

УДК 004.8

М.Д. Миронов, А.М. Кашевник

Адаптивный интерфейс на базе смартфона для системы управления шестиколёсным мобильным роботом

Описывается разработка адаптивного интерфейса для управления шестиколёсным мобильным роботом, основанного на концепции интеллектуальных пространств. Рассмотрены современные исследования в области интерфейсов управления беспилотными наземными транспортными средствами. Представлены архитектура системы управления роботом и архитектура Android-приложения, обладающего интерфейсом управления, способным адаптироваться к возможностям робота, потребностям пользователя и контексту выполняемой задачи. Приведён пример управления шестиколёсным роботом путём распределения задач управления в группе пользователей.

Ключевые слова: беспилотное наземное транспортное средство, интерфейс управления, интеллектуальное пространство, Android-приложение.

doi: 10.21293/1818-0442-2017-20-1-110-118

Благодаря технологичным материалам и высокой производительности микрочипов, современные роботы при малых габаритах обладают большими вычислительными мощностями, чем их предшественники. За счёт этого они могут выполнять всё более сложные задачи без прямого участия пользователя. В результате значительное развитие получает направление беспилотных [1] и киберфизических систем [2].

Сфера применения таких роботов может распространяться от антитеррористических операций до исследования других планет. Главным фактором достижения этой универсальности является то, что такие системы могут потенциально функционировать в условиях, в которых человек не может находиться. Чаще всего в качестве беспилотного наземного транспортного средства (unmanned ground vehicle – UGV) используются роботы различных размеров и конструкций. Снижение стоимости производства и стремительное развитие робототехники, как прикладной науки позволяют использовать роботов не только в научной или военной, но и в повседневной деятельности. В данном проекте для моделирования различных сценариев работы используется шестиколёсный мобильный робот, обладающий большим количеством различных приводов и соответственно высокой степенью подвижности всех элементов. За счёт этого робот обладает возможностью преодоления не только простых препятствий разной высоты с плоской вершиной, но и сложных составных разноразмерных препятствий, таких как лестница.

Помимо возможности доступа в различные локации, роботы должны иметь возможность управляться на расстоянии, т.к. на сегодняшний день не все процессы при функционировании робота возможно автоматизировать. Возросшее число возможностей и функций, выполняемых роботами, превратило устройства управления в сложные и громоздкие наборы рычагов и кнопок, управление которыми требует дополнительных тренировок и навыков.

На смену пультам управления могут прийти более универсальные устройства – смартфоны. В связи с развитием микроэлектроники смартфоны стали одними из самых востребованных устройств в современном мире. Такая популярность приводит к тому, что ежегодно они продаются миллиардными партиями во всём мире [3], что позволяет говорить о том, что они давно стали заменой мобильным телефонам и имеются почти у каждого человека. Растущий следом рынок мобильных технологий превращает смартфон в универсального помощника для пользователя при взаимодействии с различными физическими устройствами и сервисами в сети Интернет [4, 5]. На основе этого можно сделать вывод, что современные смартфоны, как ничто иное, подходят на роль универсального и многофункционального устройства удалённого управления. Также стоит отметить, что растущая с каждым годом вычислительная мощность позволяет им соперничать в производительности даже с персональными компьютерами середины двухтысячных годов [6], обладая при этом высокой автономностью. Речь идёт не только о мощности центрального процессора, но и о графической составляющей. Современные смартфоны могут обладать Full HD (1920×1080) разрешением экрана и диагональю до 6 дюймов. Эти характеристики позволяют отобразить на дисплее большое количество различных элементов управления роботом. Тем самым использование смартфона в качестве управляющего устройства позволяет решить проблему больших габаритов и веса устройства управления, но не проблему избыточности элементов управления.

Чаще всего для решения определённого набора задач пользователю требуется ограниченный набор инструментов и функций. Исходя из этого, становится очевидно, что чем удачнее будет организованное расположение элементов управления этими функциями, тем выше будет скорость выполнения задач оператором. В случае проектирования интерфейса управления мы имеем возможность только

компоновать элементы рядом друг с другом, располагая их в зависимости от частоты использования, по группам операций или по иному принципу. Но в конечном итоге мы всё равно получим устройство управления, весь функционал которого располагается на консоли. Работы по исследованию процессов адаптации интерфейсов управления ведутся уже значительное время [7, 8].

Процесс адаптации физических устройств управления грозит или уменьшением функционала управляемого устройства, или усложнением вызова его функций путём нажатия комбинаций клавиш. Тем самым можно сказать, что «отцифровка», перенос устройства управления на экран смартфона в виде приложения, увеличивает возможности при проектировании интерфейса управления. К примеру, появляется возможность скрывать большую часть функционала и отображать его только по требованию пользователя. Положительный эффект от такого шага очевиден, ведь при размещении на экране смартфона слишком большого количества рычагов и кнопок пользователю будет не только сложно не путать все элементы управления, но и визуальным образом интерфейс будет отпугивать избыточным разнообразием. Управление роботом будет удобным, только если смартфон будет обладать интуитивно понятным, простым и лаконичным интерфейсом управления и не будет содержать элементов, в которых нет необходимости в данный момент времени (контекстно-ориентированное отображение элементов управления). Это ведёт не только к увеличению продуктивности в решении типовых задач, но и снижает риски ошибочных нажатий.

Как упоминалось выше, большое количество элементов управления связано с ростом функциональных возможностей мобильных роботов. Также в зависимости от условий использования и выполняемых задач конфигурация роботов может сильно варьироваться, а в некоторых случаях в качестве UGV могут выступать несколько роботов, связанных друг с другом. Тем самым количество механизмов управления различными модулями возрастает в разы. С другой стороны, используемые механизмы сильно зависят от контекста, в котором находится робот. Ситуационная ориентированность функционала позволяет упростить процессы управления многокомпонентными роботами путём объединения нескольких функциональных элементов в один интерфейс доступа или скрытия ненужных в данный момент элементов. Так, к примеру, объединив доступ к нескольким приводам робота, можно добиться заметного упрощения решения задачи преодоления им различных препятствий: начиная от преодоления отдельно стоящего препятствия высотой больше радиуса колеса робота и заканчивая подъёмом по лестнице.

В рамках данного проекта для системы управления многоприводным UGV был разработан адаптивный интерфейс для смартфона под управлением Android ОС.

Современные исследования в области мобильных интерфейсов управления

В наше время смартфоны обладают широким спектром встроенных датчиков и сенсоров. Акселерометр, гироскоп, GPS и другие датчики предоставляют возможности для реализации различных функций интерфейсов управления. Высокая производительность и многопоточность мобильных процессоров обеспечивают быстрое действие программ. Большое количество оперативной и физической памяти позволяет обрабатывать большое количество входящей информации. А растущая с каждым годом ёмкость аккумуляторных батарей увеличивает автономность работы устройства. Немалую роль в работе интерфейса управления играют высокое разрешение и размер дисплеев современных смартфонов.

Существует несколько основных подходов к реализации мобильных интерфейсов управления.

Самым популярным является применение стрелок направления [9]. Кнопки в виде стрелок, направленных вверх и вниз, отвечают за так называемые дроссельные команды (движение вперёд и назад), а иконки влево и вправо, в свою очередь, отвечают за рулевое управление объектом. Данный подход является наиболее простым в освоении для пользователей в силу своей интуитивной понятности и логичности. Но он также обладает и рядом недостатков. К примеру, при движении робота по сложной траектории осуществлять повороты и быстро маневрировать будет проблематично из-за необходимости частой смены нажимаемой кнопки. Иконки других функций помимо движения могут отображаться стрелками или другими условными иконками. В первом случае наличие нескольких похожих по смыслу функциональных клавиш может сбить пользователя с толку, а во втором требует дополнительного дизайна. Кроме этого, данный подход проявляет свою несовершенство в ситуациях, требующих одновременного использования большого количества функционала.

Другим вариантом реализации интерфейсов управления является использование визуального джойстика, что увеличивает количество возможных углов поворота колёс и тем самым увеличивает манёвренность объекта. Также такой подход не влияет на интуитивность управления, т.к. он используется в качестве регуляторов освещения, мощности или скорости вращения электродвигателей, для управления различными устройствами, машинами и механизмами вплоть до космических аппаратов [10]. Для управления направлением движения робота в данном подходе достаточно одного джойстика, что задействует только одну руку пользователя. Это оставляет больше свободы для одновременного манипулирования несколькими функциями, в отличие от предыдущего варианта.

Ещё одним из решений проблемы маневрирования является применение аналога рулевого колеса. В данном случае пользователь может управлять направлением поворота путём вращения иконки руле-

вого колеса против или по часовой стрелке. На практике данный подход имеет слишком избыточный набор возможных углов поворота, из-за чего вращение иконки до крайнего значения занимает много времени.

Использование таких элементов управления, как вертикальный и горизонтальный слайдеры, также является одним из подходов к разработке интерфейса управления. Изначально слайдеры находятся в среднем положении. Отклонение вертикального элемента служит для выполнения дроссельных команд, а горизонтального, в свою очередь, – для поворота. При отпускании бегунка значение слайдера возвращается к изначальному. Такой подход используется редко по причине его неинтуитивности.

Управление с помощью встроенного в смартфон акселерометра является одним из самых технологичных подходов [11]. В этом случае программа интерфейса управления фиксирует поворот смартфона в горизонтальной и вертикальной плоскости и даёт команду роботу на изменение направления или скорости соответственно. Наличие датчика гироскопа в устройстве позволяет использовать его для управления другими операциями манипулирования, к примеру поднятием той или иной части робота. Такой подход часто применяется в мобильных играх от первого или третьего лица (Angry Birds Go, Asphalt), но, как правило, в настройках всегда присутствует возможность переключиться на более классический интерфейс управления. Не в пользу данного подхода говорят слабая точность датчиков, необходимость их калибровки и неудобства управления таким интерфейсом в движении.

Использование голосового управления роботами, используемое в статье [12], также является одним из подходов к реализации интерфейса управления. С одной стороны, такой подход является наиболее удобным в ситуациях, когда у оператора заняты руки. Но с другой – требует сложного механизма распознавания и фильтрации. Кроме этого, при высоком уровне внешнего шума сильно ухудшается качество распознавания команд.

Авторы статьи [13] провели эксперимент, в котором пользователям предлагалось, управляя роботом, пройти трассу, используя 4 различные комбинации элементов управления. Комбинируя различные элементы управления, авторы провели 4 группы тестов с 20 пользователями. На основе среднего времени прохождения трассы, количества ошибочных действий и отзывов испытуемых, авторами был сделан вывод, что использование стрелок для рулевого управления и дроссельных команд является наиболее удобным и простым для пользователя, чем другие комбинации.

Большинство упомянутых подходов управления берут своё начало в видеоиграх от первого или третьего лица. При наличии установленной на роботе видекамеры и трансляции на экран смартфона пользователь управляет роботом по тому же принципу. В данном проекте видеосигнал не использует-

ся, поэтому для управления роботом в интерфейсе изображены его вид в профиль. Такой вариант отображения объекта управления позволяет реализовать гибридный подход к управлению: набор иконок-стрелок образует собой окружность наподобие джойстика, позволяющую управлять каждым элементом робота по отдельности. В зависимости от контекста и команд пользователя происходит частичное или полное скрытие функционала, что разгружает визуальное оформление интерфейса и акцентирует внимание пользователя на текущей задаче.

Постановка задачи

В данном проекте объектом управления является прототип шестиколёсного робота на мобильной робототехнической платформе МРП-100 (рис. 1) изготавливаемого в научно-техническом центре «РОКАД» [14].



Рис. 1. Шестиколёсный робот на мобильной робототехнической платформе МРП-100

UGV представляет собой двухсекционную платформу с колесно-шагающим движителем, предназначенную для создания на ее базе мобильных роботов и мобильных робототехнических комплексов для выполнения следующих задач:

- Визуальная, радиоэлектронная, аудио-, радиационная, химическая разведка местности, промышленных и жилых помещений, объектов транспорта при любой освещенности. Электронное картографирование результатов разведки. Передача данных разведки.
- Дистанционное исследование радиологической обстановки местности.
- Очистка и дезактивация территорий.
- Помощь в проведении антитеррористических операций: постановка радиоэлектронных помех, дымовых и специальных завес, доставка и применение спецсредств нелетального действия, скрытное проникновение на захваченные и охраняемые объекты, ведение отвлекающего огня, выявление огневых точек противника.
- Охрана объектов: патрулирование территории или периметра объекта, пресечение попыток проникновения на объект, нейтрализация нарушителей.
- Выполнение сложных технологических операций.
- Работа во взрыво- и пожароопасных средах [14].

Каждая ось платформы представляет собой отдельный блок с набором приводов. Центральный блок является связующим звеном, к которому с помощью трубок крепятся идентичные направляющий и замыкающий блоки. Эти блоки оснащены колёсными приводами, а также могут отдаляться от центрального блока на длину соединяющей трубки. Центральный блок имеет по приводу на каждое колесо оси и приводы, позволяющие вращать прикреплённые к ним соединяющие трубки. Так, с учётом всего набора приводов и креплений данный полноприводный робот может опускать, поднимать и вращать направляющие блоки (оси), а также увеличивать свою длину.

В рамках исследования была поставлена задача разработать адаптивный интерфейс на базе смартфона, с помощью которого пользователь мог бы решить следующую задачу. Используя приложения на мобильном телефоне, оператор посылает роботу команду движения вперёд. Сканируя пространство перед собой, робот двигается до тех пор, пока не определит впереди себя преграду или же не будет остановлен пользователем. При обнаружении преграды UGV остановится перед препятствием и подаст звуковой сигнал. Данное событие также будет отображено на пользовательском интерфейсе управления условным обозначением. После этого пользователь сможет либо вручную, самостоятельно управляя всеми действиями робота, преодолеть препятствие, либо просто дать роботу команду «преодолеть препятствие». В последнем случае робот в автоматическом режиме будет пытаться подобрать и выполнить набор команд для совершения данного действия. При возникновении проблем или по команде пользователя робот остановится и подаст звуковой сигнал. В таком случае пользователю следует в ручном режиме сообщить последовательность команд для выполнения сценария работы. При успешном преодолении препятствия UGV продолжает своё движение прямо, без остановки.

Данный сценарий работы имитирует управление роботом при преодолении поребриков, бордюров, камней и других подобных препятствий. В качестве моделей препятствий выступают предметы, высота которых больше половины диаметра колеса робота, т.к. на препятствия меньшей высоты подобный робот должен забираться без использования подъёмных шасси.

Сценарий управления роботом группой пользователей

Очень часто для управления сложными и многокомпонентными устройствами, такими как робот-сапёр или луноход, используют несколько пользователей [15]. В данном проекте также рассматривается подобный сценарий работы. Функциональные возможности робота можно условно разделить на 2 группы операций:

- Передвижение (дрессельные команды и команды поворота колёс).
- Манипуляции (команды подъёма и опускания блоков).

В результате этого можно разграничить функционал управления роботом на 3 группы пользователей: водитель (имеет доступ только к первой группе операций); манипулятор (использует только вторую группу); а также эксперт (при невозможности преодоления препятствия в автоматическом режиме, робот через интерфейс сигнализирует именно этой группе. В таком случае пользователь переключает управление в ручной режим и помогает роботу выполнить задачу, комбинируя группы операций). Группы управления распределяются между пользователями по следующему принципу: при подключении к роботу первого оператора он автоматически получает доступ ко всем возможным операциям, тем самым выполняя все обозначенные роли. При подключении второго оператора первый пользователь получает возможность выбрать одну из двух групп управления. Второй автоматически получает права другой группы. К примеру, ранее подключившийся пользователь выбирает роль водителя, а второй – роль манипулятора. По аналогии при подключении третьего пользователя первые два по очереди получают возможность выбрать, в какой из трёх ролей выступать. Играющий роль эксперта ожидает возникновения необходимости ручного преодоления препятствия. В данном случае на основе последовательности команд робот создаст шаблон поведения. В дальнейшем при определении препятствия со схожими габаритами, следуя шаблону, робот будет пытаться преодолеть его в автоматическом режиме. В случае если робот попадает в такую ситуацию, находясь под управлением лишь двумя операторами, водитель получает доступ к операциям манипуляции и выступает в роли эксперта.

Архитектура системы управления роботом

Для взаимодействия между различными элементами системы используется платформа Smart-M3, реализующая концепцию интеллектуальных пространств (ИП) [16]. Основными составляющими платформы являются база знаний, информация в которой хранится в виде RDF-графа, и SIB (Semantic Information Broker) брокер. Последний представляет собой интерфейс взаимодействия различных устройств с базой знаний. В контексте ИП-устройства, взаимодействующие с платформой, называются информационными агентами, КР (Knowledge Processor). Агенты, подключённые к брокеру SIB, получают возможность вставлять, извлекать, редактировать или удалять информацию в базе знаний Smart-M3. Безопасность и целостность данных обеспечиваются специализированным прикладным протоколом передачи данных – SSAP (Smart Space Access Protocol), являющимся XML-ориентированным. Кроме перечисленных выше функций, КР также могут использовать механизм подписок, благодаря которому они будут получать уведомления обо всех изменениях интересующей их информации.

При её обновлении брокер SIB отправляет всем подписчикам RDF-тройки со старыми и новыми значениями. Использование платформы Smart-M3 в данном проекте позволяет решить проблему взаимо-

действия устройств, различных по своей программной и аппаратной архитектуре.

На рис. 2 изображена общая архитектура системы управления роботом. Всю систему можно условно разделить на физические компоненты и управляющие ими информационные компоненты. К

первой группе относятся смартфон(ы), используемые для управления одним или более UGV, и сами роботы. Каждому физическому компоненту в системе соответствует как минимум один информационный – агент КР, через который осуществляет взаимодействие с ИП и другими компонентами системы.

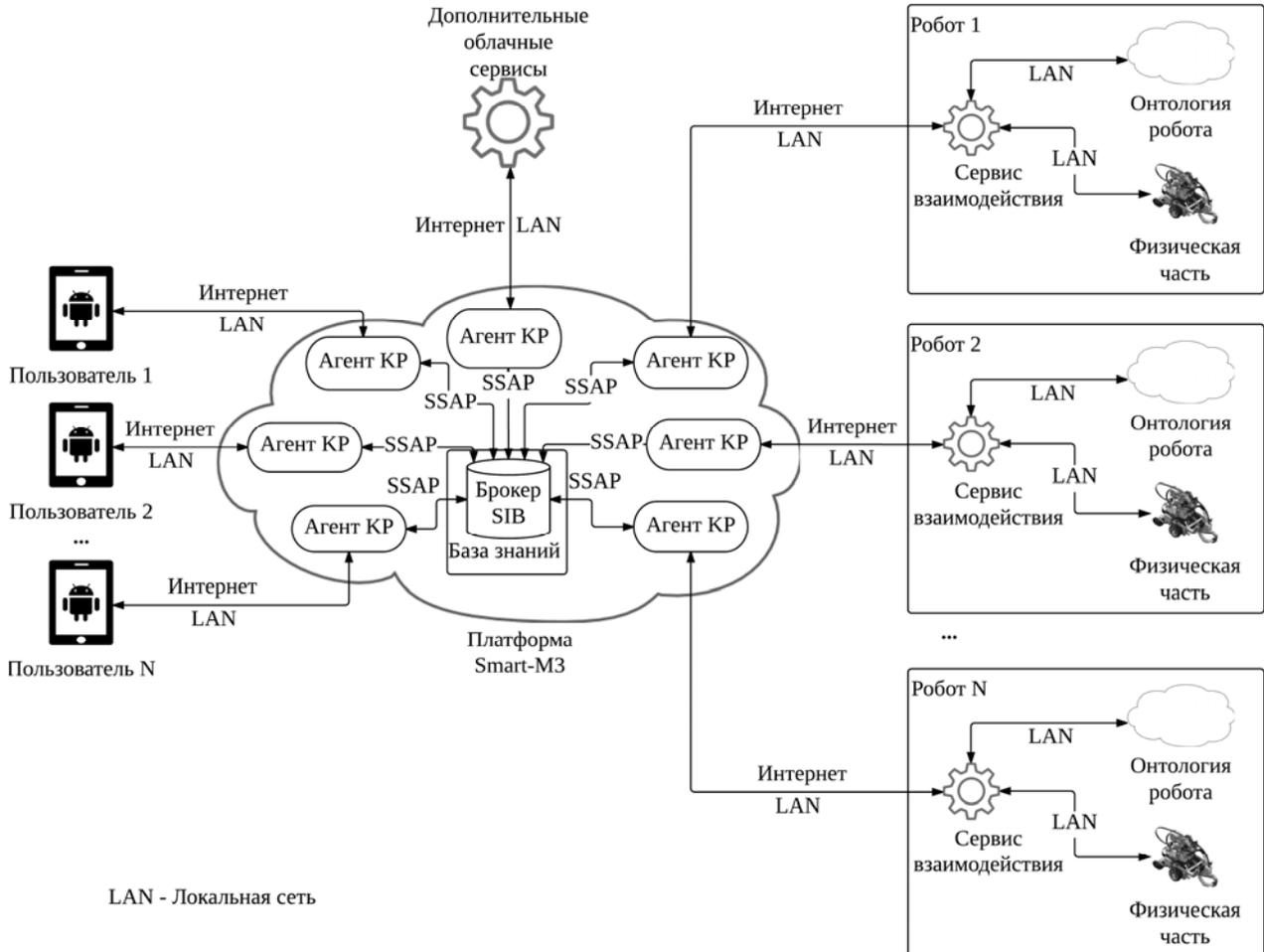


Рис. 2. Общая архитектура системы управления роботом

Кроме этого, к информационным компонентам относится сама платформа Smart-M3 и сервисы взаимодействия, получающие данные от ИП и передающие их напрямую роботам в рамках беспроводной локальной сети. Процесс управления происходит следующим образом: пользователь или группа пользователей с помощью интерфейса мобильного приложения через Интернет или локальную сеть подключается к ИП. Все дальнейшие пользовательские команды передаются в виде RDF-троек и сохраняются в базе знаний. Агенты КР-роботов, в свою очередь, с помощью механизма подписок передают команды сервису взаимодействия определенного робота или целой группе. Независимость агентов КР от системной архитектуры устройства предоставляет большой выбор путей дальнейшего развития системы и увеличения количества взаимодействующих устройств.

В качестве дополнительных сервисов могут выступать, к примеру, стационарная видеочкамера, оп-

ределяющая размеры и габариты препятствий вокруг робота, или дополнительный модуль UGV – квадрокоптер или другое беспилотное летательное устройство с камерой.

Адаптивный интерфейс

Описанные выше подходы к реализации интерфейсов управления чаще всего используются при разработке приложений для конкретного робота с фиксированным набором компонентов и модулей, что влечёт за собой массу трудностей при добавлении или же отключении различных модулей. К примеру, при подключении дополнительного подъёмного механизма к блоку робота придётся не только изменять схему работы последнего, но и реализовывать функционал управления новым объектом, что также влечёт за собой обновление или даже переустановку всего приложения интерфейса управления. Создание интерфейса управления для различных модификаций многоосевых роботов было одной из основных задач данного проекта. В зависимости от

поставленной задачи роботы могут обладать различными компонентами и возможностями. При первом подключении к ИП интерфейс подписывается на информацию о роботе и его наборе компонентов. Затем, при изменении этих данных, пользователю динамически отображаются возможные варианты управления. Так, ось с отсутствующим приводом интуитивно отображается без стрелок направления движения (вперёд, назад), а неподнимающийся блок без стрелки – вверх. Также интерфейс адаптируется под потребности пользователя. Отсутствие нагромождения стрелок, иллюстрирующих все возможности данной конфигурации робота, не путают пользователя и обеспечивают лаконичность отображаемого интерфейса. Если человек не хочет управлять всеми приводами по отдельности и приводит робота в движение по частям, интерфейс предоставляет ему возможность сдвинуть робота с места, потянув головной блок в нужном направлении. При этом команда двигаться передаётся автоматически всем имеющимся осевым приводам.

Обладая ультразвуковым датчиком, робот, встретив препятствие и остановившись, звуком уведомляет об этом пользователя. Кроме звукового сопровождения, пользователь может видеть возникшее препятствие условным изображением на интерфейсе. Тем самым управляющая программа и, в свою очередь, робот оповещают человека о необходимости взять ситуацию под свой контроль и самому принимать решение. Такая адаптация к контексту, ситуации, в которой находится UGV, позволяет избежать возможных поломок оборудования или нанесения урона окружающим предметам. Тем самым можно сказать, что разработанный интерфейс способен изменяться, адаптироваться в зависимости от возможностей робота, задач человека, контекста (текущей ситуации, в которой находится робот) (рис. 3).



Рис. 3. Структура адаптивного интерфейса

Архитектура мобильного приложения

В качестве платформы для реализации интерфейса управления была выбрана ОС Android как самая популярная и открытая мобильная платформа современности. Для разработки мобильного приложения, реализующего интерфейс управления, использовалась среда Android Studio, предоставляющая удобный функционал работы с кодом, эмулятор

для тестирования проектов, а также встроенный набор тем для дизайна приложения. Основой графического интерфейса всех Android-приложений является Activity. Класс Activity представляет визуальную активность приложения и определяет действия, которые может производить пользователь.

Данный проект реализует два класса, наследующих класс AppCompatActivity, – это IPInsert и Control. На рис. 4 изображены основные классы приложения, их основные методы и связи между ними. IPInsert реализует экран ввода IP-адреса Smart-M3 сервера. Методы класса IPAddressValidator производят валидацию введенных данных на соответствие маске IPv4. При корректности полученного IP-адреса приложение проверяет наличие интернет-соединения на устройстве и запускает процесс подключения к ИП. Компонент ListOfLastUse отображает набор последних использованных адресов. Этот компонент реализован с использованием встроенного в ОС метода хранения данных – preferences. Благодаря этому последние IP-адреса, к которым удалось успешно подключиться, будут отображаться не только после перезапуска приложения, но и после его обновления. Класс SmartM3 реализует все возможные методы обмена данными с ИП, тем самым предоставляя всем классам приложения удобный интерфейс работы с ним. Кроме этого, в данном классе производится обработка всех ошибок, возникающих при работе с Smart-M3 сервером, и производится периодическая проверка работоспособности соединения. По причине того, что политики безопасности ОС Android не позволяют производить передачу данных и другие операции по сети в главном потоке приложения, данный класс работает в отдельном потоке.

После успешного подключения в серверу ИП пользователь переходит к основному экрану приложения, реализованному в классе Control. При первом подключении, на основе полученных из ИП данных метод paintBlock отрисовывает на экране Image классы. Эти сущности хранят в себе векторные изображения в формате базового элемента-контейнера ImageView, а также координаты его крайних точек. На каждый такой элемент установлен clickListener, реализующий свой набор операций. Иными словами, каждый элемент экрана является интерактивным. На основе заложенных шаблонов модулей робота формируется массив блоков текущего робота. Каждый блок имеет уникальное имя-индекс и набор функций.

Элементы управления функциями реализуются классом ImageAgrow, обладающим методами скрытия или отображения компонента. Тем самым элементы управления отображаются в зависимости от того, нажал ли пользователь на конкретный блок или нет. Подписавшись на информацию о конфигурации робота, в дальнейшем интерфейс в реальном времени будет отображать функциональные изменения UGV. Как упоминалось выше, ОС не позволяет вызывать главный поток из процесса, работающего с сетью, поэтому вся информация о подписках или

возникающих ошибках передаётся с помощью системных сообщений, обрабатываемых экземпляром системного класса Handler. На этом же экране отображаются кнопка остановки движения робота и кнопка записи команд при преодолении препятствий в ручном режиме.

Для оповещения пользователя о возможных неполадках в работе или неверном вводе информации Activity классами используется класс ProgressDialog, реализующий методы отображения toast-оповещений и загрузочных диалогов.

Апробация интерфейса управления

Тестирование мобильного приложения с адаптивным интерфейсом проводилось в следующих

условиях. Во всех предыдущих работах, связанных с Smart-M3 платформой, сервер разворачивался на персональном компьютере, подключённом к Wi-Fi роутеру. Благодаря установке Smart-M3-сервера непосредственно на ОС Wi-Fi-роутера и тонкой настройке его характеристик в данном проекте удалось избавиться от одного клиента в сети и тем самым увеличить скорость передачи данных между клиентами. В качестве тестового варианта робота на базе Lego EV3 была сконструирована модель шестиколёсного робота МРП-100 (рис. 5), обладающего меньшим количеством функций, но способного решать приближённые к реальности задачи, описанные ниже.

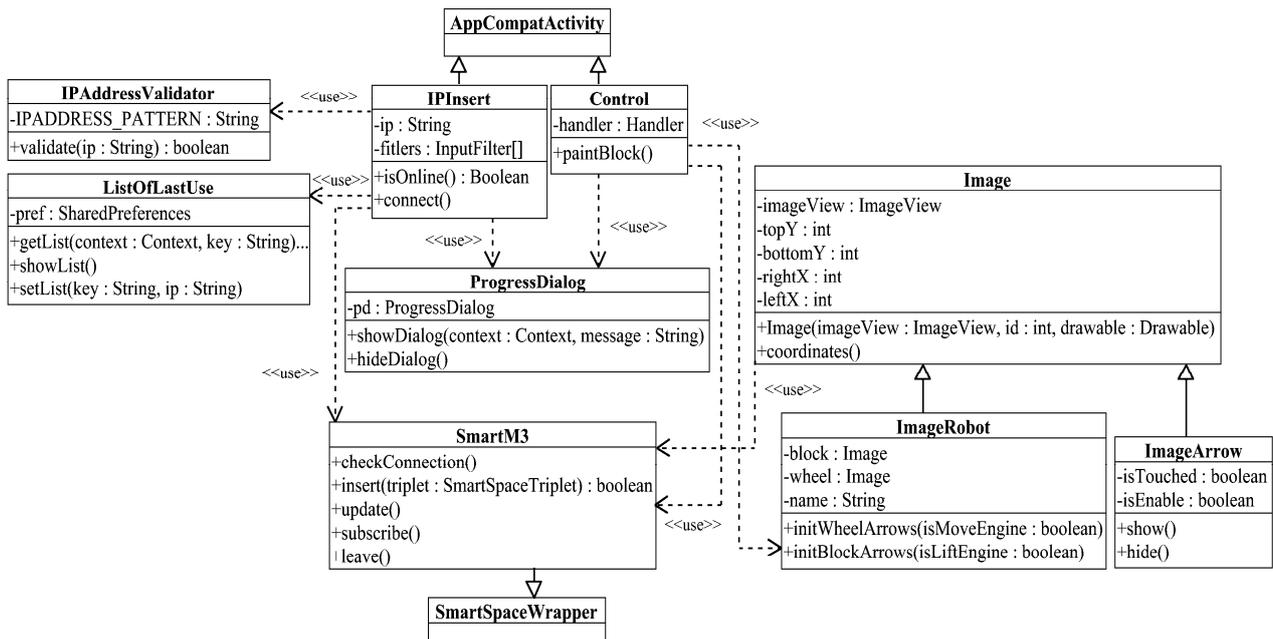


Рис. 4. Диаграмма классов приложения



Рис. 5. Опытный образец робота МРП-100, сконструированный на основе набора Mindstorms Lego EV3

На центральный блок Lego-робота установлена дополнительная пара колёс меньшего диаметра без привода для устойчивости конструкции, что связано с тем, что пластмассовые детали сильно уступают металлическим в крепости и массе. Помимо этого, ограниченное количество портов для подключения

моторов в управляющем блоке не позволило реализовать механизм подъёма задней оси и функции увеличения длины робота. С другой стороны, наличие ультразвукового датчика позволяет четко определять препятствия и тем самым точнее рассчитывать начало выполнения того или иного манёвра. Приложение

тестировалось на устройствах Samsung, Asus и ZTE. Диагональ экрана смартфонов варьировалась от 4,3 до 5,5 дюйма. Благодаря использованию векторных иконок интерфейс приложения отрисовывался корректно и чётко в независимости от длины диагонали и разрешения дисплея мобильного устройства. На рис. 6 изображены два снимка основного экрана приложения. Для пользователя представленный экран является интерфейсом управления роботом, и именно здесь располагаются все элементы подачи команд. На верхней части изображения представлен момент, когда пользователь нажал на иконку ведущего (правого) блока.

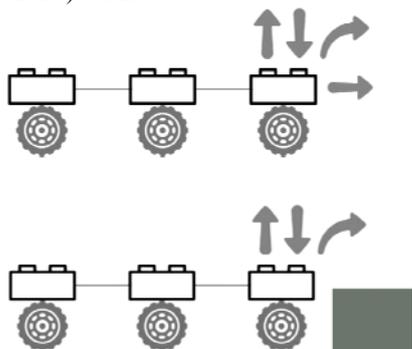


Рис. 6. Вид экрана управления приложения

Набор элементов управления остальными блоками демонстрирует ситуацию, когда робот остановился перед препятствием и ждёт команды от пользователя, который в свою очередь также нажал на иконку ведущего блока. Приведённые примеры отображают состояние интерфейса управления мобильным роботом при аналогичных действиях пользователя, но с различным ситуационным контекстом. Оба снимка подтверждают способность интерфейса адаптироваться в зависимости от возможностей робота, задач пользователя, контекстной ситуации.

Заключение

В ходе данной работы было рассмотрено применение концепции интеллектуальных пространств для разработки адаптивного интерфейса управления шестиколёсным мобильным роботом и описана архитектура системы управления роботом. Более детально в рамках текущего проекта описывается мобильное приложение, созданное для апробации адаптивного интерфейса.

В статье были рассмотрены современные исследования в области интерфейсов управления. Тестирование работоспособности и удобства использования интерфейса происходило на специально сконструированном прототипе многофункционального робота, также описанном в данной статье. За счёт расширения функционала робота-прототипа, а также увеличения количества сервисов системы управления, в качестве дальнейшей работы планируется реализация новых возможностей интерфейса управления. На текущий момент прорабатывается вариант добавления внешней веб-камеры, которая, анализируя изображения с использованием каскада Хаара, будет более точно сообщать габариты препятствий

перед роботом. Кроме этого, анализируется вариант адаптации интерфейса управления для сценария взаимодействия группы пользователей с группой роботов, также описанного в данной работе.

Представленные результаты исследований являются частью проектов № 16-29-04349 и 16-07-00462, финансируемых Российским фондом фундаментальных исследований, программой президиума РАН I.31П «Актуальные проблемы робототехники», а также бюджетной темой № 0073-2014-0005.

Литература

1. Линдсей В. Рынок беспилотных систем растёт и меняется // Мир компьютерной автоматизации. – 2009. – № 1. – С. 44–49.
2. Мещеряков Р.В. Активная карданная передача как элемент киберфизической системы / Р.В. Мещеряков, О.Ю. Осипов, Ю.М. Осипов // Изв. вузов. Приборостроение. – 2016. – Т. 59, № 11. – С. 934–938.
3. Scarsella A. Worldwide Smartphone Forecast Update, 2015–2019: December 2015 / A. Scarsella, W. Stofega. – IDC, 2015 [Электронный ресурс]. – Режим доступа: <https://www.idc.com/getdoc.jsp?containerId=US40734415>, свободный (дата обращения: 07.02.2017).
4. Borgia E. The Internet of Things vision: Key features, applications and open issues // Computer Communications. – 2014. – Vol. 54. – PP. 1–31.
5. Бородин А.В. Интернет вещей – следующий этап цифровой революции // Наука и образование. – 2014. – № 2(5). – С. 178–182.
6. The Great Equalizer 3: How Fast is Your Smartphone/Tablet in PC GPU Terms [Электронный ресурс]. – Режим доступа: <http://www.anandtech.com/show/6877/the-great-equalizer-part-3>, свободный (дата обращения: 07.02.2017).
7. Верлань А.Ф. Об организации адаптивного пользовательского интерфейса в автоматизированных системах / А.Ф. Верлань, М.Ф. Сопель, Ю.О. Фуртат // Изв. Южного фед. ун-та. Технические науки. – 2014. – №1(150). – С. 100–110.
8. Комагоров В.П. Система адаптивного управления разработкой «интеллектуального» месторождения / В.П. Комагоров, О.Б. Фофанов, Э.М. Мехтиев, А.О. Савельев, А.А. Алексеев // Доклады ТУСУРа. – 2014. – № 4(34). – С. 171–175.
9. Luo Y. Smartphone-Controlled Robot Snake for Urban Search and Rescue / Y. Luo, J. Liu, Y. Gao, Z. Lu // Lecture Notes in Computer Science. – 2014. – Vol. 8917. – PP. 352–363.
10. Seo Y. Remote Data Acquisition and Touch-Based Control of a Mobile Robot Using a Smart Phone / Y. Seo, H. Jung, C. Lee, T. Yang // Communication and Networking. – 2012. – Vol. 265. – PP. 219–226.
11. Chen G. Robot Remote Control Using Bluetooth and a Smartphone Augmented System / G. Chen, S.A. King, M. Scherger // Informatics in Control, Automation and Robotics. – 2012. – Vol. 133. – PP. 453–460.
12. Егунов В.А. Система управления мобильными роботами с помощью речевых команд / В.А. Егунов, О.А. Авед // Изв. Волгогр. гос. техн. ун-та. – 2014. – №6(133). – С. 137–138.
13. Kenny C.W. Interactive methods of Tele-operating a single unmanned ground vehicle on a small screen interface / C.W. Kenny, C.H. Foo, Y.S. Lee // Proceedings of the 6th international conference on Human-robot interaction. – 2011. – PP. 121–122.

14. Научно-технический центр «РОКАД» [Электронный ресурс]. – Режим доступа: <http://rocad.ru/>, свободный (дата обращения: 07.02.2017).

15. Басов О.О. Концептуальная и формальная модели синтеза киберфизических систем и интеллектуальных пространств / О.О. Басов, Б.В. Соколов, Р.М. Юсупов, А.Л. Ронжин // Изв. вузов. Приборостроение. – 2016. – № 11(59). – С. 897–905.

16. Кашевник А.М. Разработка интеллектуальных систем на базе платформы Smart-M3: учеб. пособие для студентов математических и технических специальностей вузов / А.М. Кашевник, Д.Ж. Корзун, С.И. Баландин. – Петрозаводск: Изд-во ПетрГУ, 2013. – 51 с.

Миронов Михаил Дмитриевич

Программист Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН); студент каф. информационных систем Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (Университет ИТМО)
Тел.: +7 (812) 328-80-71
Эл. почта: mironoff.togo@gmail.com

Кашевник Алексей Михайлович

Канд. техн. наук, с.н.с. СПИИРАН, доцент каф. информационных систем Университета ИТМО
Тел.: +7 (812) 328-80-71
Эл. почта: alexey@iias.spb.su

Mironov M.D., Kashevnik A.M.

Adaptive remote control interface based on Smartphone application for six-wheeled mobile robot

The article describes a process of developing adaptive control interface for remote control of six-wheeled mobile robot based on the concept of smart spaces. Design approaches of the control interfaces for unmanned ground vehicles are overviewed. The architecture of the robot control system and Android application have been presented. User interface of the mobile application is adapting to the robot's capabilities, user needs and context of the task. An example of six-wheeled mobile robot control process with delegating responsibilities among the group members has been described in paper.

Keywords: unmanned ground vehicle, remote control interface, smart space, Android application.