

Приложение к отчету о выполнении проекта № 16-19-00044
«Принципы распределения задач между сервисными роботами и средствами
киберфизического интеллектуального пространства при многомодальном
обслуживании пользователей» в 2017 году

Работы по проекту выполнялись согласно утвержденному плану на 2017 год, включая разработку функционально-параметрической модели, программного обеспечения, методики тестирования и критериев оценки и первой демонстрационной версии системы распределенного решения задач анализа поведения пользователя в текущей ситуации и предоставления сервисов в киберфизическом пространстве. Также велась разработка конструктивных элементов торса и манипуляторов антропоморфного робота Антарес, предназначенного для решения задач сопровождения пользователей в киберфизическом пространстве. Выполнена апробация полученных результатов на международных конференциях и подготовлены публикации. В течение года велась переписка с Роспатент, 24 ноября 2017 года получено решение о выдаче патента Российской Федерации на изобретение "Способ распределения задач между сервисными роботами и средствами киберфизического интеллектуального пространства при многомодальном обслуживании пользователей". В приложении к отчету приведено расширенное описание результатов, графические материалы и ссылки на литературу. Далее кратко описаны результаты проведенных работ по проекту. Развернутое описание, включая графические материалы, формулы представлены в приложении к отчету.

В первую очередь на основе разработанной на первом этапе структурно-параметрической модели киберфизического пространства, где в качестве элементов рассматриваются мобильные робототехнические, мобильные клиентские, встроенные, стационарные и облачные компоненты, реализующие сенсорные, активационные, вычислительные, информационные функции, **была исследована и предложена функционально-параметрическая модель.**

В начале рассмотрим структуру корпоративного киберфизического пространства (КФП) на примере многомодальной информационно-навигационной облачной системы (МИНОС). На рис. 1 представлены основные компоненты системы.



Рис. 1. Структура МИНОС

Основными компонентами МИНОС являются:

- система корпоративного телевидения, позволяющая выводить актуальную информацию в соответствии с календарем событий как на мониторы, расположенные в организации, так и на мобильные устройства пользователей;
- система автоматизации оборудования, отвечающая за видеомониторинг и сохранение электроэнергии;
- система видеоконференцсвязи, предназначенная для коммуникации зарегистрированных в системе удаленных пользователей;
- система регистрации и идентификации пользователей, позволяющая как персонифицировать сервисы в соответствии с профилем пользователя, так и обнаружить нарушение режима безопасности организации;
- система навигации пользователя внутри помещений организации, предоставляющая в распоряжение пользователя трехмерную карту организации с возможностью проложить маршрут к интересующему его объекту, а также сервис сопровождения МРП.

Пусть имеется некоторое киберфизическое интеллектуальное пространство E , которое характеризуется набором параметров $\langle Hw, Sw, S, M, U, T, R \rangle$, где Hw – аппаратные средства, включающие:

- локальные стационарные, функционирующие на территории пространства,
- стационарные сервера, удаленно поддерживающие работу пространства посредством обработки больших массивов данных, поступающих с гетерогенных устройств пользователей и пространства,
- мобильные устройства пользователей, находящиеся внутри пространства или взаимодействующие с ним удаленно;

Sw – программные средства, включающие базы данных о пользователях, мультимедийный контент и служебную информацию;

S – сервисы, предоставляемые пользователям при взаимодействии с пространством;

M – формализованное представление плана здания, включающего пространственную модель помещений, доступных пользователям и мобильным роботам для передвижения;

U – профили пользователей пространства, различающиеся по категориям, предпочтениям и другим характеристикам;

T – календарь событий пространства, описывающий регулярно проводимые мероприятия, а также время уведомления о штатных ситуациях;

L – формализованное описание результатов работы пространства, включающее анализ поведения пользователей, эффективности сервисов, статистику ошибок в работе программно-аппаратного обеспечения.

Аппаратные средства, поддерживающие работу КФП, обеспечивают реализацию следующих основных функций:

1. сбор данных о текущей ситуации посредством сети гетерогенных сенсоров и радиочастотной коммуникации между устройствами;
2. обработка данных для определения текущего контекста и реализации сценариев обслуживания пользователей;
3. вывод мультимедийной информации и служебных данных на
 - локальные стационарные устройства пространства;
 - мобильные роботы;
 - клиентские устройства.

Далее опишем конфигурацию мобильной робототехнической платформы MRP , функционирующей в КФП МИНОС (рис. 2):

$$MRP = \langle H, S, SRV \rangle,$$

где $H = \langle M, T \rangle$ - аппаратное обеспечение, включающее:

$M = \{m_i, i \in N\}$, $N = \{1, \dots, n\}$ – оборудование подвижной части платформы (двигатели, плата управления, датчики расстояния, система заряда и питания, сетевое оборудование),

$T = \{m_j, j \in C\}$, $C = \{1, \dots, c\}$ – терминал, включающий камеру, микрофон, устройство вывода звука и сенсорный экран для взаимодействия с пользователем;

$S = \{S_{dist}, S_{mot}, S_{pow}, S_{CPS}, S_U, S_{cam}, S_{mic}, S_{INT}\}$ – программное обеспечение, включающее следующие модули:

S_{dist} – блок обработки информации датчиков расстояния и одометрии,

S_{mot} – блок управления движением,

S_{pow} – блок управления питанием,

S_{CPS} – блок коммуникации с КФП (посредством проводной или беспроводной связи),

$S_U = \{S_U^{pref}, S_U^{loc}, S_U^{act}, S_U^{stat}\}$ – блок обработки информации о пользователе, включающий:

S_U^{pref} – учет предпочтений пользователей,

S_U^{loc} – последнее зарегистрированное местоположение пользователя,

S_U^{act} – последние действия пользователя в КФП,

S_U^{stat} – психофизическое и эмоциональное состояние пользователя;

S_{cam}, S_{mic} – блоки обработки данных, поступающих с камеры и микрофона, включая:

S_{spch} – блок распознавания речи,

S_{face} – блок распознавания лиц,

S_{INT} – блок взаимодействия с пользователем, включающий:

S_{INF} – блок поиска необходимой информации,

S_{TV} – блок корпоративного телевидения,

S_{VC} – блок видеоконференцсвязи,

S_{esc} – блок навигации и сопровождения пользователя;

$SRV = \{SRV_{INF}, SRV_{TV}, SRV_{VC}, SRV_{esc}\}$ – сервисы, предоставляемые пользователю, включая:

SRV_{INF} – поиск необходимой информации,

SRV_{TV} – показ корпоративного телевидения,

SRV_{VC} – видеоконференцсвязь,

SRV_{esc} – навигация и сопровождение пользователя.

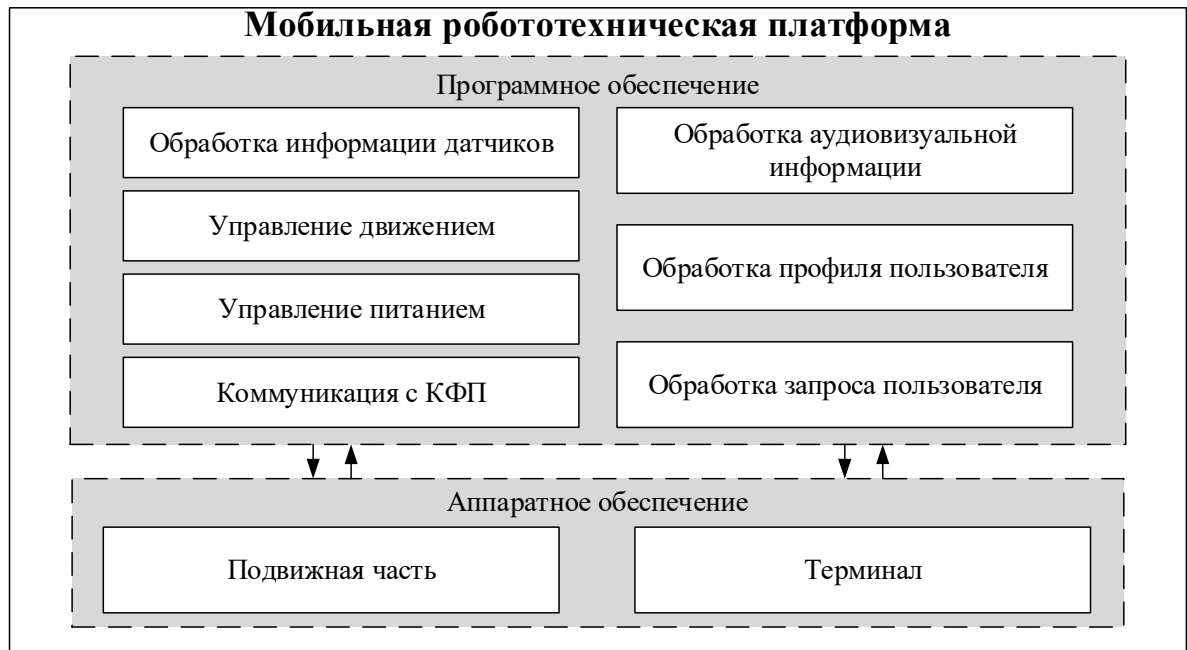


Рис. 2. Структура мобильной робототехнической платформы

МРП, работающая в интеллектуальном пространстве МИНОС, может предложить пользователю следующие сервисы [1]: в отсутствие каких-либо активных действий со стороны пользователя, в то время, пока он находится в пределах видимости МРП, но не в активной зоне D_{act} (расстояние от МРП до конца активной зоны d_{act}), на экране МРП

отображаются слайды корпоративного телевидения. В случае, если пользователь идентифицирован и имеется достаточно информации о его предпочтениях, показ слайдов может быть персонифицирован. Как правило, пользователь уже был идентифицирован и локализован другими средствами КФП, поэтому информация о профиле пользователя передается МРП от КФП.

В случае, если пользователь подходит к МРП на расстояние $d \leq d_{act}$, на экране отображается меню, предлагающее пользователю:

- 1) продолжить показ слайдов;
- 2) найти интересующую его информацию об организации;
- 3) воспользоваться видеоконференцсвязью;
- 4) показать на интерактивном плане организации интересующее пользователя помещение;
- 5) сопроводить пользователя к интересующему его помещению.

Выбрать соответствующий пункт меню пользователь может посредством голосового управления либо при помощи сенсорного экрана. В случае бездействия пользователя МРП переходит в режим показа слайдов. В случае присутствия в D_{act} двух и более пользователей, МРП выбирает для дальнейшего взаимодействия ближайшего к ней пользователя (на основании алгоритма детектирования лиц оцениваются размеры области лиц пользователей относительно размера кадра [2]). Ближайшим к МРП считается пользователь, размер области лица которого наибольший).

Информация о выборе пользователя и его поведении передается в модуль обработки профилей пользователей МИНОС с целью анализа его активности (что предусматривает не только возможность персонификации сервисов для данного пользователя, но и проверку режима безопасности, в случае, если, к примеру, он был зарегистрирован системой видеомониторинга в другом месте в это же время).

Предложенная обобщенная структура и формализованное описание МРП разработаны для интеграции МРП в КФП МИНОС, однако могут быть легко модифицированы под конкретные требования других КФП. Ввиду этого факта, а также ограниченного формата статьи, автором не делался упор на строгое описание аппаратной части. Предлагаемая МРП является модульной, предполагает событийно-ориентированное управление и может быть использована не только в рамках КФП, но и, к примеру, в модульной, роевой робототехнике.

Далее рассмотрим возможные пути взаимодействия пользователя с киберфизической системой (рис. 3). С точки зрения КФП, обобщенный профиль пользователя состоит из следующих компонентов:

- профиль пользователя в системе (если пользователь зарегистрирован);
- журналы действий пользователя, включая: историю его запросов, историю перемещений, историю проложенных маршрутов, историю вызовов видеоконференцсвязи;
- информацию о его RFID-пропуске (при наличии);
- биометрические данные (3D-модель лица, набор признаков для идентификации по голосу и т.д.);
- профиль его клиентского устройства.

КФП располагает о пользователе следующей информацией. Если пользователь зарегистрирован в системе, то в его профиле хранятся данные о ФИО пользователя, его статусе (сотрудник, посетитель, прочее), его полномочиях в системе, а также, если это сотрудник, то о подразделении организации, в котором он работает и занимаемой должности. Кроме того, в профиль может быть добавлена и другая информация, например, о дне рождения сотрудника – для показа поздравительных слайдов в модуле корпоративного телевидения.

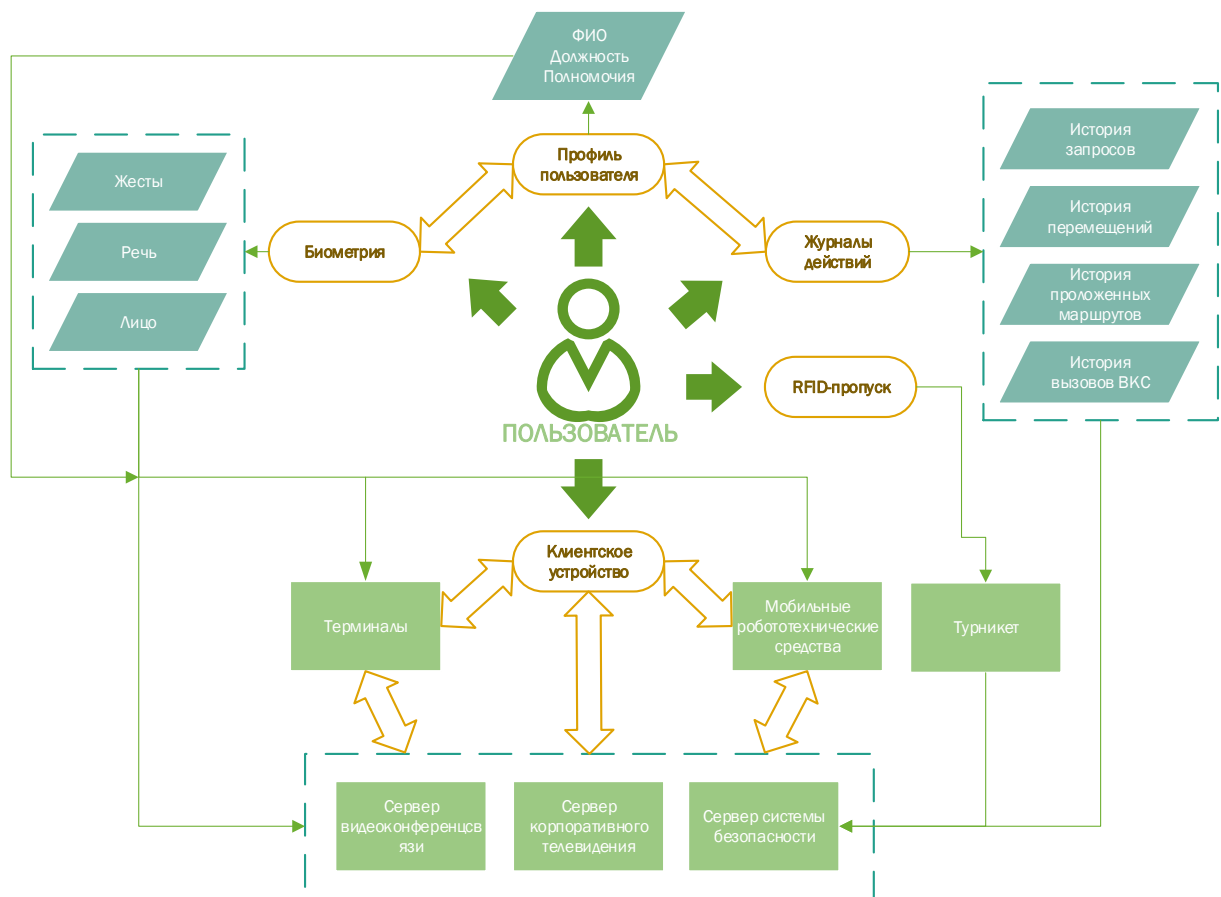


Рис. 3. Взаимодействие пользователя с КФП.

При разработке функционально-параметрической модели распределенного решения задач анализа поведения пользователя в текущей ситуации и предоставления сервисов в киберфизическом пространстве также был рассмотрен вопрос протоколов между киберфизическими компонентами, реализующих передачу служебных и мультимедийных данных в КФП посредством технологий интернета вещей с использованием гибридных устройств связи.

Использование различных протоколов и технологий в IoT-сетях ведет к отсутствию унификации модулей и увеличению количества несовместимых друг с другом решений. Современные IoT-сети не рассчитаны на передачу потоковых аудио- и видеоданных, что сужает область их применения. Также современные IoT-сети должны содержать области связи с устройствами, передающими данные в глобальную сеть интернет и учитывать аппаратную и программную специфику работы данных устройств. Авторами предложен универсальный подход к организации IoT-сетей с использованием гибридных модулей, которые обеспечивают гибкость, масштабируемость, энергоэффективность и универсальное использование сети для передачи различных типов мультимедийных данных. Данный подход учитывает программные и аппаратные особенности устройств передачи данных в IoT-сетях, что позволяет автоматизировать процесс связи модулей, выбранных пользователем.

IoT-сети являются одним из наиболее перспективных решений для повышения уровня автоматизации в различных сферах деятельности человека. В настоящее время IoT-технологии успешно внедряются как в сфере услуг, так и в реальном секторе экономики, например, в добывающей промышленности. Помимо передачи различной информации, IoT-сети могут, в частности, облегчить решение задачи локализации и навигации автономных подвижных объектов в помещениях (пользователей, робототехнических платформ и др.).

Одной из особенностей беспроводных IoT-сетей является общая эфирная среда, реализованная на основе единой технологии передачи данных, что усложняет множественный доступ «умных вещей» к эфирной среде.

Рассмотрим зависимость скорости передачи данных от расстояния и зададимся максимально возможными расстояниями между модулями для стабильной передачи необходимых типов данных.

Для определения зависимости фактической скорости передачи данных от расстояния между передающей и приемной антеннами Wi-Fi сети воспользуемся данными, представленными в документации стандартов IEEE 802.11a и 802.11g. На основе первичного анализа документации можно сделать вывод о применимости линейной регрессии для аппроксимации экспериментальных данных. Выведем уравнение линейной регрессии вида $y = ax + b$, где b - свободный член уравнения, a - угловой коэффициент или градиент оцененной линии; представляющий собой величину, на которую y увеличивается в среднем, x увеличивается на одну единицу. Исходными данными для вывода уравнения являются массив значений расстояний между передающей и приёмной антеннами $distance_{a,g}$ и фактического битрейта $bitrate_{a,g}$. Исходные данные для стандарта IEEE 802.11a:

$$distance_a = [24, 45, 60, 69, 76, 84, 91, 100], \text{ м};$$

$$bitrate_a = [54, 48, 36, 24, 18, 12, 9, 6], \text{ Мбит/с}.$$

Исходные данные для стандарта IEEE 802.11g:

$$distance_g = [30, 53, 76, 84, 100, 107, 110, 114, 122, 128, 134, 137], \text{ м};$$

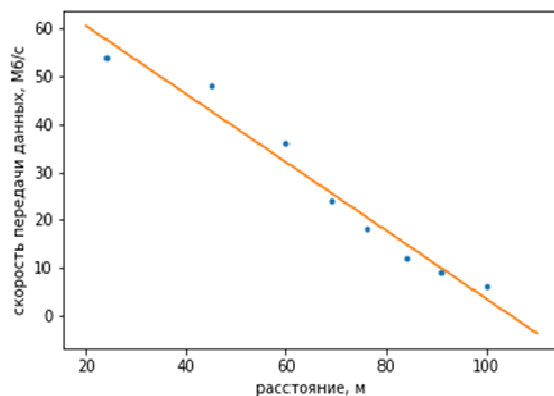
$$bitrate_g = [54, 48, 36, 24, 18, 12, 11, 9, 6, 5.5, 2, 1], \text{ Мбит/с}.$$

Для определения коэффициентов a и b воспользуемся методом наименьших квадратов. В результате проведенных расчетов получим: уравнение регрессии для стандарта IEEE 802.11a, описываемое формулой (1), уравнение регрессии для стандарта IEEE 802.11g, описываемое формулой (2):

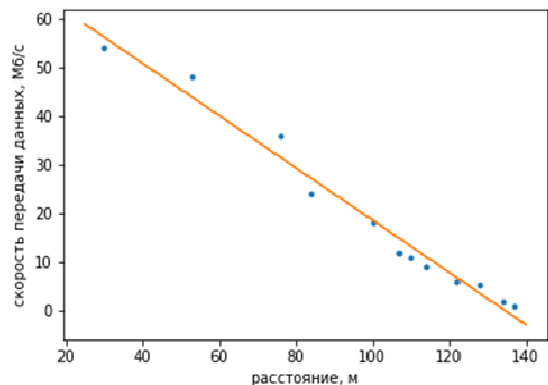
$$bitrate_a = -0.71distance_a + 74.82; \quad (1)$$

$$bitrate_g = -0.54distance_g + 72.3. \quad (2)$$

В формулах (1) и (2) $bitrate_{a,g}$ - фактическая скорость передачи данных, Мбит/с, $distance_{a,g}$ - расстояние между передающей и приёмной антенной. На рис. 4 представлены графики, полученные на основе выведенных формул.



а)



б)

Рис. 4. Зависимость фактической скорости передачи данных от расстояния между передающей и приемной антенной: а) стандарт IEEE 802.11a, б) стандарт IEEE 802.11g

На графиках видно, что указанные стандарты способны поддерживать минимальный битрейт на уровне 10 Мбит/с на расстоянии 80-90 м в помещении без учета преград и перекрытий. Для определения зависимости фактической скорости передачи данных от расстояния между модулями Bluetooth воспользуемся формулой (3) [3], полученной на основе формулы Фрииса для распространения радиосигнала в свободном пространстве [4]:

$$RSSI = P_o - 10n \lg \left(\frac{d}{d_o} \right) \quad (3),$$

где d - расстояние от устройства до передатчика, м; d_o - расстояние от устройства до точки, в которой выполнялось измерение мощности сигнала P_o устройства (принималось равным 1 м); P_o - мощность сигнала устройства, измеренная на единичном расстоянии d_o от устройства (на расстоянии 1 м имеет значение 40 дБм); n - коэффициент потерь мощности сигнала для распространения в воздухе (принимался равным 2, преграды не учитывались); $RSSI$ - полная мощность принимаемого приемником сигнала, дБм. По фактическим значениям $RSSI$, взятым из официальной документации стандарта Bluetooth IEEE 802.15.1, построим уравнение регрессии вида $y = ax + b$. Исходными данными являются:

$$distance = [0, 10, 20, 40, 60, 80, 100] \text{ м};$$

$$RSSI = [-35, -38, -36, -37, -45, -48, -43], \text{ дБм}.$$

Переменная $distance$ является массивом значений расстояний, а переменная $RSSI$ - массив значений мощности принимаемого сигнала.

Для построения зависимости скорости передачи данных от расстояния между модулями, работающими по технологии Bluetooth, преобразуем значения $RSSI$ из дБм в Вт, воспользовавшись формулой (4):

$$P_w = 10^{((P_{dBm} - 30)/10)}. \quad (4)$$

Далее рассчитаем ёмкость канала по теореме Шеннона-Хартли(5):

$$C = B \log_2 \left(1 + \frac{RSSI}{N} \right), \quad (5)$$

где C - ёмкость канала; B - полоса пропускания канала, $RSSI$ - полная мощность сигнала, N - полная шумовая мощность. На рис. 5 представлен график, отражающий зависимость скорости передачи данных от расстояния между Bluetooth-модулями.

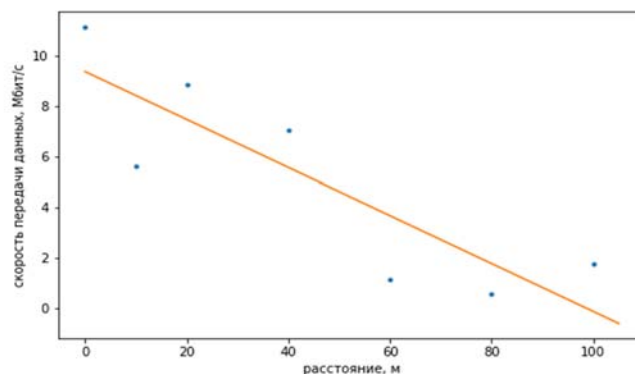


Рис. 5. Зависимость фактической скорости передачи данных от расстояния между передающей и приёмной антеннами (стандарт IEEE 802.15.1)

На графике видно, что Bluetooth-модули, в отличие от Wi-Fi-устройств, способны поддерживать фактическую скорость передачи данных на уровне 3,5 Мбит/с на расстоянии 65-70 м (в помещении без учета влияния преград). Данный показатель является удовлетворительным для передачи текстовых сообщений.

Для передачи различных типов данных посредством IoT-сетей устройства системы должны обеспечивать необходимую пропускную способность каналов связи. Для более эффективной и быстрой передачи аудио- и видеоданных по сети используются кодеки, которые позволяют уменьшить объем передаваемых данных. Чтобы задаться минимальными параметрами скорости передачи данных, которым должны удовлетворять IoT-устройства, рассмотрим наиболее распространённые и широко используемые кодеки. В табл. 1 приведена необходимая минимальная скорость передачи данных для каждого из наиболее популярных кодеков (данные приведены для разрешения 640x480, кадровой частоты 30 кадров в секунду).

Таблица 1

Фактическая скорость передачи выходных основных видеокодеков			
Кодек	H.264	MJPEG	VP8
Необходимый битрейт, Кбит/с	736	17800	11064

Из таблицы 1 видно, что наиболее требовательным к ресурсам сети является кодек MJPEG, поэтому возьмем его битрейт в качестве минимального порога пропускной способности канала при передаче видеоданных. Далее рассмотрим основные аудиокодеки.

Среди кодеков, применимых для кодирования речевых звуков и фонем, а также музыкальных звуков, можно выделить следующие (табл. 2).

Таблица 2

Фактическая скорость передачи выходных данных популярных аудиокодеков						
Кодек	MP3	FLAC	Apple Lossless Audio Codec	Opus	iSAC	iLBC
Необходимый битрейт, Кбит/с	320	1200	256	510	56	16

Из таблицы 2 видно, что наиболее требовательным к ресурсам сети является кодек FLAC, поэтому возьмем его битрейт в качестве минимального порога пропускной способности канала при передаче аудиоданных. На основе объёма данных, генерируемых различными кодеками, можно примерно оценить предложенную нагрузку на IoT-сеть.

Для выбора устройств связи в системе, рассмотрим основные характеристики популярных технологий реализации беспроводных IoT-систем (табл. 3).

Таблица 3

Основные технологии для создания IoT-сетей					
Характеристики	ZigBee	Z-Wave	Insteon	Bluetooth 4.0	Wi-Fi
Используемые частоты	2,4 ГГц	868-908 МГц	~915 МГц	2,4 ГГц	2,4 ГГц
Пропускная способность	До 250 Кбит/с	До 100 Кбит/с	1,2 – 180 Кбит/с	До 1 Мбит/с	До 300 Мбит/с
Дальность сигнала в помещении	10 – 20 м	10 – 30 м	До 25 м	До 60 м	До 200 м
Максимальное количество устройств в сети	65536	232	Не ограничено	8 устройств на одну пикосеть, макс. 10 пикосетей.	30
Характеристики	ZigBee	Z-Wave	Insteon	Bluetooth 4.0	Wi-Fi
Поддерживаемые режимы	Клиент, точка доступа (CC2530)	Клиент, точка доступа	Сеть из независимых устройств	Клиент, точка доступа	Клиент, точка доступа, клиент и точка доступа
Количество подключаемых устройств к одному устройству	8	232	Не ограничено	7	30

Современные IoT-системы могут быть реализованы множеством способов, но для их унификации необходимо определить ряд технологий передачи данных, обладающих низким энергопотреблением и необходимой пропускной способностью для различных типов данных. Наиболее перспективными технологиями, согласно приведенным характеристикам и анализу литературы, представляются Wi-Fi и Bluetooth. Рассмотрим их наиболее распространенные и доступные по цене встраиваемые аппаратные решения. В табл. 4 представлены основные характеристики модулей НМ-10, НМ-11, НМ-12.

Таблица 4

Основные характеристики модулей НМ-10, НМ-11, НМ-12

Название	НМ-10	НМ-11	НМ-12
Версия Bluetooth	4.0 BLE	4.0 BLE	4.0 EDR и BLE
Рабочее напряжение	3.3 V	3.3 V	3.3 V
Скорость передачи	9600 бит/с	115200 бит/с	115200 бит/с
Чувствительность	≤ -84 дБм на 0.1%	≤ -84 дБм на 0.1%	≤ -84 дБм на 0.1%

Основываясь на данных таблицы 4 можно сделать вывод, что технология Bluetooth 4.0 может обеспечить передачу массивов текстовых данных при низких энергозатратах, но ее скорость и пропускная способность канала недостаточны для передачи аудио- и видеопотоков данных. Далее рассмотрим Wi-Fi технологию, которая обеспечивает достаточную пропускную способность для передачи мультимедийных данных.

Помимо Bluetooth-модулей, для передачи данных в IoT-системах используется, к примеру, широко распространенная линейка модулей ESP. Используя модуль ESP32, можно обеспечить передачу мультимедийных данных в IoT-сети при помощи Wi-Fi технологии. Технология Bluetooth позволит организовывать пикосети в режиме низкого энергопотребления и осуществлять передачу текстовых данных. Таким образом можно распределить нагрузку по передаче различных типов данных между несколькими стандартами сетей и обеспечить параллельную передачу информации. Кроме того, Bluetooth-технология позволяет создавать системы для локализации различных объектов в пространстве. Для узлов сети, не требующих передачи потоковых аудио- и видеоданных, можно использовать Bluetooth-модули линейки HC.

Для настройки гибридных сетей конечным пользователем, выбора оптимальных зон установки шлюза и устройств для соединения с глобальной сетью интернет необходимо создать автоматизированную систему, которая учитывала бы показанную выше разницу в скорости передачи данных различными модулями, а также их радиус действия. Для решения этой задачи были разработаны алгоритмы функционирования данной системы, представленные на рис. 6 и 7.

На вход алгоритма (см. рис. 6) подается массив необработанных устройств IoT-сети. Для каждого из устройств последовательно выполняются следующие шаги. Извлекаются данные о типе устройства, с помощью которых необходимо определить, с какими типами устройств может быть произведено сопряжение данного модуля. Определив тип устройства, система ищет ближайшее устройство для дальнейшего сопряжения. Если зоны работы устройств не пересекаются, то в сеть будет добавлено необходимое количество повторителей сигнала, которые обеспечат связь одного устройства с другим. После обеспечения взаимодействия зон работы устройств будет организована связь между ними. Для корректной работы алгоритма необходимо проверить количество оставшихся возможных связей у устройства. Если лимит связей исчерпан, то устройство помещается в массив недоступных и больше не будет использовано для поиска связей. Последним шагом будет проверка наличия еще необработанных устройств. После завершения работы данного алгоритма получим связи всех устройств IoT-сети.

Следующий алгоритм, представленный на рис. 7, позволит создать соединение локальной сети с глобальной сетью интернет. На вход алгоритма, представленного на рис. 7, подаются данные о конечном устройстве для соединения с глобальной сетью

интернет. Эти данные используются для создания области оптимальных зон установки шлюза или выбора ближайшего устройства с необходимым типом связи: пользователь должен указать точку в пространстве или устройство из IoT-сети.

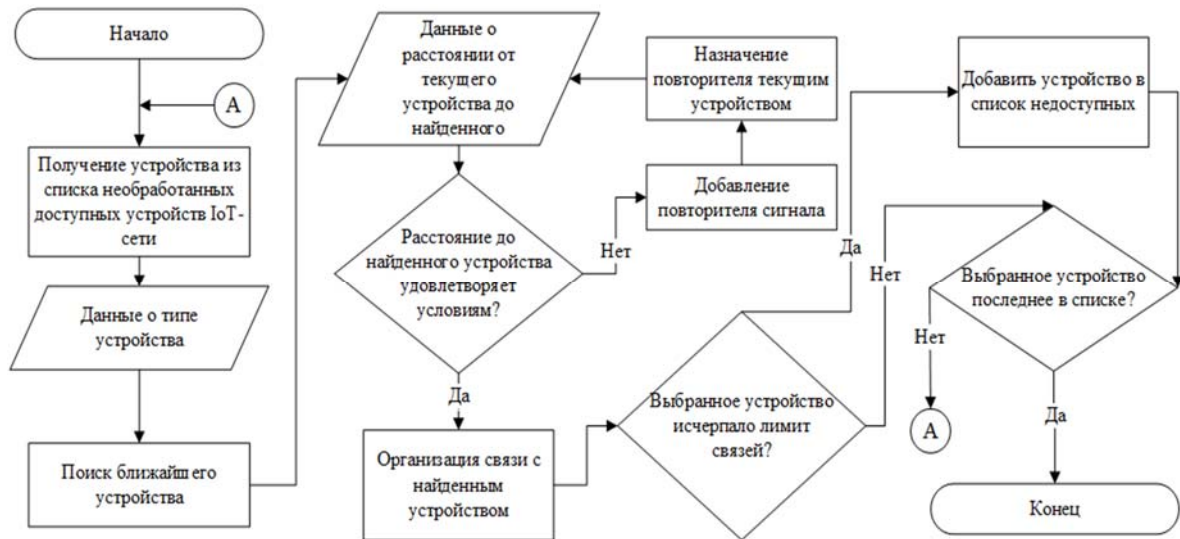


Рис. 6. Алгоритм создания связей между IoT-устройствами

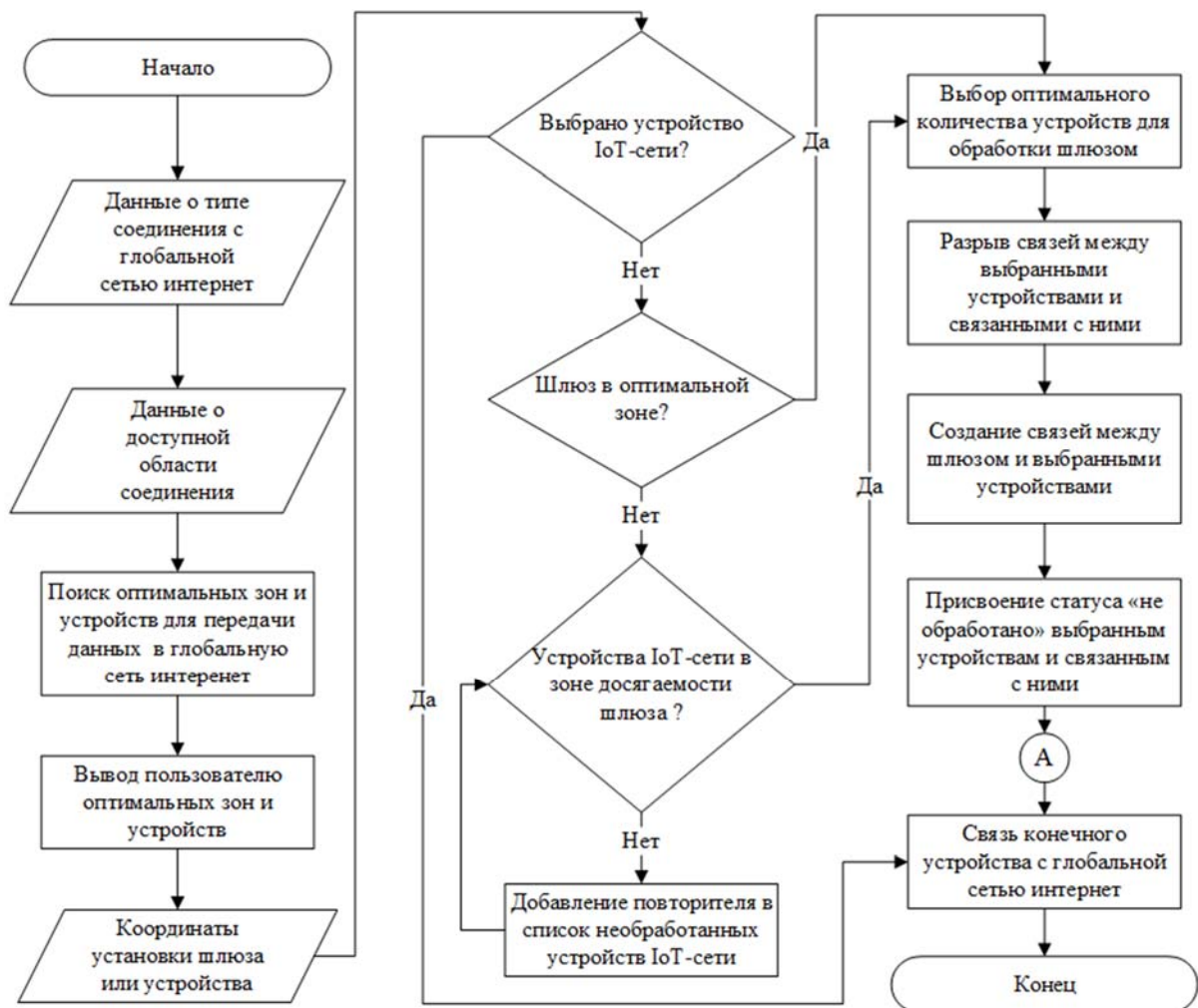


Рис. 7. Алгоритм соединения локальной IoT-сети с глобальной сетью интернет (блок А относится к рисунку 6)

Далее система проверяет данные, введенные пользователем. Если выбрано устройство IoT-сети, то алгоритм переходит к последнему этапу работы. В случае выбора точки установки шлюза будет произведена проверка на оптимальность зоны работы шлюза и наличие в зоне действия шлюза устройств. Если зоны действия шлюза будет недостаточно для установления связи со всеми устройствами, в сеть будут добавлены повторители сигналов, иначе выбирается оптимальное количество устройств для соединения со шлюзом. Связи данных устройств и связанных с ними модулей разрываются и им назначается статус «не обработано». После этого данные устройства будут переданы на обработку алгоритму создания связей между IoT-устройствами (см. рис. 6). Последний этап – связь конечного устройства с глобальной сетью интернет.

Основываясь на предложенных алгоритмах настройки связей между IoT-устройствами, опишем интерфейс системы (рис. 8).

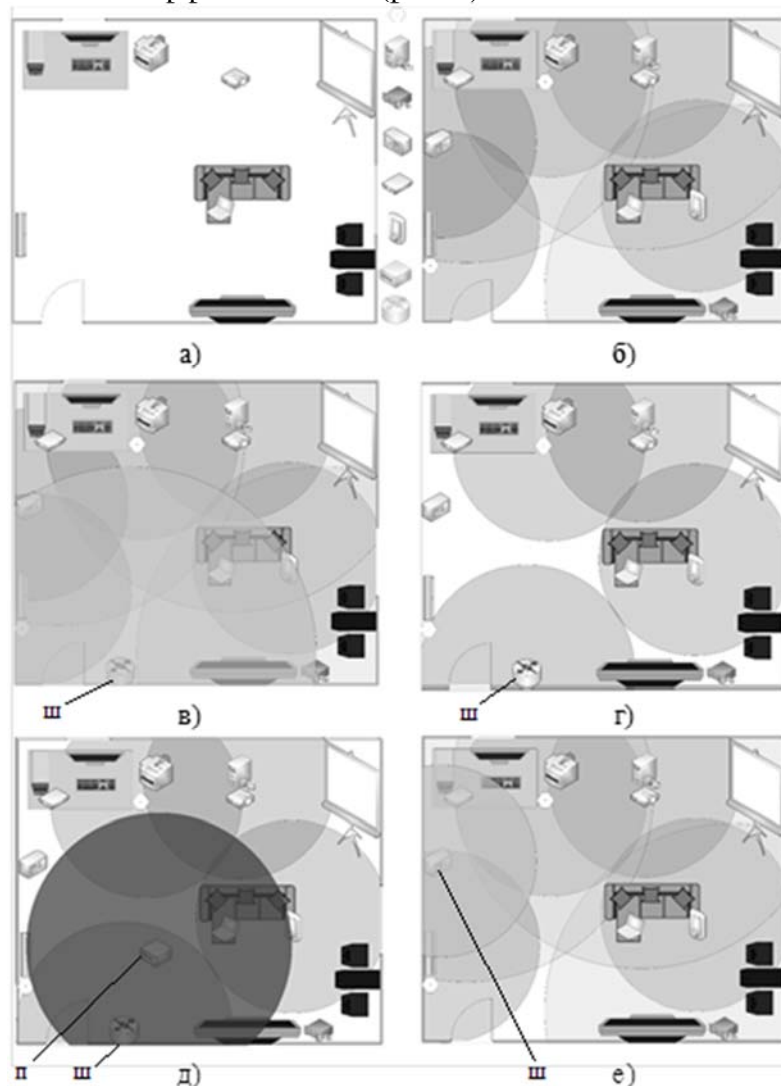


Рис. 8. Интерфейс системы настройки гибридной сети

На первом этапе (а) пользователю доступны все устройства (кроме шлюза и повторителя), которые он может расположить на свободной территории комнаты. После того, как пользователь расставил все устройства, система покажет зоны действия устройств (б): больший диапазон действия имеет Wi-Fi-модуль, меньший – Bluetooth-модуля. Если пользователь в качестве шлюза выбрал отдельное устройство, система предложит установить его в удобном месте ((в): шлюз обозначен буквой «ш»), после чего система на основе зоны действия шлюза определит возможность создания сети. В случае, если зоны действия шлюза недостаточно для надежной работы сети (г), система предложит установку повторителей сигнала в необходимых для этого местах ((д): повторитель обозначен буквой

«п»), и будет ожидать подтверждения пользователем установки повторителей, либо переустановки шлюза в другое место. В случае, если пользователь решил сделать выбранное IoT-устройство шлюзом, система покажет обновленную зону действия шлюза-устройства ((е): устройство-шлюз обозначено буквой «ш») и определит, сможет ли шлюз связаться со всеми устройствами в сети. После того, как были проведены необходимые расчеты для настройки сети, система будет ожидать подтверждения пользователем выбранной конфигурации.

Далее рассмотрим особенности разработанного **программного обеспечения распределенного решения задач анализа текущей ситуации и предоставления пользовательских сервисов** в киберфизическом пространстве, отличающееся использованием интероперабельных облачных информационных и вычислительных ресурсов и реализацией компонентов многомодальных интерфейсов на различных операционных системах и гетерогенных устройствах.

Разработанные модели распределения сенсорных, сетевых, вычислительных и сервисных задач между компонентами киберфизической системы (мобильными роботами, встроенными устройствами, мобильными клиентскими устройствами, стационарным сервисным оборудованием, облачными вычислительными и информационными ресурсами) предполагают, что мобильным робототехническим комплексам отводятся только те задачи, которые не могут быть решены стационарными устройствами (например, сопровождение человека роботом). Среди стационарных компонентов киберфизической среды выделяется распределенная корпоративная информационная система, в задачи которой входит отслеживание событий (приход и уход сотрудников и гостей, проведение мероприятий), регистрация и распознавание посетителей, хранение их профилей, взаимодействие с посетителями с помощью сенсорных экранов, трансляция сообщений, новостных и информационных блоков на стационарные экраны.

Рассмотрим более подробно архитектуру системы, отвечающей за обработку событий во времени, а также диспетчеризацию и отображение информационных материалов. Среди существующих систем этому набору задач наиболее точно соответствуют системы Digital Signage. Так как к гибкости, динамичности и расширяемости компонентов киберфизической среды, предъявляются высокие требования, существующие открытые решения Digital Signage (Xibo, Concerto v2, Vodigi и др.) следует тщательно рассмотреть. Выявлено, что эти системы обладают следующими недостатками: отсутствуют средства шаблонизации, т.е. возможность автоматически формировать множество медиафайлов из одного шаблона, используя внешний источник (например, базу сотрудников организации); высокая инерционность системы даже при соответствующих настройках, что критично при необходимости отображения мгновенных оповещений (например, приветствие сотрудника, сигнал тревоги). Клиентские приложения недостаточно проработаны (например, в Xibo высокая нагрузка на браузер при большом количестве объектов в расписании), а способы задания условий для активации событий не являются гибкими. Также можно отметить зависимость от платформы (например, есть клиенты Xibo для Windows и Android, но для Ubuntu существует только альфа-версия). Поэтому была разработана новая система для отображения цифрового контента в соответствии с предъявляемыми требованиями. Разработанная система состоит из сервера, мониторов и администраторских терминалов, взаимодействующих по сети.

Рассмотрим составляющие системы подробнее. Ее ключевым компонентом является сервер. К его функциям относится хранение информации о мониторах и их группах, пользователях, отслеживание событий, хранение медиаконтента, формирование расписаний отображения медиаконтента для каждого монитора, формирование и передача мониторам медиаконтента и управляющих команд, предоставление администратору и операторам веб-интерфейса для управления системой. Рассмотрим более подробно архитектуру сервера (рис. 9.).

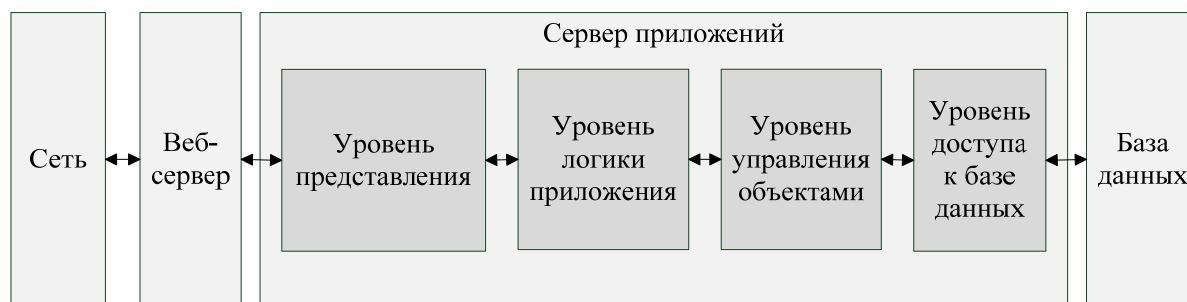


Рис. 9. Архитектура сервера

К основным компонентам сервера относятся база данных, где хранится информация об объектах информационной системы, сервер приложений, отвечающий за выполнение задач сервера, и веб-сервер, предоставляющий доступ клиентам (администраторам, операторам, мониторам).

Алгоритмы, лежащие в основе функционирования сервера, опираются на объектно-ориентированную модель данных. Классы, определяющие объекты, упорядочены в структуру (рис. 10).

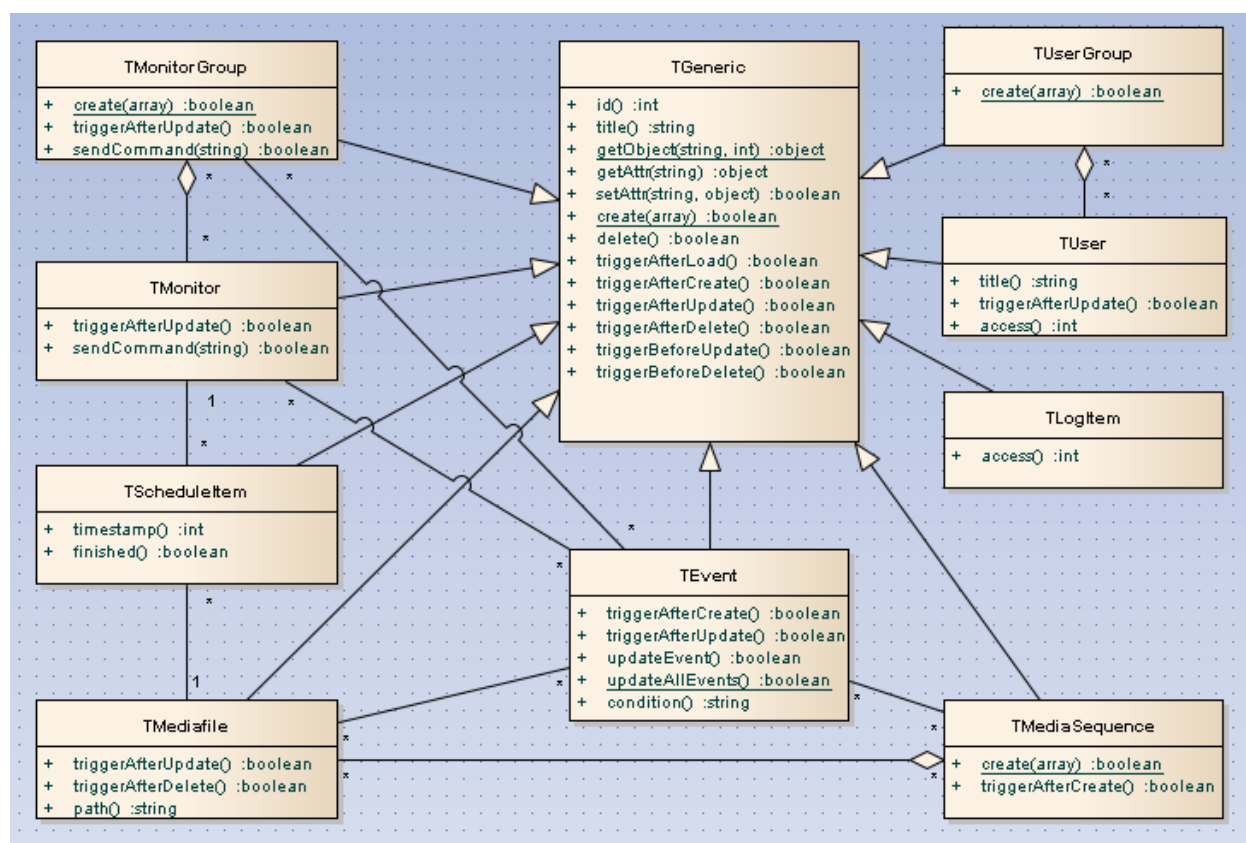


Рис. 10. Фрагмент системы классов

Все классы наследуют базовому типу TGeneric. Благодаря этому к основным, не зависящим от специфики типа, свойствам, можно обращаться через унифицированный интерфейс. Иерархия включает такие типы, как пользователь информационной системы (TUser), группа пользователей (TUserGroup), монитор (TMonitor), медиафайл – видео, изображение, HTML (TMediafile), событие, которое активируется при определенных условиях – например, в заданный момент времени (TEvent), и другие.

Основной функцией сервера является динамическое формирование расписания медиафайлов для монитора. Расписание представляет собой упорядоченный список

(очередь). Новые медиафайлы добавляются в конец очереди по мере того, как медиафайлы из его начала воспроизводятся и исключаются из очереди.

При разработке программного обеспечения были предложены новые архитектурные решения модуля видеоконференцсвязи, обеспечивающие возможность его интеграции в киберфизическую систему МИНОС, а также обобщенная структура модуля видеоконференцсвязи (ВКС) с реализацией частных и контролируемых публичных аккаунтов. Сетевое взаимодействие представлено в виде трех отдельных слоев, отвечающих за управление сетевым соединением с сервером, соединением пользователей, а также соединением для передачи мультимедийных данных участников видеоконференцсвязи. Произведены замеры времени отклика соединения при организации ВКС. В среднем, отклик для двух пользователей составил 5.81 мс.

Внедрение модуля ВКС в распределенное киберфизическое пространство (КФП) предприятия либо организации существенно расширяет возможности пользователей, а также предоставляет дополнительные инструменты для контроля безопасности и управления рабочим процессом.

Кроссплатформенные веб-технологии позволяют адаптировать работу приложения ВКС под различные программно-аппаратные платформы и обеспечивать контроль над обычными и управляемыми аккаунтами в единообразной адаптируемой среде в режиме реального времени.

Рассмотрим структуру модуля ВКС, разрабатываемого в рамках Многомодальной Информационно-Навигационной Облачной Системы (МИНОС) (рис. 11) [5, 6]. С точки зрения КФС, обобщенный профиль пользователя ВКС состоит из следующих компонентов:

- профиль пользователя в системе (если пользователь зарегистрирован);
- журналы действий пользователя, включая: историю его запросов, историю перемещений, историю проложенных маршрутов, историю вызовов видеоконференцсвязи;
- данные о его клиентском устройстве.

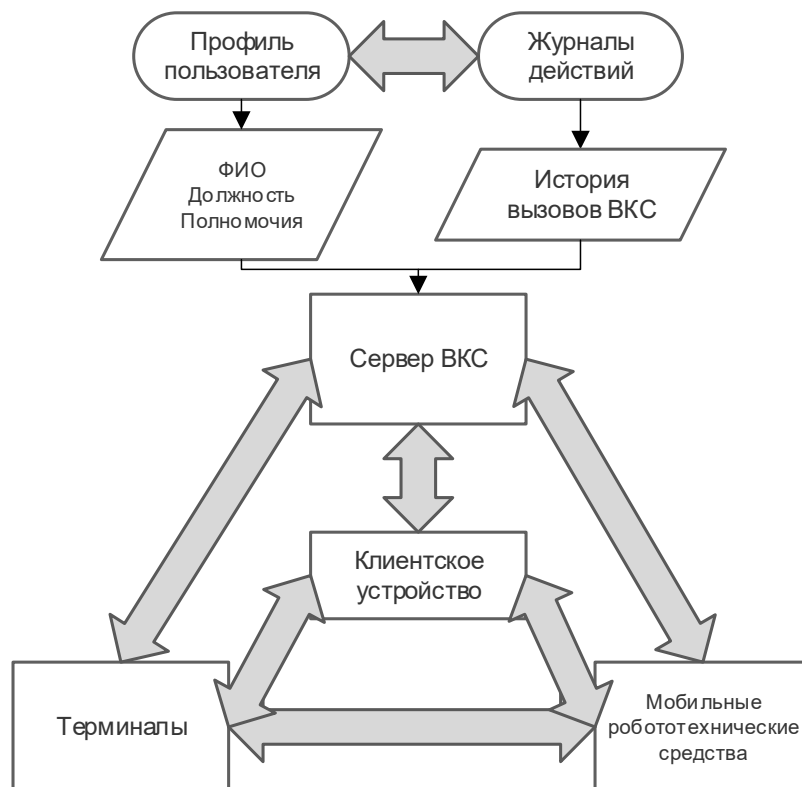


Рис. 11. Обобщенная структура модуля ВКС

КФС располагает о пользователе следующей информацией. Если пользователь зарегистрирован в системе, то в его профиле хранятся данные о ФИО пользователя, его

статусе (сотрудник, посетитель, прочее), его полномочиях в системе, а также, если это сотрудник, то о подразделении организации, в котором он работает и занимаемой должности. Кроме того, в профиль может быть добавлена и другая информация, например, о дне рождения сотрудника – для показа поздравительных слайдов в модуле корпоративного телевидения, уведомлений сотрудников по ВКС и т.д.

Отталкиваясь от предложенной схемы, рассмотрим подробнее возможную реализацию модуля ВКС в рамках КФС. Модуль ВКС должен быть доступен на устройствах, интегрированных в КФС, включая: терминалы, мобильные робототехнические платформы, рабочие компьютеры сотрудников, а также клиентские устройства пользователей. Объединение перечисленных устройств в единую локальную сеть позволит модулю ВКС функционировать в рамках организации независимо от сети Интернет, что может быть полезно при потере связи во внештатных ситуациях или для обеспечения безопасности предприятия.

Архитектура модуля ВКС пиринговая, что позволяет пользователям подключаться друг к другу напрямую, не используя сервер в качестве промежуточного звена связи. Однако, в разрабатываемом решении для КФС серверная часть используется для хранения промежуточных данных, данных пользователей и соединения с другими облачными сервисами, в частности, для авторизации пользователей и обмена сигнальными данными. Подобная реализация передачи и хранения «сигнальных» данных на клиенте и сервере обеспечивает буферизацию и последующую обработку «сигнальных» данных, исключая их потерю и поддерживая взаимодействие между группами клиентов [7].

Разрабатываемый модуль ВКС предполагает наличие так называемых публичных аккаунтов (рис. 12), обладающих двумя важными особенностями. Во-первых, публичный аккаунт предоставляет доступ к ВКС незарегистрированным пользователям (с некоторыми ограничениями); во-вторых, публичные аккаунты являются контролируруемыми – т.е. зарегистрированные пользователи могут принудительно к ним подключиться. Контролируемые аккаунты в архитектурном плане отличаются от обычных только взаимодействием с сервером. В случае с пиринговым соединением и передачей мультимедийных данных они подобны обычным аккаунтам [8].

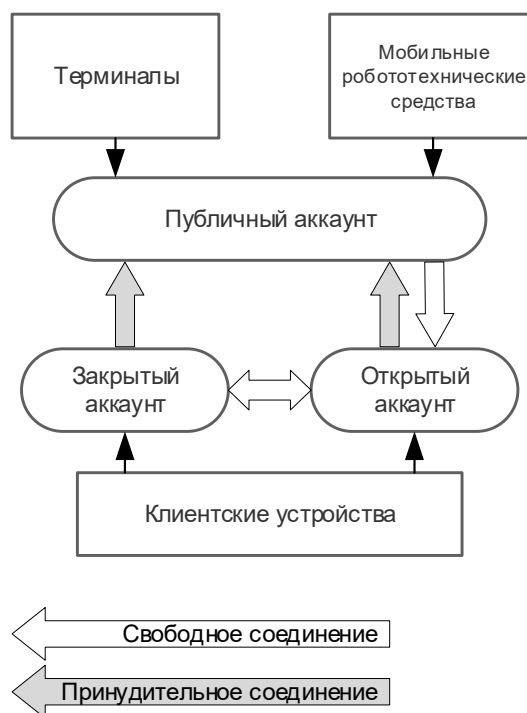


Рис. 12. Типы аккаунтов ВКС

На рис. 12 представлены типы аккаунтов ВКС и способы их взаимодействия. Аккаунты пользователей подразделяются на публичные и персональные, последние, в свою очередь, могут быть открытыми или закрытыми. Так, публичные аккаунты по умолчанию установлены на терминалах КФС и мобильных платформах, предусматривающих возможность использования ВКС. Они не требуют регистрации в системе ВКС и предназначены для оперативной связи с сотрудниками, имеющими открытый персональный аккаунт. Персональные аккаунты зарегистрированных в системе пользователей могут соединяться друг с другом без ограничений (при наличии вызываемого пользователя онлайн и принятии им вызова), кроме того, они обладают возможностью установить принудительное соединение с любым публичным аккаунтом системы. В отличие от открытых персональных аккаунтов, закрытые не доступны для связи со стороны публичных аккаунтов. При желании, зарегистрированный пользователь может войти в свою учетную запись через терминал или мобильную платформу, чтобы воспользоваться своим персональным аккаунтом.

Реализация предложенной гибридной схемы организации сетевого соединения пользователей приложения видеоконференцсвязи представлена рядом программных компонентов, объединенных в одну архитектуру. Чтобы разрабатываемое программное обеспечение соответствовало критерию пластичности, сетевое взаимодействие представляется в виде трех отдельных слоев:

1. слой, отвечающий за управление сетевым соединением с сервером приложения;
2. слой, отвечающий за управление соединением пользователей, вступающих в видеоконференцсвязь;
3. слой, отвечающий за управление соединением для передачи мультимедийных данных участников видеоконференцсвязи.

Каждый слой выражен совокупностью объектов, функций, формирующих выраженную абстракцию над логикой обработки запросов к графическому интерфейсу и обращений к соответствующему программному интерфейсу (WebRTC, WebSocket, GraphQL).

Каждый из программных компонентов, представляющих описываемые слои, в своей основе имеет схожую логику управления соединением – операции по открытию, закрытию транспортных каналов, передачи и обработки поступающих сообщений между соединившимися пользователями. В разработанном программном обеспечении данная логика отражена в классах *Transport*, *Channel* и его подтипов.

Transport инкапсулирует логику работы с создаваемым соединением, предоставляя клиентам обобщенный программный интерфейс управления (рис. 13).

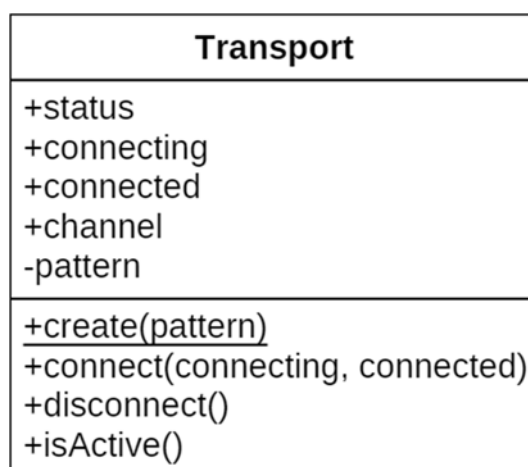


Рис. 13. Диаграмма класса *Transport*

Статический метод *create(pattern)* класса *Transport* используется для инстанцирования новых объектов класса. Метод принимает стратегию создания канала для

передачи данных между пользователями [9]. Стратегия представляет собой функцию, возвращающую объект типа *Channel* и принимающую в качестве аргументов соединяемых пользователей. При создании, объект находится в состоянии ожидания подключения, что выражено значением «inactive» свойства *status*. Метод *connect(connecting, connected)* позволяет клиенту выполнить соединение пользователей, информация о которых была передана в аргументах, с использованием переданной ранее стратегии. Информация о пользователях должна быть передана посредством объектов типа *PeerData*. При удачном соединении, объект класса переходит в состояние «active», записывает информацию о соединенных пользователях в свойствах *connecting* и *connected*, и сохраняет созданный канал в свойстве *channel*. В случае возникновения ошибок, метод *connect* генерирует исключение с соответствующим проблеме сообщением. Для отключения соединения пользователей класс предоставляет метод *disconnect()*, при использовании которого объект класса удаляет сохраненные данные о созданном соединении и переводит *status* в состояние «inactive». Таким образом, объект класса *Transport* может быть использован в различных сеансах связи.

Класс *Channel* является базовым для объектов, скрывающих в себе логику создания канала связи для обмена данными между двумя пользователями. Channel сводит различия в управлении каналов разных типов к общему для всех интерфейсу, обеспечивая единообразную работу с каналами на всех сетевых уровнях приложения (рис. 14).

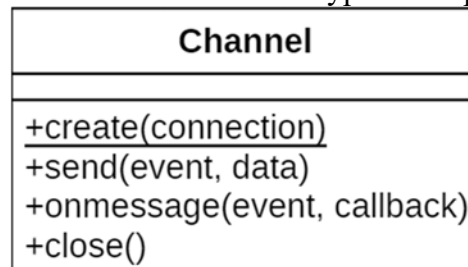


Рис. 14. Диаграмма класса *Channel*

Инстанцирование объекта класса *Channel* выполняется посредством статического метода класса *create(connection)*. В качестве аргумента метод ожидает простой объект *connection* с методами *send(event, data)*, *onmessage(event, callback)* и *close()*, оперирующими активным соединением пользователей. *Channel* выступает в роли класса-адаптера, предоставляя потомкам возможность адаптации интерфейсов создаваемых каналов к общему для всех виду. Подтипы *Channel* инкапсулируют логику создания каналов, расширяя конструктор базового класса. В конструкторе подтипов происходит выполнение соединения пользователей, переданных в аргументе, и последующий вызов конструктора базового класса с передачей созданного *connection* в аргументе.

В разработанном программном обеспечении для реализации слоев сети использованы следующие подтипы, образованные от *Channel*: *IOChannel*, *P2PChannel*, *MP2PChannel* (рис. 15).

IOChannel принимает в аргументах конструктора данные о соединяемых пользователях, представленных объектами с полями *name* и *address*. При создании объекта выполняется соединение клиента, представленного в *connecting*, с *connected* по WebSocket, используя записанный в *connected* адрес.

P2PChannel использует в качестве механизма соединения API *RTCPeerConnection* и *RTCDataChannel* из WebRTC. *P2PChannel* при создании объекта принимает в аргументах конструктора данные об соединяемых пользователях и канал передачи сигнализирующих сообщений. Канал передачи представляет собой объект, образованный от *Channel*. Канал передачи используется на этапе открытия соединения, для обмена между клиентами необходимой для установления сеанса связи сообщений. После завершения соединения канал может быть закрыт.

Класс *MP2PChannel* имеет схожую с *P2PChannel* логику работы. Для соединения пользователей он использует API *RTCPeerConnection* из WebRTC и *MediaStream API*, соединяя пользователей для обмена мультимедийной информацией.

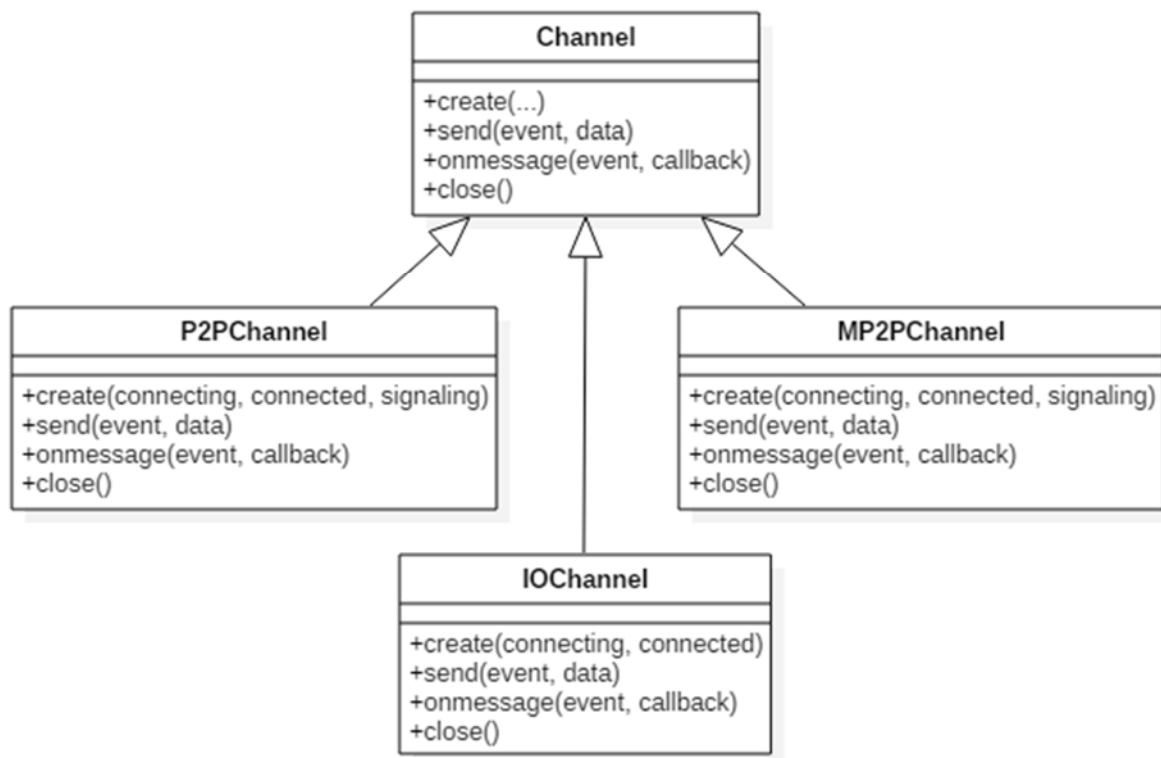


Рис. 15. Диаграмма иерархии классов, образованных от *Channel*

При вызове *connect(connecting, connected)*, объект *Transport* создает канал связи между пользователями с использованием переданной в аргументе конструктора стратегии. Таким образом, *Transport* агрегирует *Channel* или его подтипы (рис. 16).

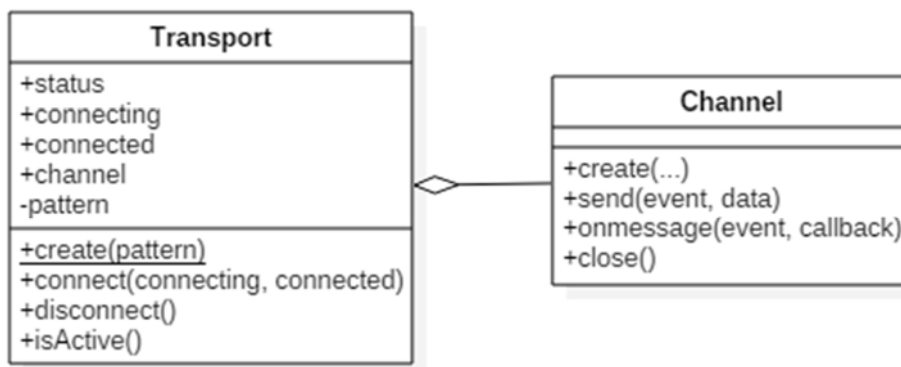


Рис. 16. Диаграмма связи *Transport* и *Channel*

Объекты класса *PeerData* в разработанном программном обеспечении используются как контейнеры для хранения информации о пользователях, участвующих в сетевом взаимодействии. *PeerData* позволяет компонентам использовать в качестве исходных данных единообразные персистентные структуры данных, обеспечивая упрощение взаимодействия внутри программного обеспечения (рис. 17).

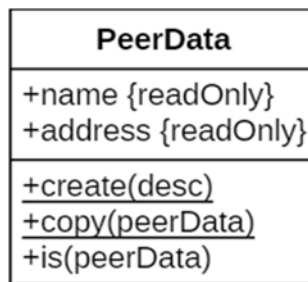


Рис. 17. Диаграмма класса *PeerData*

PeerData обладает свойствами *name* и *address*. *name* является уникальным идентификатором пользователя, участвующего в межсетевом взаимодействии. *address* предоставляет адрес, по которому пользователь может быть локализован. При создании объекта класса, конструктор *PeerData* принимает объект с информацией о пользователе, ожидая при этом, что *desc* имеет свойства *name* и *address*. В случае получения некорректного объекта-дескриптора, конструктор генерирует исключение с описанием обнаруженной проблемы.

PeerData предоставляет ряд методов, упрощающих работы с созданным контейнером. Статический метод *copy(peerData)* позволяет сделать чистую копию существующего объекта *PeerData*. Метод *is(peerData)* позволяет сравнивать объекты *PeerData*. Данные методы позволяют обеспечивать неизменяемость данных, сохраненных в *PeerData*, в процессе их использования компонентами программного обеспечения.

При успешном соединении пользователей объект *Transport* сохраняет копии переданных *connecting* и *connected* в своих свойствах. Таким образом, поля *connecting* и *connected* объекта класса *Transport* имеют тип *PeerData*. Описанные отношения между классами *Transport* и *PeerData* выражены в виде композиции, представленной на рис. 18.

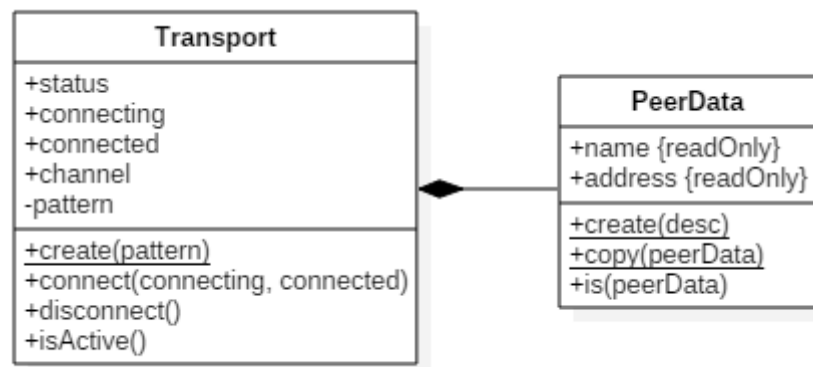


Рис. 18. Диаграмма связи *Transport* и *PeerData*

Объекты *Transport* отражают связь между парами пользователей, участвующих в сетевом взаимодействии. Каждый слой программного обеспечения сетевого взаимодействия выражен в виде отдельного модуля и сформирован определенным набором объектов *Transport*. Первый слой программного обеспечения сети использует объект *Transport* с переданной ему при создании фабричной функцией *createIOChannel(connecting, connected)*. Функция является оберткой для вызова метода *create(connecting, connected)* класса *IOChannel*. Клиентское приложение вызывает метод *connect(connecting, connected)* объекта *Transport*, используя в качестве аргументов объекты типа *PeerData* с информацией о себе и о сервере приложения, тем самым выполняя их соединение. Модуль слоя сети устанавливает ряд обработчиков сообщений на объект *Channel*, записанный в объекте *Transport*, необходимых для синхронизации состояния клиентского приложения с состоянием других пользователей. Кроме того, модуль устанавливает ряд обработчиков событий на действия пользователя, выполняемые с интерфейсом приложения, для

отправления соответствующих сообщений на сервер. Второй слой при инициации нового сеанса ВКС попарно соединяет каждого из участников связи, создавая для этого объекты типа *Transport* с передачей им фабричной функции *createP2PChannel(connecting, connected, signaling)*. В качестве сигнального сервера будет использован созданный на прошлом слое WebSocket-канал. Третий слой выполняет соединение участников ВКС согласно сформированным в процессе инициации ВКС отношениям между участниками, используя для соединения объект *Transport* с переданной ему функцией *createMP2PChannel(connecting, connected, signaling)*. В качестве сигнального сервера будет использован объект *P2PChannel*, созданный на втором слое сети.

Представленная сетевая архитектура программного обеспечения приложения ВКС обладает иерархией отношения между отдельными компонентами. Решения, принимаемые на первом слое сетевой архитектуры – например, отключения соединения с сервером приложения – может повлиять на работу второго и третьего слоя. В свою очередь, второй слой может повлиять на работу третьего. Иерархические отношения между слоями сетевой архитектуры в программном обеспечении выражены посредством композиции. На стороне клиентского слоя представлены объектами класса *NetworkNode*. При создании объект *NetworkNode* принимает ссылку на слой выше по иерархии. Класс предоставляет метод *createChild()*, создающий дочерний объект *NetworkNode*. Объект класса содержит информацию о всех соединениях на своем уровне (включая информацию о самих пользователях) и может оперировать дочерним *NetworkNode* на свое усмотрение.

Подтипы класса *NetworkNode* инкапсулируют логику работы для каждого слоя сетевой архитектуры. Отношения между уровнями сети, сформированные в приложении видеоконференцсвязи, отражены рис. 19.



Рис. 19. Организация слоев архитектуры сетевого соединения на стороне клиента

Разделение сети на несколько уровней, действующих в своих рамках практически независимо от остальных, позволяет приложению продолжать осуществлять свои функции при отказе ключевых узлов. В случае отказа сервера приложения работа пользователей не будет нарушена, поскольку между ними по-прежнему будет существовать прямое соединение. Таким образом, активные видеоконференции не будут прерваны. Помимо этого, дополнительная функциональность может быть включена в реализацию каждого из уровней сети как часть композиции модуля, обеспечивая простой алгоритм масштабирования сетевой архитектуры приложения.

В то же время, использование описанной схемы соединения обладает недостатками. Второй слой сетевого соединения, обеспечивающий прямое соединение пользователей приложения, подразумевает создание сетевого соединения по схеме «каждый-с-каждым», формируя при этом избыточное количество активных соединений участников. Тем не менее, данный подход обеспечивает надежное соединение пользователей в сети, поскольку отключение отдельных узлов сети не сможет повлиять на соединение оставшихся.

Разрабатываемая архитектура соединения пользователей в приложении ВКС обладает рядом преимуществ, способствующих конкурентоспособности разработанного приложения на фоне аналогов. Предложенная организация модулей программного обеспечения сетевого соединения приложения ВКС предоставляет пространство для внедрения дополнительных функций в приложение. В рамках КФС управляемые аккаунты могут использоваться в модулях обеспечения безопасности и управления доступом, к примеру, для обеспечения контроля за рабочими местами, производством и другими объектами. Организация хранения и обработки данных позволяет внедрить модуль ВКС в

КФП, а также сделать удобным его использование как на терминалах и мобильных робототехнических платформах, так и на клиентских мобильных устройствах.

Теперь рассмотрим разработанную методику тестирования и критерии оценки выполнения сценариев многомодального взаимодействия пользователей с мобильными роботами и другими обеспечивающими устройствами окружающего киберфизического пространства.

Взаимодействие пользователя с КФП – сложный процесс, субъективное качество которого обусловлено различными факторами. Основными показателями выполнения сценариев многомодального взаимодействия пользователей с КФП являются так называемые «качество восприятия» (КВ, Quality of Experience, QoE) и «качество обслуживания» (КО, Quality of Service, QoS). Первое понятие, как правило, рассматривается в более широком смысле и включает в себя второе в качестве одного из аспектов.

В литературе предложены следующие объективные и субъективные определения КВ: объективное КВ определяет КО, доставляемое пользователю с точки зрения измеримых показателей производительности услуг, сети и приложений, в то время как субъективное моделирует качество, воспринимаемое человеком с точки зрения эмоций, биллинга услуг и опыта взаимодействия. Показатели качества аудиовизуальных мультимедийных потоков могут быть подразделены на три уровня: уровень сервисов (service layer), уровень приложений (application layer) и уровень передачи данных (transport layer). Кроме того, КО подразделяется на следующие категории: качество реализации протоколов, качество контента и качество медиапотока. Качество каждого уровня представлено параметрами КО и может быть определено как набор показателей КВ. Различают ключевые показатели производительности (КПП, Key Performance Indicators, KPI) и ключевые показатели качества (КПК, Key Quality Indicators, KQI) для оценки и управления КО. КПП — это количественные параметры, которые отражают основные факторы конкретного элемента или сервиса КФП. КПК оценивают конкретные аспекты производительности конкретных элементов или сервисов КФП посредством нескольких параметров, включая КПП.

Субъективные показатели восприятия потребителем аудиовизуальной информации – показатели качества потребителя (ПКП, Customer Quality Indicators, CQI), – качественные показатели, которые ощущаются и воспринимаются пользователями сервисов КФП. КФП включают в себя показатели биллинга сервисов и поддержки клиентов, а также предоставляемые услуги, такие как корпоративное телевидение (КТ) и видеоконференцсвязь (ВКС). Таким образом, КПК определяются как параметры КО, которые составляют ПКП, а КПП как метрики, составляющие КПК. КПК сопоставляются с ПКП и могут вычисляться на основе нескольких КПП.

В данной работе не ставилось целью оценивание субъективных показателей КВ, посему ограничимся теми параметрами КО, которые дают наиболее эффективную объективную оценку предоставленным пользователю сервисам. При разработке методики оценивания выполнения сценариев многомодального взаимодействия пользователей с мобильными роботами и другими обеспечивающими устройствами окружающего киберфизического пространства предложено использовать следующие параметры:

- модуль ВКС:
 - время отклика приложения
 - время установления соединения
- качество передачи аудиосигнала
 - качество передачи видеосигнала
 - задержка сигнала
 - потеря данных аудиопотока
 - потеря данных видеопотока
 - зашумленность аудиопотока
 - зашумленность видеопотока

- модуль КТ:
 - корректность отработки расписания показа слайдов
 - корректность персонифицированного показа слайдов (зависит от показателей системы распознавания пользователей)
- модуль навигации:
 - погрешность локализации пользователя
 - погрешность построения маршрута
- модуль поиска информации
 - корректность обработки запроса
- модуль сопровождения пользователя:
 - погрешность локализации мобильного робота
 - погрешность локализации пользователя мобильным роботом
 - погрешность сохранения расстояния между пользователем и мобильным роботом во время передвижения

Помимо вышеперечисленных, оцениваются следующие вспомогательные параметры отработки «служебных» алгоритмов, используемых различными модулями КФП:

- погрешность системы распознавания речи
- погрешность системы распознавания лиц
- время установления соединения между элементами КФП
- корректность отработки протоколов передачи данных между элементами КФП
- задержка передачи данных между элементами КФП
- корректность отработки взаимодействия КФП с клиентскими устройствами пользователя.

В последние годы мультимедийные приложения на мобильных устройствах пользуются большой популярностью. Благодаря быстрому развитию технологий беспроводной связи пропускная способность мобильных соединений становится достаточной для поддержания потокового видео в реальном времени, что позволяет интегрировать пользовательские мобильные устройства в КФП. Однако из-за их ограниченных возможностей потоковая передача видео высокой четкости (HD) по-прежнему представляет собой непростую задачу. Зачастую мобильные клиенты имеют ограниченный размер экрана, относительно медленное сетевое соединение, а также ограниченное время работы от аккумулятора, поэтому при разработке приложений ВКС для мобильных устройств следует учитывать следующие аспекты [10]:

1) Изменчивость сетевой среды: в отличие от проводной сети, мобильное соединение зачастую неустойчиво, а пропускная способность может значительно изменяться. Чтобы гарантировать бесперебойную и высококачественную доставку видео для мобильного клиента, система должна динамически регулировать скорость передачи видеопотока, отправленного клиенту.

2) Разнообразие возможностей мобильных устройств: мобильные устройства очень разнородны с точки зрения аппаратных и программных решений для обработки видео. У высокопроизводительных смартфонов часто есть аппаратный ускоритель для декодирования видео, тогда как большинство мобильных телефонов не могут поддерживать аппаратное декодирование. Для повышения производительности видеобработки система должна отправлять видео в подходящем разрешении в соответствии с возможностями клиентов.

При разработке КФП следует учитывать эти особенности функционирования мобильных клиентских устройств, чтобы оптимизировать обработку мультимедийных данных соответствующим образом. Как правило, средства ВКС ориентированы на захват клиентским устройством аудиовидеоданных только одного пользователя, и чаще всего, сеанс связи организован между двумя устройствами. Однако при проведении сеансов связи между несколькими пользователями такой подход зачастую недостаточен, и требуется многоканальная обработка аудиовизуальных сигналов на стороне каждого клиента [11].

Предложенная в работе схема соединения пользователей приложения ВКС, используемая в разработанном программном обеспечении, позволила добиться значений времени отклика соединения, представленных на рис. 20. Измерения отклика выполнялись с интервалом в 500 мс. В среднем, отклик для двух пользователей, находящихся в одном здании, составил 5.81 мс. Передача осуществлялась по Wi-Fi внутри локальной сети.



Рис. 20. Значения времени отклика пирингового соединения, установленного посредством разработанного программного обеспечения

Возникновение пиков (выбросов) на графике, наблюдаемое для измерений под номерами 40-76, 252, 357, может быть связано с особенностями порядка исполнения программы интерпретатором кода. Кроме того, поскольку передача осуществлялась по WiFi, увеличение задержки сигнала могло произойти вследствие возникновения помех в окружающей среде.

Далее рассмотрим первую демонстрационную версию системы распределения задач по многомодальному обслуживанию пользователей между мобильными роботами и другими обеспечивающими устройствами киберфизического пространства реализованную на основе многомодальной информационно-навигационной облачной системы («МИНОС»). Некоторые сценарии ее функционирования представлены на рис. 21.

Система взаимодействует с пользователями через интерфейсы и точки входа (меню, экраны, QR-коды), которые предлагают пользователю сервисы в зависимости от контекста и типа пользователя. Новые пользователи должны получить учётную запись в системе и предоставить свои биометрические данные. Гости института могут запросить общую информацию об организации, информацию о расположении помещений, вызвать сотрудника с помощью сервиса видеоконференцсвязи. В этих случаях сценарий взаимодействия можно представить в виде графа. Например, рис. 22 иллюстрирует сценарий идентификации пользователя: субсервисы (вершины графа), составляющие сервис идентификации, и генерируемые события (рёбра графа). Объектно-ориентированная модель сервиса идентификации показана на рис. 23.

На рис. 23 для каждого метода идентификации (Method) каждый пользователь (Guest, Employee, Cooperator) может иметь отдельный профиль (Profile), содержащий идентификатор, пароль, биометрические или другие аутентификационные данные, позволяющие отличить его от других пользователей. Пользователи могут быть идентифицированы в различных зонах (Zone), где присутствует необходимое оборудование (турникеты, камеры, двери с электронными замками). Зоны идентификации могут быть внешними, расположенными на границе организации, и внутренними на её территории. Они распределены по территориям, т.е. этажам, отделам, и т.п. Факты идентификации (Attempt) хранятся в базе данных.

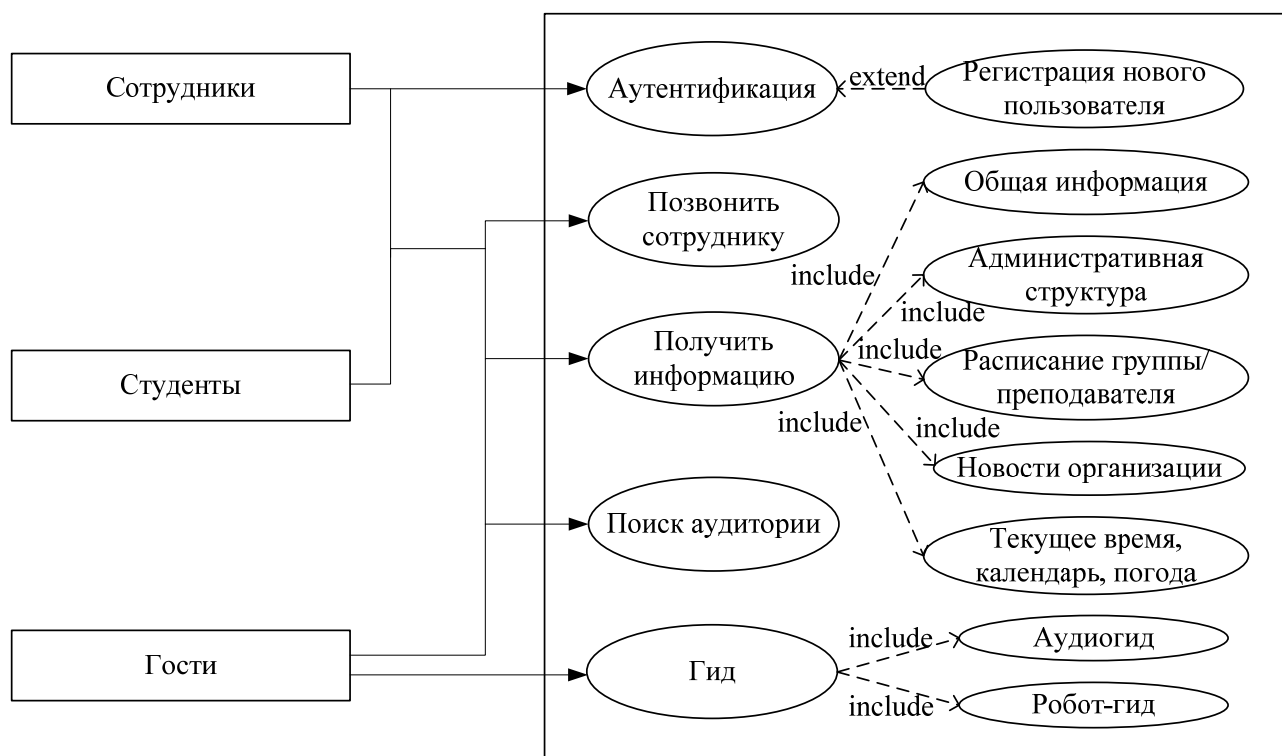


Рис. 21. Use Case-диаграмма сценариев взаимодействия

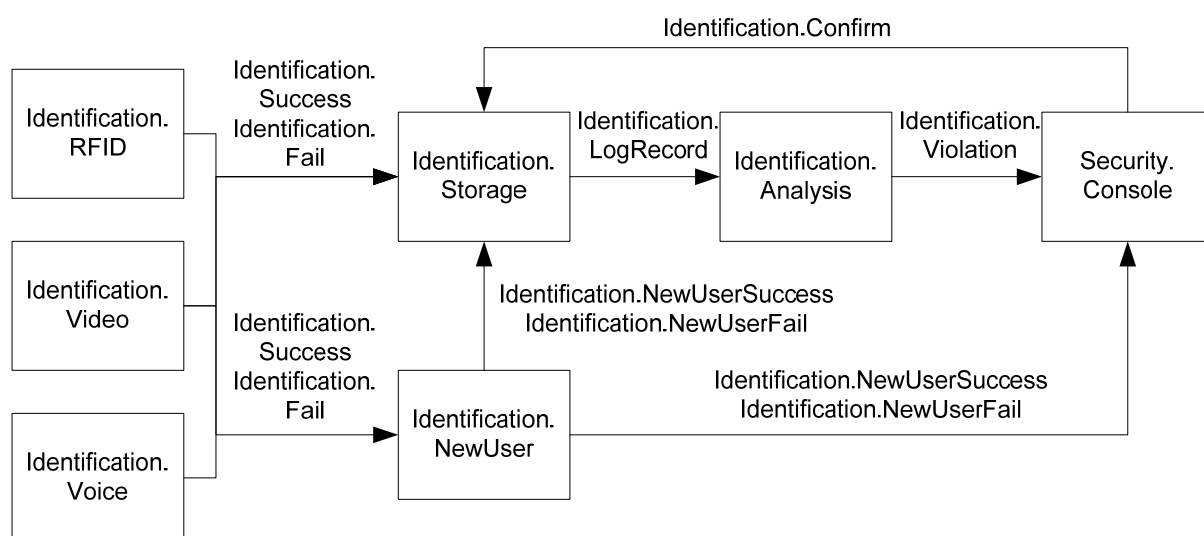


Рис. 21. Сценарий идентификации

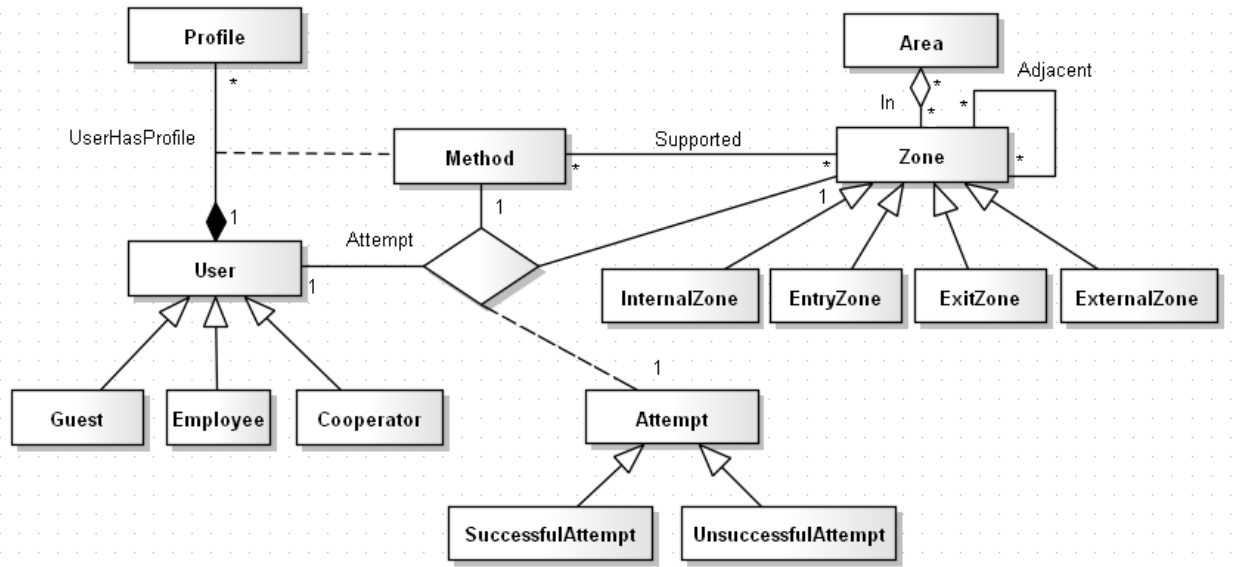


Рис. 23. Диаграмма классов для сервиса идентификации

Экземпляры классов могут быть представлены как сущности

$$e = \langle k, A \rangle, \quad (1)$$

где k – первичный ключ сущности, $A = \langle a_1, a_2, \dots, a_n \rangle$ - набор её атрибутов. Сущности с одним и тем же ключом рассматриваются как эквивалентные:

$$Eq(e_1, e_2): k_1 = k_2. \quad (2)$$

Пусть следующие выражения означают, что заданная сущность e принадлежит C и имеет атрибут a , равный v :

$$Is(e, C): A \in R_C, \quad (3)$$

$$Has(e, a, v): a = v, \quad (4)$$

где R_C - отношение, содержащее все объекты класса C . Также для любого отношения R используются обозначения:

$$R(e_1, e_2, \dots, e_n): \langle k_1, k_2, \dots, k_n \rangle \in R, \quad (5)$$

$$R_{i1, i2, \dots, im}(e_{i1}, e_{i2}, \dots, e_{im}): Ee_{im+1} Ee_{im+2} \dots Ee_{in} R(e_1, e_2, \dots, e_n). \quad (6)$$

Этот набор обозначений позволяет строить высказывания о существовании или несуществовании объектов и их отношений в терминах логики предикатов первого порядка. Временные и причинно-следственные отношения могут быть учтены с помощью темпоральной логики действий (TLA). Применение темпоральных операторов позволяет формализовать сценарии. Например, таблица 5 описывает некоторые ситуации нарушения безопасности.

Для любой формулы $S_i(x_1, x_2, \dots, x_n)$ можно построить высказывания о наличии (отсутствии) ситуации i :

$$Ex_1 Ex_2 \dots Ex_n S_i(x_1, x_2, \dots, x_n), \quad (7)$$

$$Ax_1 Ax_2 \dots Ax_n \sim S_i(x_1, x_2, \dots, x_n). \quad (8)$$

Высказывания, относящиеся к более сложным ситуациям (например, сочетанию S_i и S_j) могут быть построены аналогичным образом. Применение известных методов верификации [13-15] позволяет определять истинность высказываний. Это позволяет

управлять реестром событий и предпринимать необходимые для защиты действия, а также автоматизировать этот процесс. Проверка совместности формул из табл. 5 и формул, описывающих текущее состояние системы, позволяет также предсказывать возникновение ситуаций.

Таблица 5. Сценарии нарушения

Ситуация	Формула
Одновременная идентификация одного человека в разных местах	$S_1(u, z_1, z_2) = User(u) \wedge Zone(z_1) \wedge Zone(z_2) \wedge SuccessfulAttempt(u, z_1) \wedge SuccessfulAttempt(u, z_2) \wedge \sim Eq(z_1, z_2) \wedge \sim Adjacent(z_1, z_2)$
Разные результаты идентификации одного человека с использованием разных методов (например, если сотрудник использует чужой пропуск)	$S_2(z, u_1, u_2) = User(u_1) \wedge User(u_2) \wedge Zone(z) \wedge SuccessfulAttempt(u_1, z) \wedge SuccessfulAttempt(u_2, z)$
Идентификация посторонних на внутренней территории организации (предполагается, что новые посетители регистрируются в системе при первом визите)	$S_3(u, z) = InternalZone(z) \wedge SuccessfulAttempt(u, z) \wedge \exists p (Profile(p) \rightarrow \sim UserHasProfile(u, p))$
Идентификация человека на внутренней территории организации при отсутствии метки входа	$S_4(u) = (\exists z_0 Entry(z_0) \rightarrow \sim SuccessfulAttempt(u, z_0)) \wedge \forall z (InternalZone(z) \wedge SuccessfulAttempt(u, z))$
Двойной вход	$S_5(u) = \exists z_1 \exists z_2 User(u) \wedge EntryZone(z_1) \wedge EntryZone(z_2) \wedge (SuccessfulAttempt(u, z_1) \wedge \exists z_3 ExitZone(z_3) \rightarrow \sim (SuccessfulAttempt(u, z_3)) \wedge SuccessfulAttempt(u, z_2))$

Стремление разработчиков создавать кроссплатформенные, легко сопровождаемые, встраиваемые приложения нашло своё проявление, в частности, в широком использовании Web-технологий. Web-приложения, которые изначально представляли собой статический HTML-код (гипертекст), в настоящее время оснащены множеством мощных инструментов (HTML5, EcmaScript 6, различные библиотеки для клиентского и серверного выполнения задач, визуализации, управления событиями, связи с другими веб-ресурсами) и используются для решения множества сложных прикладных задач – редактирование электронных документов (сервис Google Docs), создание 3D-карт и моделей (Web GL, BabylonJS), обеспечение видеоконференцсвязи [16]. Веб-интерфейсы широко применяются для управления сложными вычислительными комплексами, облачными сервисами. На основе веб-технологий разработаны стандарты представления данных (XML, JSON).

При этом веб-приложения отличаются низкими требованиями к аппаратному и программному обеспечению, возможностью централизованного обновления, высоким потенциалом в области создания распределённых систем, ориентированностью на сетевые технологии. Это делает веб-технологии перспективным инструментом создания интеллектуальных пространств, но для интеграции корпоративных информационных ресурсов необходимы динамические модели сценариев поведения сервисов, реализация которых позволяет достигать целей интеллектуального пространства (например, надлежащее информирование сотрудников или обеспечение безопасности предприятия), а

также интерфейсы, протоколы и архитектуры программного обеспечения, позволяющие устанавливать связи между сервисами.

Для построения интеллектуального пространства как распределённого многокомпонентного приложения можно использовать паттерн проектирования «модель – представление – контроллер», где в качестве модели выступает внутреннее состояние пространства, представлением являются компоненты, обеспечивающие связь пространства с внешним миром (устройства ввода и вывода, многомодальные интерфейсы, движущиеся части), а контроллер выполняет согласование состояний модели и представления. Модель включает логику приложения, которая направлена на достижение целей как интеллектуального пространства в целом, так и его отдельных сервисов (идентификации, информирования, и т.д.). Вид состоит из исходных программных и аппаратных компонентов и включает интерфейс, позволяющий им обмениваться информацией с контроллером.

Таким образом, приложение можно представить в виде следующего набора уровней (табл. 6) Коммуникации выполняются между уровнями, смежными по вертикали, с помощью генерации событий и обмена JSON-структурами с информацией о них.

Таблица 6. Структура приложения

Модель	Прикладной уровень – отвечает за решение задач интеллектуального пространства в целом.
	Сервисный уровень – отвечает за функционирование отдельных сервисов интеллектуального пространства.
Контроллер	Сигнальный уровень – отвечает за передачу сигналов (данных, событий) между сервисами и интерфейсами.
Представление	Интерфейсный уровень – предоставляет компонентам физического уровня интерфейс для коммуникации с другими компонентами интеллектуального пространства.
	Физический уровень – исходные компоненты, включаемые в интеллектуальное пространство.

Корпоративная многомодальная информационно-навигационная облачная система (МИНОС) строится на основе сервис-ориентированной архитектуры (COA) и состоит из слабосвязанных сервисов, взаимодействующих по унифицированным протоколам с использованием веб-сокетов. В сравнении с подходом REST применение веб-сокетов позволяет на протяжении длительного времени поддерживать соединение в активном состоянии и не выполнять его открытие и закрытие при каждой передаче данных. Это особенно важно при использовании безопасного соединения (в протоколах TLS, HTTPS и др.), когда создание логического канала требует генерации ключей. Эта операция длится дольше, чем передача среднестатистического JSON-сообщения, и использование традиционного REST-подхода приведёт к существенным накладным расходам и снижению времени отклика сервисов. Кроме того, веб-сокеты позволяют контролировать статус соединений и оперативно определять сервисы, по каким-либо причинам отключившиеся от системы.

Сервисы разделяются на две категории: системные, которые обеспечивают выполнение задач, необходимых для функционирования системы в целом, и прикладные, реализующие цели системы. Перечень сервисов и их краткое описание приведены в табл. 7.

Таблица 7. Сервисы МИНОС

Сервис	Описание
Системные сервисы	
Сервис доступа к данным	предоставляет доступ к информации, которая может быть использована разными сервисами (например, сведения об учётных записях, пользователях и т.д.), и контролирует правомерность доступа
Репозиторий сервисов	предоставляет информацию о зарегистрированных в системе сервисах, их статусе и методам доступа к сервисам
Интерфейс администратора	позволяет управлять системой в режиме онлайн
Управление аккаунтами	позволяет запрашивать и изменять список учётных записей, их ролей и привилегий
Классная доска	позволяет сервисам публиковать информацию о своих событиях для обработки другими сервисами
Сервер сценариев	анализирует состояние системы с использованием формализованного описания сценариев и генерирует события на основе обнаруженных и/или предсказанных ситуаций
Прикладные сервисы	
Сервисный хаб	позволяет пользователям получать доступ к сервисам с использованием мобильных приложений
Сервис корпоративного телевидения	транслирует медиаконтент (видеоролики, объявления, информационные сообщения) на клиентские устройства (стационарные экраны)
Сервис корпоративного портала	позволяет использовать информацию от сервисов МИНОС на сайте организации
Сервис коммуникации	позволяет пользователям управлять информационным пространством с помощью мультимедийных технологий (речь, жесты)
Сервис навигации	предоставляет клиентским устройствам информацию об их физическом расположении в корпоративном пространстве, о расположении подразделений организации и маршрутах их достижения
Сервис видеоконференцсвязи	организует видеоконференцсвязь между клиентскими устройствами в корпоративном пространстве
Сервис идентификации	позволяет идентифицировать посетителей различными методами (с помощью электронных ключей, биометрических признаков)

Построение интегрированного интеллектуального пространства (smart space) предполагает, что активность всех прикладных сервисов тесно связана. Вместе с тем, сервис-ориентированная архитектура предполагает слабую связь компонентов. Более того, сервисы независимы и не обязаны знать о существовании других сервисов и учитывать их наличие, иначе архитектура системы потеряла бы гибкость. В этом случае целесообразно использование классной доски, где автономные сервисы публикуют информацию о событиях, которая может быть обработана другими сервисами.

Каждый сервис может публиковать сообщения о событиях в формате JSON, имеющие следующую структуру:

```
{
  id : <идентификатор события>,
  class : <класс(ы) события>, // определяют тип сообщения
                                   // для получателя
  source : { // информация об объекте, сгенерировавшем сообщение
    service : <ID сервиса>,
    node : <ID узла сети>,
    signature : <электронно-цифровая подпись сервиса>
  },
  target : <класс сервиса>, // сервисы-получатели (опционально)
  timestamp : <время генерации>,
  expires : <срок действия>,
  status : <состояние>, // непрочитано, в обработке, завершено
  contents : <содержимое сообщения>,
}
```

Для этого сервис выполняет запрос к сервису классной доски. Запрос содержит JSON-объект с параметрами сообщения. В ответ сервис получает объект, содержащий код успеха и идентификатор события, позволяющий отслеживать его статус в дальнейшем.

Разработанная демонстрационная версия системы позволяет координировать корпоративные сервисы для достижения целей на уровне организации, таких как обмен информацией, мониторинг безопасности, и т.п. на основе применения аппарата темпоральной логики действий над множеством объектами и отношений в модели данных корпоративного интеллектуального пространства.

В настоящее время система находится на стадии активной разработки. Разработано клиентское и серверное программное обеспечение для управления сервисами корпоративного телевидения и видеоконференцсвязи. Сервис корпоративного телевидения использует стандартные сетевые технологии, не привязана к конкретным программно-аппаратным платформам. Он соответствует критериям расширяемости и переносимости и может использоваться в качестве компонента киберфизической среды в различных организациях.

Дальнейшее развитие сервисов включает расширение возможностей идентификации пользователей с использованием методов [17], основанных на биометрии, RFID, использовании мобильных устройств. В частности, интеграция систем идентификации и корпоративного телевидения [18] позволит повысить интерактивность и персонализацию последнего. Предполагается реализовать управление выводом информации с помощью жестов и речи [19-20]. Другим направлением разработки является предоставление сервиса навигации, который даёт посетителю возможность определять его/её расположение с помощью сканирования QR-кодов. Веб-приложение позволит посетителю ориентироваться на территории организации и получать необходимую информацию с помощью мобильных устройств.

Другой задачей проекта на втором этапе являлась разработка конструктивных элементов торса и манипуляторов антропоморфного робота Антарес для решения задач сопровождения пользователей в киберфизическом пространстве.

Базовая модель торса была разработана с целью определения максимально приемлемых массогабаритных свойств данного узла и дальнейшей пошаговой оптимизацией для получения легкого и прочного каркасного модуля, подходящего для размещения в его внутреннем пространстве основного вычислительного модуля и системы электропитания [21-23].

Перед проведением начальной оптимизации базовой модели торса была рассмотрена простейшая сборка торса, расчетный вес которой составлял 1,95 кг. Модель состояла из простых прямых пластин, соединенных между собой болтовыми соединениями с целью получения замкнутой конструкции для равномерного распределения нагрузок. Сервоприводы в такой сборке в верхней части торса установлены для прямого соединения с манипуляторами и модулем головы, в нижней части сервоприводы тазового механизма установлены с дополнительной зубчатой передачей, с целью снижения нагрузки на редуктор сервопривода. При проведении симуляции на первичную модель торса воздействовали вертикальная нагрузка в 5 Н и момент вращения вдоль сагиттальной плоскости, равный 0,6 Н·м. Максимальное внутреннее напряжение составляет $5,66 \cdot 10^5$ Н/м², и располагается в плечевой пластине в месте крепления двигателя. Во время приведения манипулятора в вертикальное положение вдоль сагиттальной плоскости торс способен выдержать предельную нагрузку 65 Н вертикальной силы и 8,64 Н·м в виде момента вращения на валу двигателя.

Затем к сборке торса прикладывается вертикальная нагрузка в 5 Н и момент вращения вдоль фронтальной плоскости, равный 0,6 Н·м. Максимальное внутреннее напряжение составило $4,5 \cdot 10^5$ Н/м², и приходится на втулку тазового механизма и зубчатой передачи. Во время приведения манипулятора в вертикальное положение вдоль фронтальной плоскости торс способен выдержать предельную нагрузку 48 Н вертикальной силы и 8,32 Н·м в виде момента вращения на валу двигателя. После проведения исследования и симуляций с первичной сборкой двигателя были определены важные точки в разрабатываемом модуле торса для антропоморфного робота Антарес, которым было уделено внимание при разработке новой концепции несущей конструкции для туловища робота, были усилены плечи и тазовый механизм, каждый сервопривод во второй версии прототипа был закреплен винтовыми соединениями с пластинами торса для исключения смещения осей выходных валов во время движения или возможной деформации пластин, а также общего укрепления конструкции. Плечевые сервоприводы были вынесены в отдельные надстройки на общем корпусе сервопривода. Это было сделано для упрощения общего вида модуля и исключения из конструкции слабых мест. После предварительной оптимизации в симуляции была исследована вторая модель сборки торса робота. На плечевые сервоприводы предварительно оптимизированной сборки торса прикладывается вертикальная нагрузка по 5 Н (рис. 24), заменяющая вес руки с предполагаемым полезным грузом.

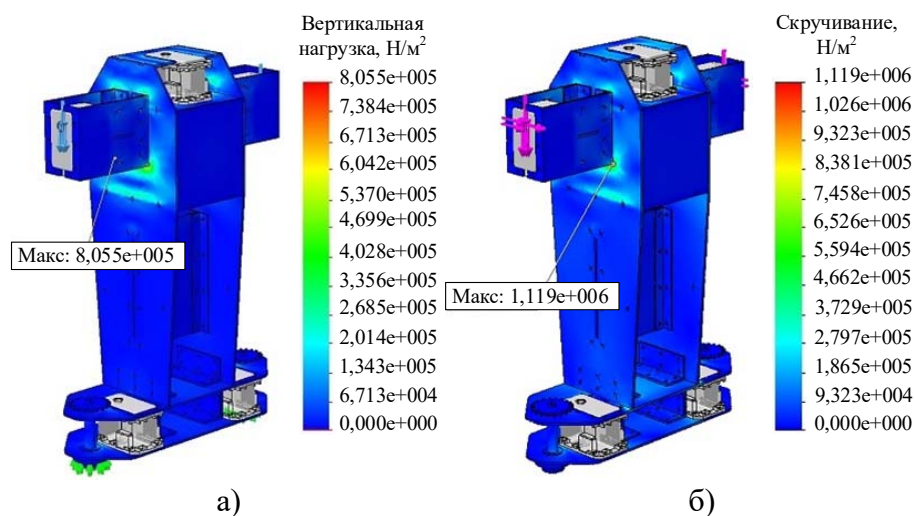


Рис. 24. Результаты симуляции нагрузок на оптимизированную сборку торса и тазового механизма: а) вертикальная нагрузка; б) скручивание

Максимальное внутреннее напряжение (рис. 24а) при этом составляет $8,05 \cdot 10^5$ Н/м² и приходится на пластину плеча, на которую устанавливается сервопривод. Для проверки сборки на максимальную внешнюю нагрузку была проведена симуляция нагрузки, соответствующей расчетному весу руки с постепенным прибавлением дополнительного веса предполагаемой полезной нагрузки, после чего было установлено, что торс способен выдержать 135 Н веса каждой руки. При дальнейшем проведении симуляции для проверки на скручивание к каркасу торса прикладывается момент вращения вдоль центральной оси модуля, равный 1,3 Н·м. Кроме момента вращения дополнительное воздействие оказывает вес руки, равный 5 Н. Максимальное внутреннее напряжение (рис. 24б) составляет $1,12 \cdot 10^6$ Н/м² и приходится на пластину надстройки плеча в месте винтового крепления к основной боковой пластине торса. Как и в первой симуляции, в третьем исследовании модели на сборку торса прикладывалась вертикальная нагрузка в 5 Н и момент вращения вдоль сагиттальной плоскости, равный 0,63 Н·м. Максимальное внутреннее напряжение (рис. 25а) составило $1,1 \cdot 10^6$ Н/м² и приходит на плечевую пластину в месте винтового крепления к боковой пластине торса. Во время поднятия руки в сагиттальной плоскости торс способен выдержать предельную нагрузку 75 Н вертикальной силы и 9,53 Н·м момента вращения на валу двигателя. Четвертое исследование первично оптимизированного каркаса торса показывает результат приложения вертикальной нагрузки в 5 Н, и момента вращения вдоль фронтальной плоскости, равный 0,63 Н·м.

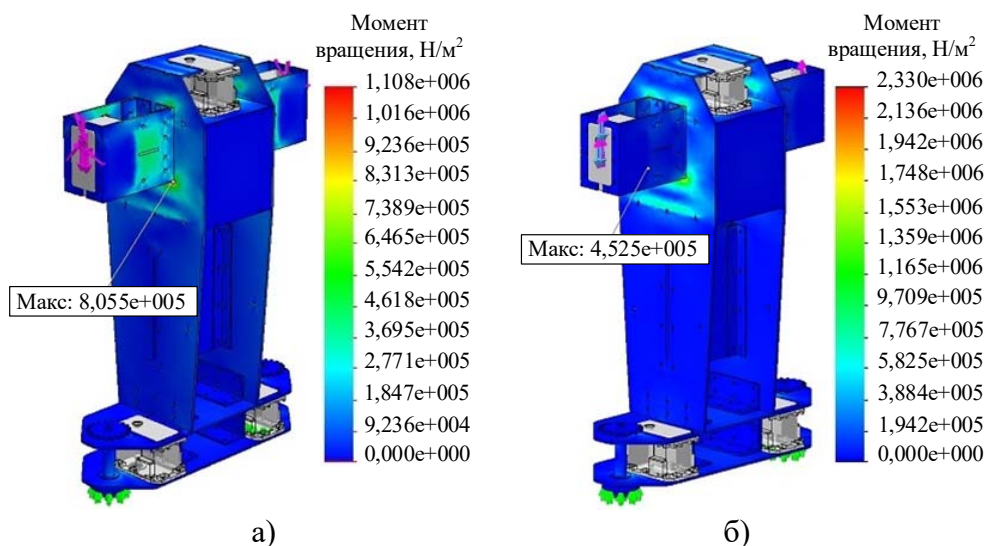


Рис. 25. Результаты симуляции нагрузок на оптимизированную сборку торса и тазового механизма: а) момент вращения в сагиттальной плоскости; б) момент вращения во фронтальной плоскости

Максимальное внутреннее напряжение (рис. 25б) составляет $2,33 \cdot 10^6$ Н/м² и приходится на плечевую пластину в месте винтового крепления к боковой пластине торса. Во время приведения манипулятора в вертикальное положение вдоль фронтальной плоскости торс способен выдержать предельную нагрузку 53 Н вертикальной силы и 9,48 Н·м в виде момента вращения на валу двигателя. После изменения не только формы деталей, но и создания принципиально новой конструкции каркаса, удалось снизить вес конструкции, при этом прочность и надежность конструкции увеличились. Прочность возросла в 2 раза, т.е. модель стала крепче, чем была ранее. Вес сборки после первоначальной оптимизации составляет 1,24 кг. При приложении различных внешних сил к оптимизированной сборке торса внутреннее напряжение будет приближено к точке предела текучести. По результатам проведения моделирования и симуляции сборок каркасных элементов торса робота Антарес была построена оптимальная модель торса (рис. 26), включающая в себя основной несущий каркас, предназначенный для установки модулей головы, манипуляторов и педикуляторов. Моделирование показало, что для

сборки и дальнейшего обслуживания робота оптимально будет изменить концепт построения каркаса торса и отделить плечевые пластины и сервоприводы от каркасных элементов торса как модуля (рис. 26б, в), так как при оптимизации полученных деталей они более не нужны как дополнительные ребра жесткости и наиболее оптимально будет включить их в состав модулей манипуляторов робота.

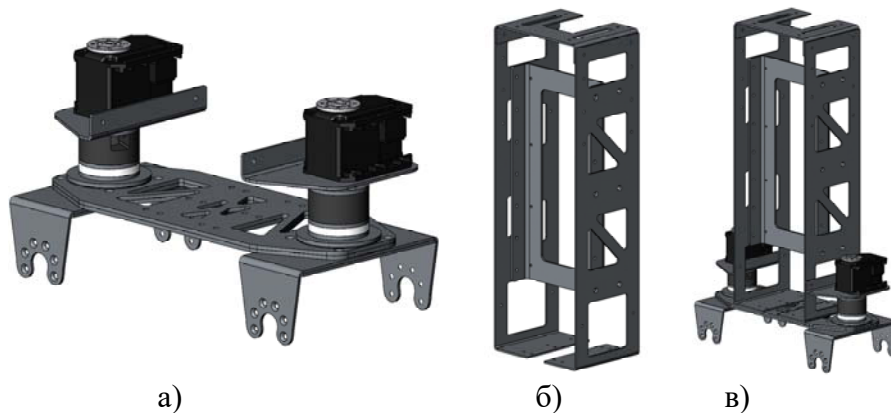


Рис. 26. Оптимизированный модуль торса и тазового механизма: а) таз; б) базовый короб торса; в) модуль торса в сборе

После первичного моделирования и оптимизации последующих версий конструкций торса было определено необходимое внутреннее пространство для размещения вычислительных модулей, контроллеров, проводки и аккумуляторной системы питания. Начиная со второй версии модуля моделирование проходило с учетом этого необходимого пространства. Ниже (рис. 27) можно увидеть сравнение версий каркасного модуля торса (рис. 27а, б, в).

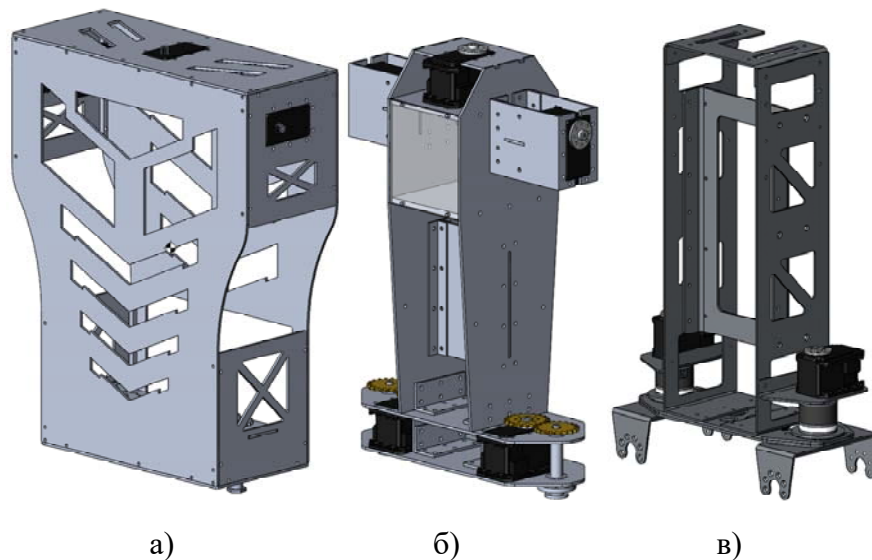


Рис. 27. Сравнение результатов моделирования каркасных элементов торса: а) первичная модель; б) модель с новой концепцией; в) оптимизированная модель

В оптимизированной модели были учтены результаты моделирования внешних нагрузок и потенциального износа исполнительного механизма таза, слабые элементы конструкции были переработаны, либо исключены из несущего каркаса робота. Вес оптимизированной конструкции торса, включая тазовый механизм, составил 700 г, из которых 150 г занимают сервоприводы. В дальнейшем при разработке каркаса робота будет большее внимание уделяться разработке механизма модульного интерфейса подключения конечностей и сенсорного модуля робота.

Отдельным модуль является разработанный манипулятор (рис. 28) для рук антропоморфного робота Антарес.

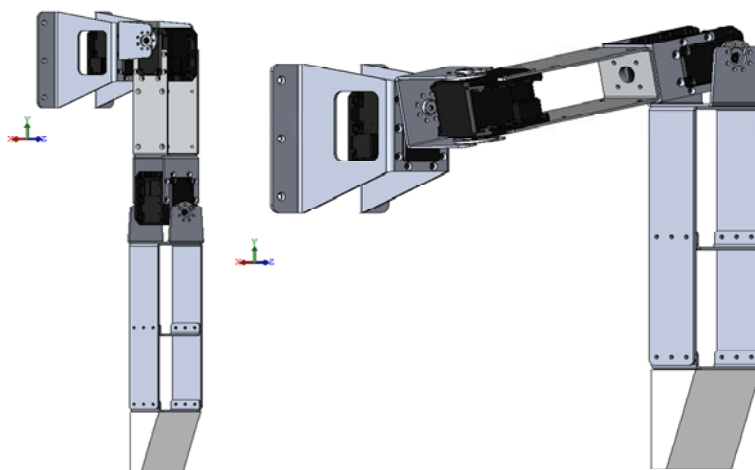


Рис. 28. Опорный манипулятор антропоморфного робота Антарес

В базовом варианте манипуляторы разработаны как опорные механизмы, необходимые для поддержания равновесия и восстановления робота в положение стоя. Опорные стойки могут быть заменены на захватные устройства. Конструкция манипуляторов рассчитана на размещение субконтроллера, позволяющего подключить дополнительный аккумулятор с целью снижения электрической нагрузки на основную цепь питания, а также упростить управление сервоприводами манипулятора, подключив сервоприводы к субконтроллеру.

В итоге к концу 2017 года был разработан каркас антропоморфного робота Антарес, включающий в себя модули торса, головы, манипуляторов и педипуляторов (рис. 29). Робот оснащен 22 сервоприводами, необходимыми для передвижения, контроллером и двумя вспомогательными субконтроллерами, камерой. Переработано двухмоторное колено для упрощения кинематической схемы робота и оптимизации системы управления. В дальнейшем для оучувствления робота будет разрабатываться система датчиков, включающая в себя датчики стопы для обратной связи о равномерном контакте с поверхностью, по которой движется робот, гироскоп и акселерометр для определения положения робота в пространстве. Будет прорабатываться модульная концепция робота для облегчения сборки, замены деталей и реконфигурирования робота. Разрабатывается захватный механизм для манипуляторов робота.

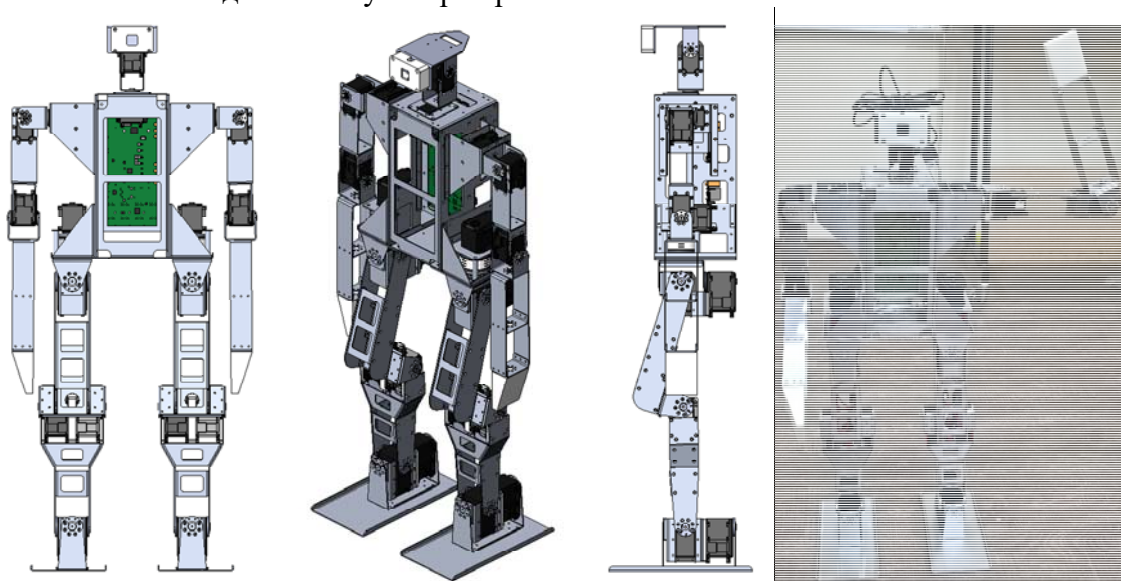


Рис. 29. Антропоморфный робот Антарес

В целом концепция интеллектуального пространства, как помещения, оборудованного в соответствии с его назначением («умный дом», «умная теплица», «интеллектуальный зал совещаний»), позволяющего автоматизировать рутинные процессы или вовсе исключить необходимость управления человеком-оператором, ставит перед исследователями и разработчиками целый ряд задач на стыке разных областей, таких как человеко-машинное взаимодействие (Human-Machine Interaction), слияние данных (Data Fusion), распознавание образов (Pattern Recognition), и т.д. Основными проблемами проектирования интеллектуальных пространств остаются недостаточность данных, описывающих контекст взаимодействия и сложность его формализации, а также конфликт двух парадигм (ограниченность встроенных ресурсов на робототехнических платформах / негибкость стационарных вычислительных центров). Развитие беспроводных сетевых и облачных технологий позволяет частично устранить данную проблему. Кроме того, интеллектуализация пространства организации/предприятия/частного дома (и т.п.) требует индивидуального решения в каждом конкретном случае и с трудом масштабируется в виду различной конфигурации помещений и их технических характеристик. Вместе с тем, индивидуальный подход в каждом конкретном случае позволяет добиться более тонкой подстройки под предъявляемые к интеллектуальному пространству требования, а также обеспечить широкий спектр сервисов, подобранных в соответствии с непосредственными запросами пользователей.

Литература

1. Vatamaniuk I., Levonevskiy D., Saveliev A., Denisov A. Scenarios of Multimodal Information Navigation Services for Users in Cyberphysical Environment // Lecture Notes in Computer Science. 2016. vol. 9811. pp. 588-595.
2. Ronzhin A., Vatamaniuk I., Železný M. Algorithms for Acceleration of Image Processing at Automatic Registration of Meeting Participants // Lecture Notes in Computer Science. 2014. vol. 8773. pp. 89-96.
3. Park, J.J. and Yang, L.T. and Lee, C. Future Information Technology: 6th International Conference on Future Information Technology, FutureTech 2011, Crete, Greece, June 28-30, 2011. Proceedings. — Springer, 2011. — P. 89-90. — 526 p
4. Рутледж, Д. Энциклопедия практической электроники: Пер. с англ. — М.: ДМК Пресс, 2002. — С.249-250.
5. Levonevskiy D. K., Vatamaniuk I. V., Saveliev A. I. MINOS Multimodal Information and Navigation Cloud System for the Corporate Cyber-Physical Smart Space, Programnaya Ingeneria [Software Engineering], 2017, vol. 8, no. 3, pp. 120–128 (In Russian).
6. Vatamaniuk I., Levonevskiy D., Saveliev A., Denisov A. Scenarios of Multimodal Information Navigation Services for Users in Cyberphysical Environment. In proc. SPECOM-2016, Budapest, Hungary. Springer. 2016. vol. 9811 LNAI, pp. 588–595.
7. Saveliev A. I., Somenkov N. V. Architecture of Client Part of Peer-To-Peer Videoconferencing Application. Matematicheskie metody v tehnikе i tehnologijah-MMTT [Proceedings of the International Conference Mathematical Methods in Technique and Technologies - MMTT], 2016, no. 9, pp. 176–179 (In Russian).
8. Saveliev A. I. Algorithms of Data Processing in Supervised Accounts of a Videoconferencing System. Informatsionno-upravliaiushchie sistemy, 2016, no. 3(82), pp. 15–23 (In Russian).
9. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 2000. 403 p.
10. Cheng R., Wu W., Lou Y., Chen Y. A cloud-based transcoding framework for real-time mobile video conferencing system. Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on. IEEE, 2014. pp. 236–245.

11. Saveliev A. I., Ronzhin A. L., Vatamaniuk I. V. Architecture of Data Exchange with Minimal Client-Server Interaction at Multipoint Video Conferencing. *Lecture Notes in Computer Science*. 2014. vol. 8638 LNCS. pp. 164–174.
12. Demri, S., Gastin, P.: Specification and verification using temporal logics. In: *Modern applications of automata theory*. Iisc Research Monographs, vol. 2. World Scientific, Singapore (2009)
13. Fainekos, G.E., Girard, A., Pappas, G.J.: Temporal logic verification using simulation. In: Asarin, E., Bouyer, P. (eds.) *FORMATS 2006*. LNCS, vol. 4202, pp. 171–186. Springer, Heidelberg (2006). doi:10.1007/11867340_13
14. Liu, C., Orgun, M.A.: Verification of reactive systems using temporal logic with clocks. *Theoret. Comput. Sci.* 220(2), 277–408 (1999)
15. Saveliev, A.I., Ronzhin, A.L.: Algorithms and software tools for distribution of multimedia data streams in client server videoconferencing applications. *Pattern Recognition and Image Analysis* 25(3), 517–525 (2015)
16. Kanev, K. et al.: A human computer interactions framework for biometric user identification. In: *JJAP Conference Proceedings*. The Japan Society of Applied Physics, vol. 4 (2015)
17. Ivanov D., Sokolov B., Dashevsky V. RFID-based Adaptive feedbacks between supply chain scheduling and execution control. In: *Proceedings of 18th IFAC World Congress, Milano (Italy), 28 August–2 September 2011 [CD]*. International Federation of Automatic Control, Prague, pp. 435–440 (2011)
18. Ronzhin, A.L., Karpov, A.A., Lobanov, B.M., Tsirulnik, L.I., Jokisch, O.: Phonetic-morphological mapping of speech corpora for recognition and synthesis of Russian speech. *Informacionno-upravljajushhie sistemy* 25(6), 24–34 (2006). (in Russian)
19. Kashevnik, A.M., Ponomarev, A.V., Savosin, S.V.: Hybrid systems control based on smart space technology. *SPIIRAS Proc.* 35(4), 212–226 (2014)
20. Vatamaniuk, I.V., Ronzhin, A.L.: Application of digital images blur estimation methods for audiovisual monitoring. *Informacionno-upravljajushhie sistemy* 4, 16–23 (2014). (in Russian)
21. Павлюк Н.А., Будков В.Ю., Бизин М.М., Ронжин А.Л. Разработка конструкции узла ноги антропоморфного робота Антарес на основе двухмоторного колена // *Известия ЮФУ. Технические науки*. — 2016. — № 1 (174). — С. 227–239.
22. Pavluk, N., Denisov, A., Kodyakov, A., Ronzhin, A. Mechanical Engineering of Leg Joints of Anthropomorphic Robot // *International Conference on Power, Energy and Mechanical Engineering 2016, MATEC Web of Conferences*. 2016. — 77. — art. no. 04006.
23. Павлюк Н.А., Ронжин А.Л. Конструктивные решения нижних конечностей для антропоморфного робота Антарес // *Экстремальная робототехника*. — 2016. — Т.1. — № 1. — С. 422–427.