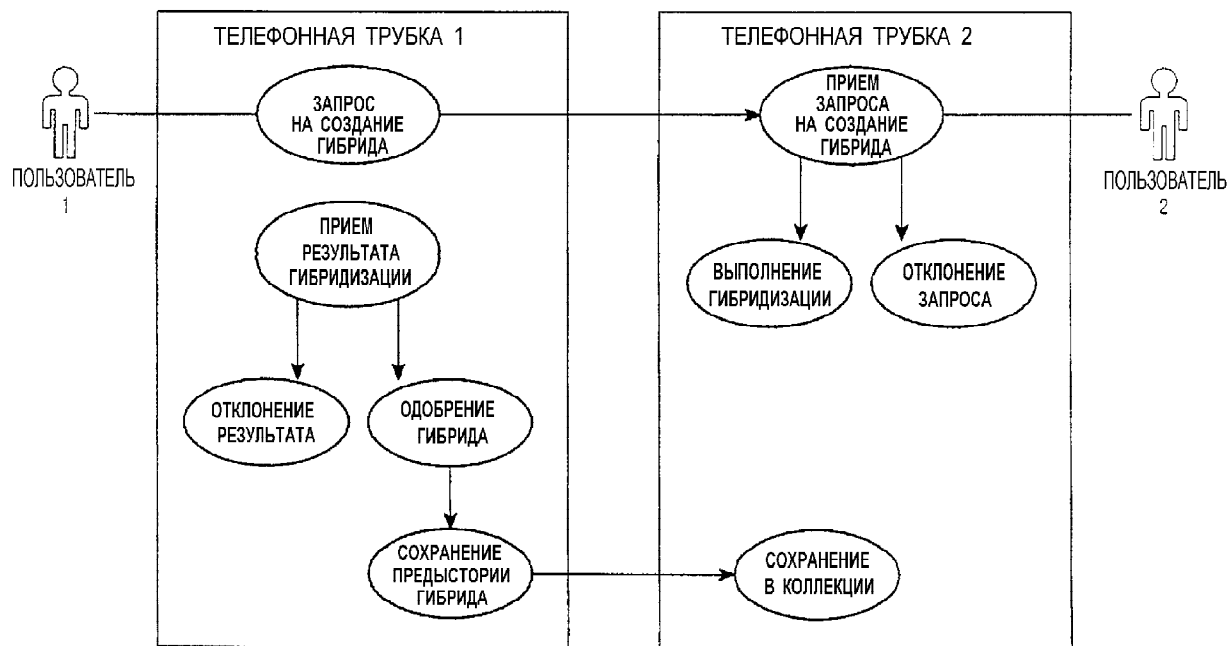
Фиг. 18А



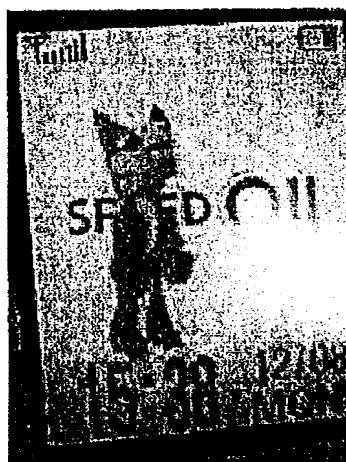
Фиг. 18В



Фиг. 18С



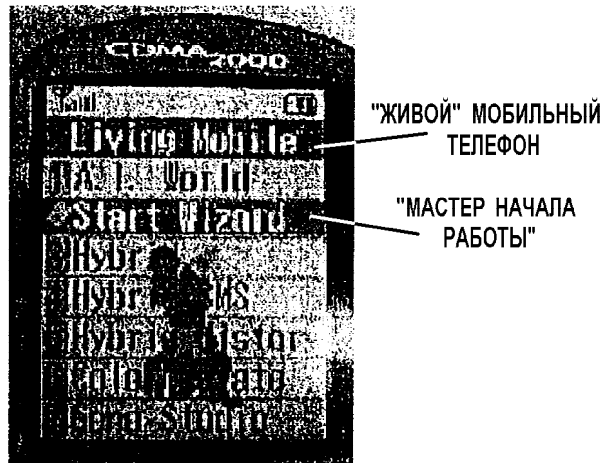
Фиг. 19



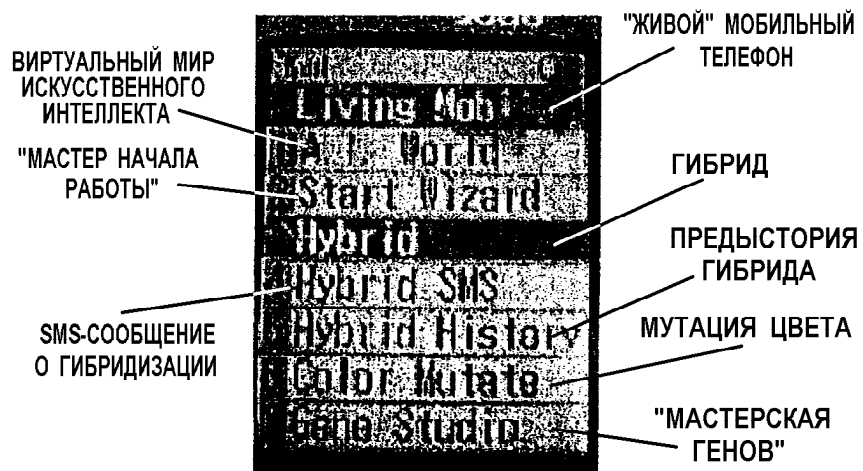
Фиг. 20



Фиг. 21



Фиг. 22



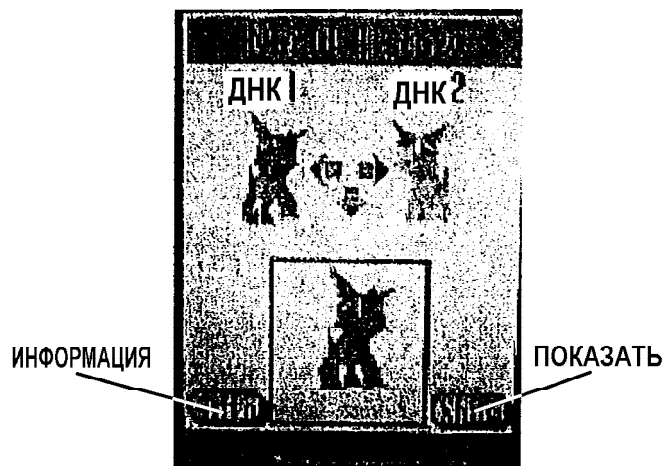
Фиг. 23



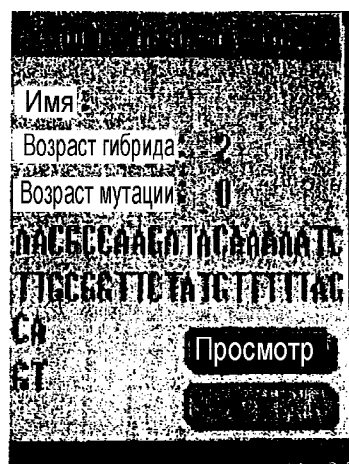
Фиг. 24А



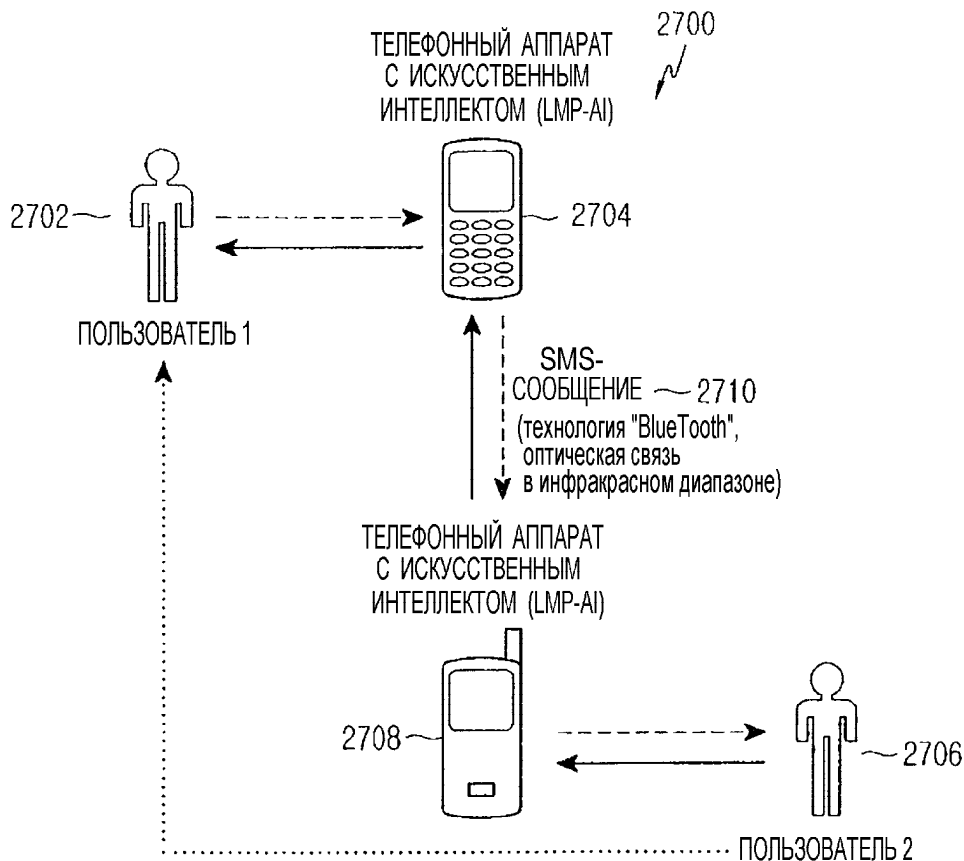
Фиг. 24 В



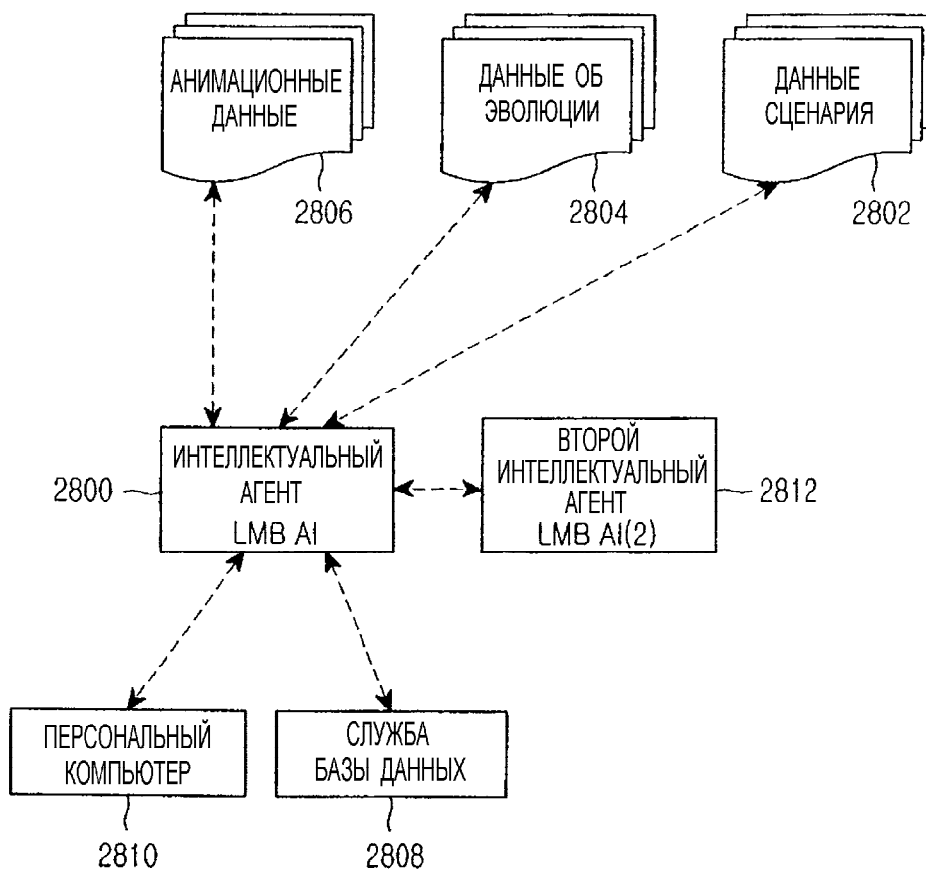
Фиг. 25



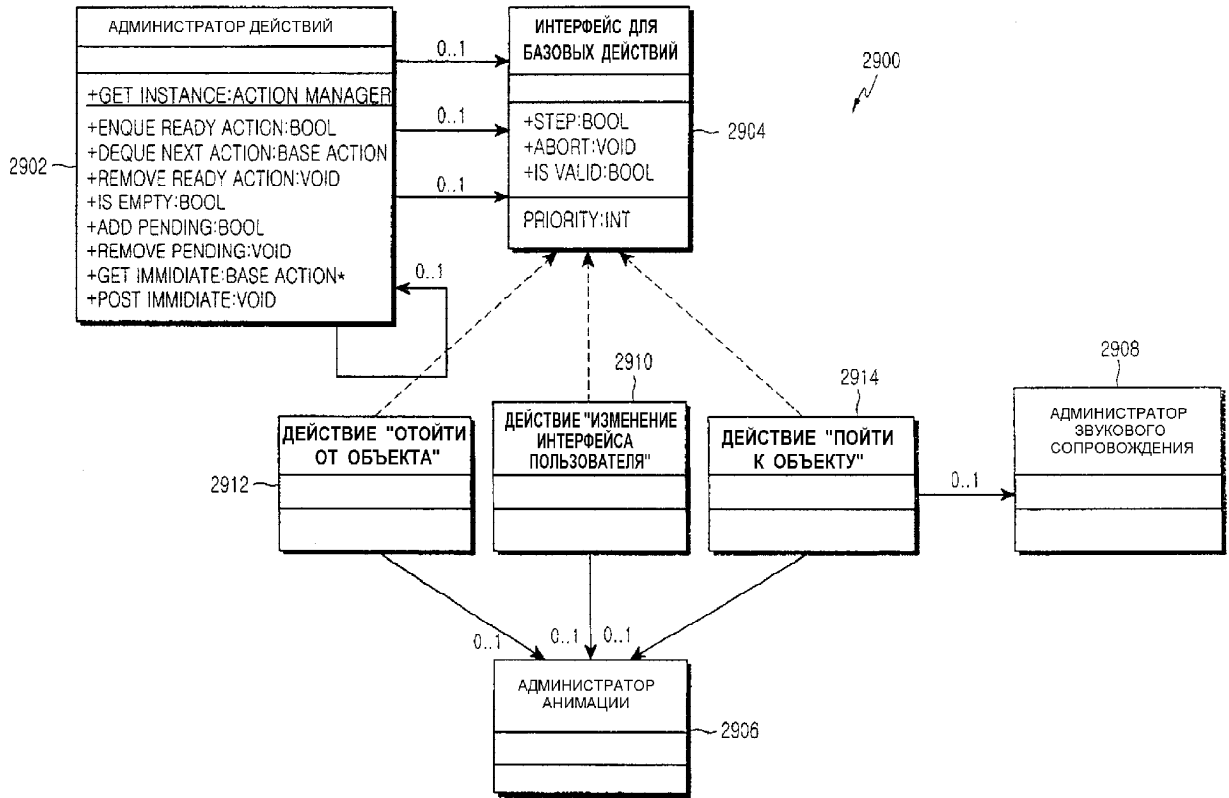
Фиг. 26



Фиг. 27



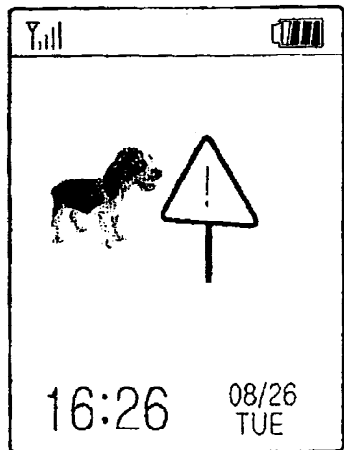
Фиг. 28



Фиг. 29



Фиг. 30А



Фиг. 30 В