

25 ноября 2019 года в Военном инновационном технополисе «ЭРА» (г. Анапа) проведена конференция на тему «Состояние и перспективы развития современной науки по направлению «Технологии энергообеспечения. Аппараты и машины жизнеобеспечения».

Цель конференции – Организация обмена информацией о новых научно-технических разработках, объединения ведущих научных школ, поиска партнёров в области разработки перспективных технологии энергетики, аппаратов и машин систем жизнеобеспечения в Вооружённых Силах Российской Федерации.

Задачи конференции:

- создание условий для эффективного взаимодействия органов военного управления с предприятиями ОПК на площадке ВИТ «ЭРА»;
- обмен опытом в области инновационных решений по направлению энергетики, технологий, аппаратов и машин систем жизнеобеспечения;
- обмен мнениями и уточнение приоритетных направлений развития химмотологии топлив, масел, смазок и специальных жидкостей в интересах Вооружённых Силах Российской Федерации, разработки композитных конструкционных материалов, аппаратов и машин систем жизнеобеспечения объектов военной инфраструктуры, источников электропитания и систем распределения энергетических ресурсов;
- уточнение вопросов формирования совместных научных коллективов для эффективного проведения исследований в областях деятельности лаборатории.

В конференции приняли участие как доктора и кандидаты наук, докторанты и адъюнкты (аспиранты) образовательных учреждений и научных организаций, так и операторы научных рот.

Основные результаты работы участников конференции отражены в сборнике статей. Содержание статей представлено в авторском изложении.

Ответственный редактор  
**капитан-лейтенант Ржавитин В.Л.**  
Компьютерная верстка  
**Минасян М.А.**  
**Репин Д.В.**

АНАЛИЗ СУЩЕСТВУЮЩИХ КОМПОЗИТНЫХ МАТЕРИАЛОВ И ОПРЕДЕЛЕНИЕ ПЕРСПЕКТИВ ИХ ДАЛЬНЕЙШЕГО РАЗВИТИЯ. ....	181
<b>Фролов А.В., Плотникова Я. Р.</b>	
ИСПОЛЬЗОВАНИЕ КОМПОЗИЦИОННЫХ МАТЕРИАЛОВ В АВИАЦИИ .....	188
<b>Шайдуллин И.Н., Смелик А.А., Шевченко Я.В., Губанов Е.В.</b>	
ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ ПРОЗРАЧНЫХ БРОНЕМАТЕРИАЛОВ .....	196
<b>Бакеев М.М., Фролов А.В.</b>	
НАПРАВЛЕНИЯ СОВЕРШЕНСТВОВАНИЯ ТЕХНИЧЕСКИХ СРЕДСТВ СЛУЖБЫ ГОРЮЧЕГО АРМИЙ СТРАН НАТО .....	203
<b>Трусов Д.Н., Вдовичев А.А., Ржавитин В.Л., Смелик А.А.</b>	
АНАЛИЗ РАЗВИТИЯ КОНСТРУКЦИИ ТОПЛИВНЫХ ФОРСУНОК .....	209
<b>Марков А.Р., Горшков С.Н., Иконников А.В.</b>	
АЛГОРИТМЫ ОЦЕНКИ КАЧЕСТВА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ .....	220
<b>Горшков С.Н., Маслов Н.С.</b>	
АНАЛИЗ БИБЛИОТЕК BOOST.STATECHART И BOOST.META STATE MACHINE ДЛЯ РЕШЕНИЯ ЗАДАЧИ УПРАВЛЕНИЯ БПЛА КОПТЕРНОГО ТИПА .....	225
<b>Иконников А.В., Марков А.Р., Горшков С.Н.</b>	
ПОДХОДЫ В ОБРАБОТКЕ ИЗОБРАЖЕНИЙ ПРИ РАСПОЗНАВАНИИ ОБЪЕКТОВ .....	231
<b>Усеинов И.А., Щербанев А.Ю., Кириченко А.А., Коваленко Р.В., Горшков С.Н.</b>	
ПРИНЦИП РАБОТЫ С НАВИГАЦИОННОЙ АППАРАТУРОЙ ПРИ ОРИЕНТИРОВАНИИ НА МЕСТНОСТИ .....	238
<b>Прокофьев М.А., Поляков Р.Г., Горшков С.Н.</b>	
АНАЛИЗ СТОХАСТИЧЕСКОГО ПОДХОДА К МОДЕЛИРОВАНИЮ БОЕВЫХ ДЕЙСТВИЙ .....	245
<b>Захаренков И.Г., Горшков С.Н.</b>	
МЕТОДИКА РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ МОБИЛЬНЫХ ТЕРМИНАЛОВ ОПЕРАТОРА РОБОТОТЕХНИЧЕСКИХ КОМПЛЕКСОВ.....	253
<b>Поляков Р.Г., Горшков С.Н., Прокофьев М.А.</b>	
СРАВНИТЕЛЬНЫЙ АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ БОЕВЫХ ДЕЙСТВИЙ .....	268

# МЕТОДИКА РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ МОБИЛЬНЫХ ТЕРМИНАЛОВ ОПЕРАТОРА РОБОТОТЕХНИЧЕСКИХ КОМПЛЕКСОВ

**Захаренков И.Г.**, магистр, оператор 4 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация, e-mail: [stepler028@gmail.com](mailto:stepler028@gmail.com), тел.: 89030164737.

**Горшков С.Н.**, старший оператор 4 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация, e-mail: [sergik-gorshkov@mail.ru](mailto:sergik-gorshkov@mail.ru), тел.: 89097633050.

## Аннотация

В статье представлена работа по созданию методики разработки пользовательского для мобильных терминалов оператора робототехнических комплексов. Рассматриваются задачи построения удаленных терминалов системы управления РТК. Представлен анализ современных разработок для мобильных терминалов. Описаны этапы разработки механизма конфигурации мобильного терминала на основе формата JSON.

**Ключевые слова:** военный инновационный технополис «ЭРА»; робототехнический комплекс (РТК); мобильный терминал оператора; человеко-машинный интерфейс (HMI); пользовательский интерфейс (UI); пульт ДУ.

Современные тенденции развития военных робототехнических комплексов направлены на создание дистанционно управляемых устройств, предназначенных для решения различных задач. В настоящее время большое применение получили беспилотные летательные аппараты (БПЛА). Наземные и морские РТК являются отстающими и не достигли таких же показателей.

Ограничение использования дистанционно управляемой техники обусловлено техническими трудностями и проблемами внедрения в современный облик Вооруженных Сил РФ.

Распространение боевых РТК может привести к необходимости создания специальных систем передачи данных, которая в свою очередь позволит унифицировать пульта дистанционного управления (ДУ) для различных РТК.

Пульт управления – устройство созданное для упрощения работы и мобильности оператора РТК. Пульта могут быть выполнены как в виде простых генераторов импульсов, так и в виде интеллектуальных устройств, с программируемым поведением кнопок [1].

Выносной (дистанционный) предназначен для ручного управления требуется мобильность. Он должен отвечать следующим требованиям:

- безопасность;

- удобство в обращении;
- прочность;
- практичность.

Аппаратный интерфейс пульта осуществляется через специальный защищенный блок (IP65). Локальное соединение осуществляется на расстоянии до 50 м. Специальный блок поддерживает связь с устройством управления и ядром системы управления (СУ).

Терминальная задача СУ ориентирована на проявление взаимодействия устройства СУ с окружающей средой: прежде всего это диалог с оператором, в том числе с другими системами управления. Для поддержания диалога существуют технические средства. Техническими средствами поддержания диалога являются прежде всего пассивный терминал (панель оператора) или активный терминал (персональный компьютер), а кроме того, интерфейс с управляющими устройствами внешней среды (рис.1). Терминальная задача обозначилась столь явно, что терминал устройства ЧПУ все чаще становится отдельным конструктивным элементом. [1]

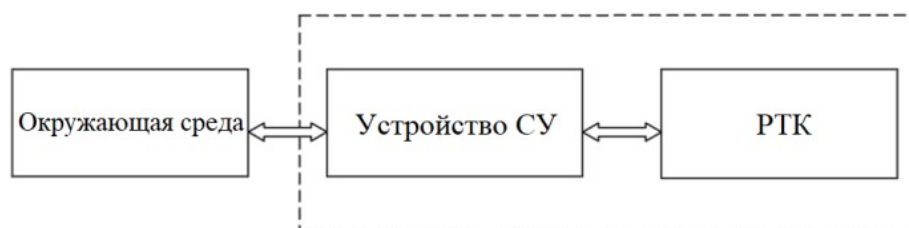


Рисунок 1 – Система внешних взаимодействий

Терминальная задача поддерживается современными устройствами практически также, как персональным компьютером. Диалог с оператором охватывает управление РТК в различных режимах, операции с СУ, некоторые действия системного характера. Информационный обмен необходим в интегрированной пространственной распределенной системе управления, имеющей признаки локальной вычислительно-управляющей сети. Одним из видов взаимодействия СУ РТК с окружающей средой является ввод информации (программы управления/маршрута). Для этого используются различные считывающие устройства.

Существует 2 вида решения терминальной задачи:

Классический – штатный терминал системы оператора, соединенный кабелем через ядро системы РТК с выносным пультом. Выносной пульт ручного предназначен для случаев, когда от оператора при управлении требуется мобильность. Пульт должен отвечать требованиям безопасности, удобства в обращении, прочности и практичности. Плюсами использования такого пульта являются: наличие физического джойстика, которым удобно управлять движением РТК, а также подключение с помощью защищённого от электро-

магнитных помех кабеля. К минусам использования можно отнести подключение при помощи кабеля ограниченной длины, который не позволяет оператору свободно передвигаться, имеет большие габаритные размеры и стандартный набор элементов управления.

Специфические – в основном направленные на повышение мобильности оператора РТК:

1. Web-терминал – справляется с задачей управления РТК. Особое место занимает задача создания и организации межмодульного взаимодействия основанная на базе web-технологий с применением web-сервера данных. Распределённая система управления производством формируется посредством набора web-серверов, специализирующихся на реализации двух основных задач это предоставление удалённого доступа к функциям диагностики и управления оборудованием и организацией интерактивного взаимодействия оператора. Web-сервер предоставляет доступ к функциям системы управления РТК web-клиентам, позволяя им осуществлять мониторинг, диагностирование и управление комплексом.

Плюсы: данный подход позволяет удаленно подключиться к системе управления с любого устройства, имеющего доступ к локальной сети или интернет (в зависимости от конфигурации сети и веб-сервера) и наличии веб-браузера.

Минусы: требует загрузки web-страниц, которые могут по-разному отображаться в различных версиях браузеров. Так же мобильный терминал использует в качестве промежуточного звена связи web-сервер.

2. Мобильный терминал – является перспективным направлением. Такой терминал может быть построен на базе мобильного приложения. Такое приложение может быть написано при помощи универсального кроссплатформенного решения Qt (или Qt + QML) [2].

Преимуществом данного подхода реализации терминальной задачи является то, что у многих людей имеется в наличии мобильное устройство, и по сравнению с web-терминалом мобильное приложение не требует загрузки web-страниц. Так же мобильный терминал не использует в качестве промежуточного звена связи web-сервер, что повышает надёжность использования терминала, так как исключается отказ работы web-сервера. К недостаткам данного подхода можно отнести то, что мобильный терминал взаимодействует с СУ РТК через беспроводную сеть Wi-Fi, которая имеет слабую защищенность от помех.

В настоящее время портативные устройства получили широкое распространение в мире вследствие своей доступности и большому списку решаемых задач. Это стало возможным благодаря увеличению вычислительной

мощности электроники и вследствие этого применению операционных систем.

Поскольку мобильные операционные системы имеют разные «экосистемы» для разработки приложений, а также имеют разные уровни открытости, пользовательские интерфейсы, доступ к файловой системе, необходимо учитывать все различия между ними. Существенные различия операционных систем требуют различного инструментария создания приложений.

В настоящее время существует множество решений для построения мобильных приложений, среди них можно выделить PhoneGap, Xamarin, Qt. Ниже приведена таблица (таблица 1) основных характеристик:

*Таблица 1. Сравнительный анализ инструментальных средств.*

Фреймворк	Язык разработки	Лицензия	Поддержка встраиваемых систем	Высокая производительность приложения
PHONEGAP	JS, HTML5	APACHE	Нет	Нет
XAMARIN	C#	MIT	Нет	Да
QT	C++, QML	GPLV2	Да	Да

Преимущества Qt заключаются в использовании языка программирования C++, который является более производительным за счёт компиляции в машинный код под конкретную платформу, а не под виртуальную машину как в случае с C#, или интерпретации в браузере (JavaScript). Возможность компиляции для встраиваемых систем так же является преимуществом, т.к. это позволяет использовать приложение не только на мобильных устройствах, но и во встраиваемых системах. Так же неоспоримым преимуществом является наличие open-source лицензии.

Преимущества Qt являются:

- хорошая графика: Qt Quick (QML) базируется на OpenGL ES;
- производительность разработки: приложение разделяется на front-end (QML + JavaScript) и back-end (Qt C++) части, что способствует front-end разработчикам сфокусироваться на разработке красивого UI, а back-end разработчики уделяют больше внимания стабильности и производительности;
- кроссплатформенность: можно разработать один код сразу под все поддерживаемые операционные системы (Windows, Linux, MacOS, Android, iOS и др.);
- открытость: Qt является активно развивающимся open-source проектом, разработка которого ведётся открытым сообществом.

Мета-объектный язык QML возможно использовать в качестве связи программиста с графическим разработчиком (дизайнером).

С помощью QML дизайнер может говорить с разработчиком на одном языке, они могут модифицировать исходный код. Это обеспечивает возможность создавать прототипы программ достаточно быстро. Созданный дизайнером прототип на языке QML будет выглядеть и уже являться стартовой версией для готового приложения, с помощью которого разработчики смогут работать дальше. Прототипы сразу могут тестироваться в реальных условиях, каждый разработанный прототип будет доступен к использованию.

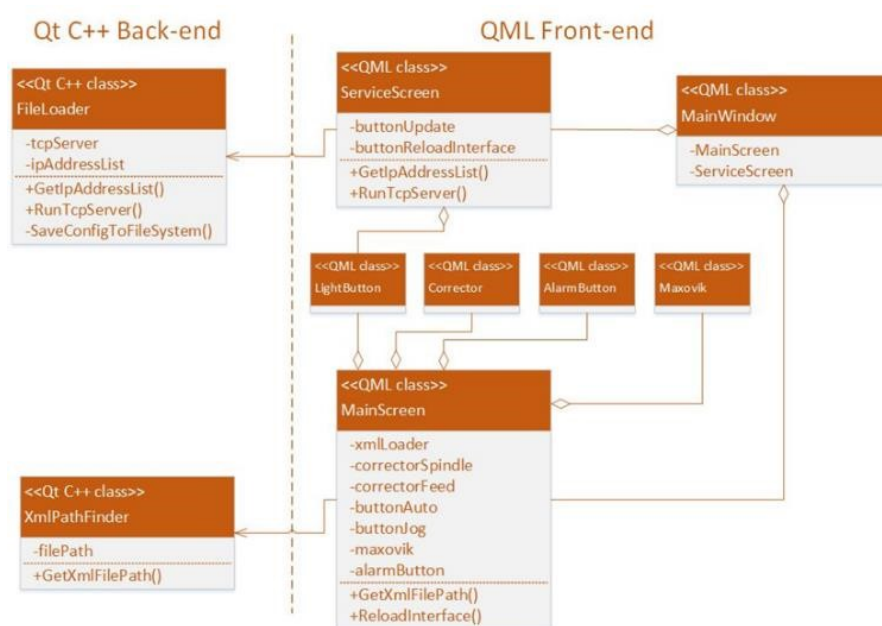


Рисунок 2 – Архитектура приложения мобильного терминала с использованием Qt-QML

Методика разработки заключается в использовании связки Qt-QML, благодаря которой пользовательский интерфейс можно разделить на 2 части (рис.3):

1. Front-end. Для реализации используется язык программирования QML. Создается дизайн пользовательского интерфейса.
2. Back-end. Используется Qt (C++) для создания логики приложения, работы с сетью, файловой системой и т.п.

В качестве используемого формата данных используется JSON – «гибкий» формат, нацеленный на отправку данных между веб-серверами и браузерами или приложениями [3].

Гибкость обуславливается отсутствием строгой схемы, такая гибкость часто затрудняет чтение и понимание данных.

Применяя кроссплатформенную среду разработки Qt вместе с языком разметки QML не сложно создать необходимые для пользовательского интерфейса элементы управления. Оператор сможет менять конфигурацию этих элементов с помощью XML-документа.

В случае, если у оператора нет доступа к СУ РТК, то пользовательские настройки интерфейса можно загрузить туда, откуда потом его можно было бы легко прочесть.

Проще говоря, решением проблемы является реализация записи файла настроек самим приложением. В приложении создается сервисный экран, на котором запускается ТСР-сервер. Он ожидает приема входных соединений. Параллельно на компьютере запускается клиент, вводится IP-адрес. После подключения, загружается XML-документ. Затем в приложении, при нажатии на кнопку «refresh» происходит обновление интерфейса.

Самым простым элементом является создание «кнопки с индикацией». Кнопка с индикацией (рис.3) - элемент управления, который эмулирует реальную кнопку. Элемент предназначен для выбора режима работы РТК или его отдельных функций.

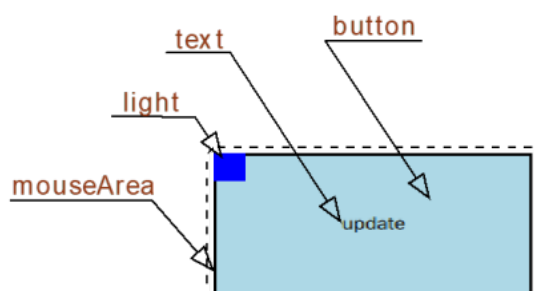


Рисунок 3 – Кнопка с индикацией

На диаграмме классов (рис.4) представлены классы стандартной библиотеки QML и самостоятельно разработанный класс. Созданный класс LightButton содержит класс button (Rectangle), light (Rectangle), text (Text), mouseArea (MouseArea). Классы Rectangle, Text, MouseArea являются стандартными из набора библиотеки QML.

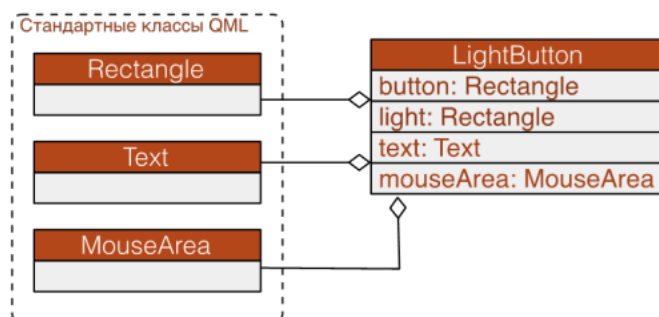
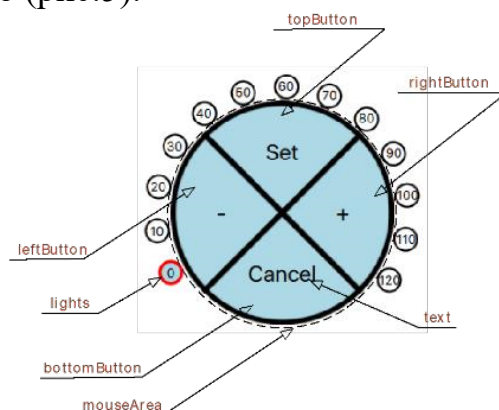


Рисунок 4 – Диаграмма классов кнопки с индикацией



Многопозиционный переключатель или селектор – элемент управления, который эмулирует механический потенциометр. Он позволяет выбрать нужную позицию при помощи кнопки уменьшения позиции и кнопки увеличения позиции. Эти кнопки передвигают индикатор против и по часовой стрелке соответственно (рис.5).



*Рисунок 5 – Многопозиционный переключатель*

В случае использования пульта ДУ с маленьким есть большая вероятность промахнуться, нажав не на ту кнопку. Чтобы не допускать таких аварийных ситуаций имеются две кнопки: кнопка установки значения (SET) и кнопка отмены (CANCEL). Когда выбирается нужная позиция, то позиция окончательно устанавливается после нажатия кнопки SET, при этом индикатор меняет цвет на красный. Затем, если необходимо изменить значение, то нажимая на кнопки (+) или (-), то выбранное ранее значение останется подсвечено красным, а выбор установки значения продолжится и индикатор с голубом цветом будет перемещаться. При случайном нажатии не на ту кнопку можно отменить установку выбранной позиции с помощью кнопки отмены и индикатор поменяет свой цвет обратно на голубой.

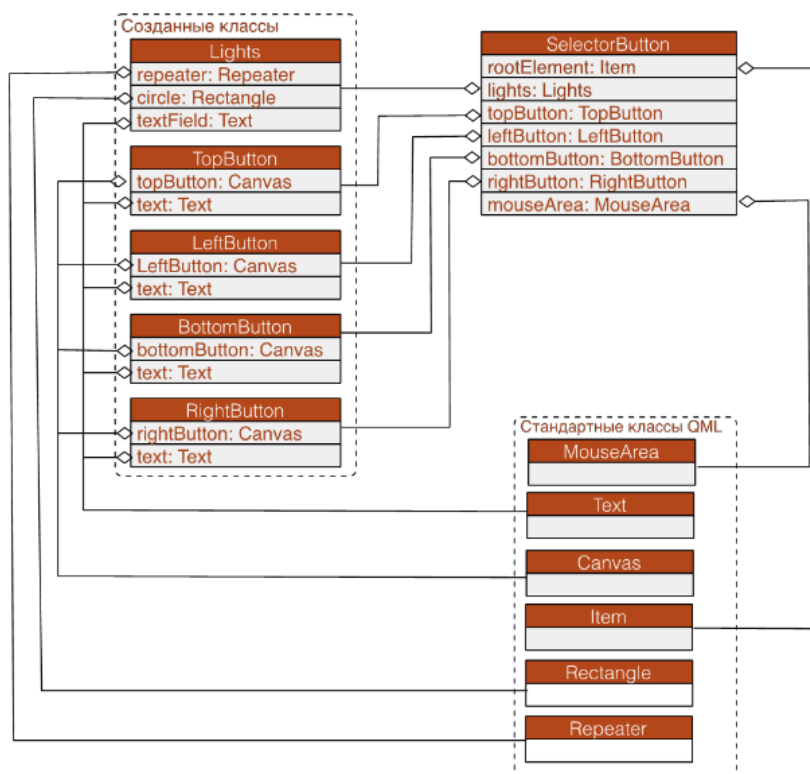


Рисунок 6 – Диаграмма классов многопозиционного переключателя

На диаграмме классов (рис.6) представлен вид данного элемента представления. Созданный класс SelectButton является родительским для всех остальных классов: rootElement (Item), lights (Lights), topButton (TopButton), leftButton (LeftButton), bottomButton (BottomButton), rightButton (RightButton), mouseArea (MouseArea).

Маховик (рис.7) является устройством управления приводами механизмов. Используется для преобразования угловых перемещений в последовательность электрических сигналов, которые содержат информацию о направлении и величине перемещения.

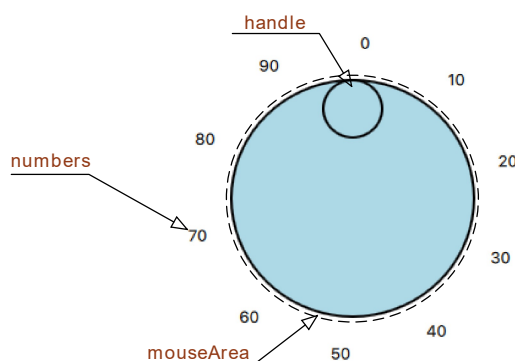


Рисунок 7 – Маховик

Как правило, маховиками оператор пользуется для выполнения точных операций в случае управления схватом РТК. Такой эмулятор обеспечивает необходимые требования для работы оператора.

На диаграмме классов (рис.8) показано, что для реализации эмулятора маховика необходимо создать классы. В классе Handwheel происходит отрисовка самого маховика в месте с джойстиком. А в классе Numbers – шкала, которая так же как и элемент могут изменяться оператором.

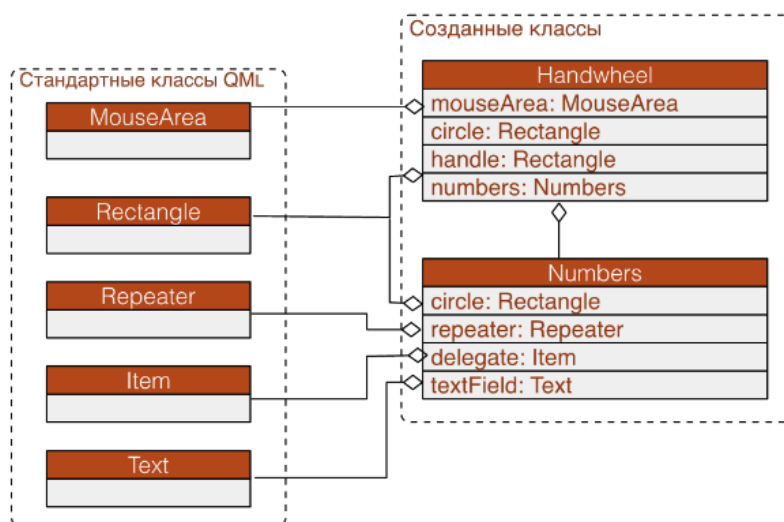


Рисунок 8 – Диаграмма классов маховика

Для связи ядра СУ РТК и терминальных клиентов применяется сетевой стандарт TCP/IP. Ключевым моментом является наличие двух каналов связи между ядром системы и терминальными клиентами, как это показано на рисунке 9.

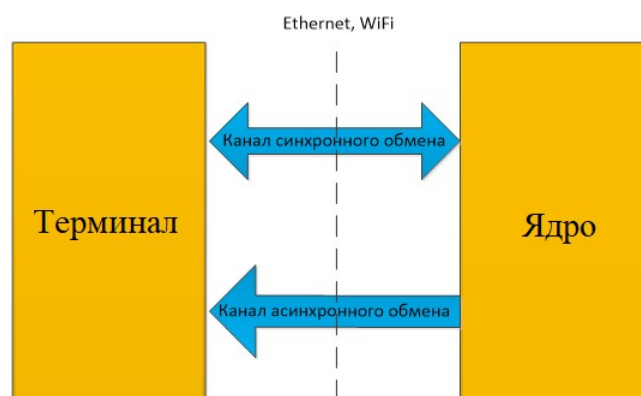


Рисунок 9 – Взаимодействие терминала с ядром

Канал синхронного обмена является основным, он обеспечивает получение данных от ядра, передачу данных в ядро и посылку ядру команд.

Канал асинхронного обмена предназначен для уведомления клиентов об изменениях, происходящих в ядре.

Взаимодействие реализуется посылкой и приёмом пакетов данных, структура которых ниже будет полностью раскрыта. Таким образом, текущий протокол является открытым.

На логическом уровне определены понятия, использование которых позволяет снизить трафик при решении стандартных задач. В частности, определён механизм подписки на изменения данных. Например, клиент может получить текущие координаты осей и «подписаться» на получение уведомлений об их изменении. Тогда представление клиента о координатах осей синхронизируется автоматически посылкой соотв. уведомлений по асинхронному каналу.

### Методика разработки.

На рисунке 10 представлена разработанная методика построения удалённых терминалов.

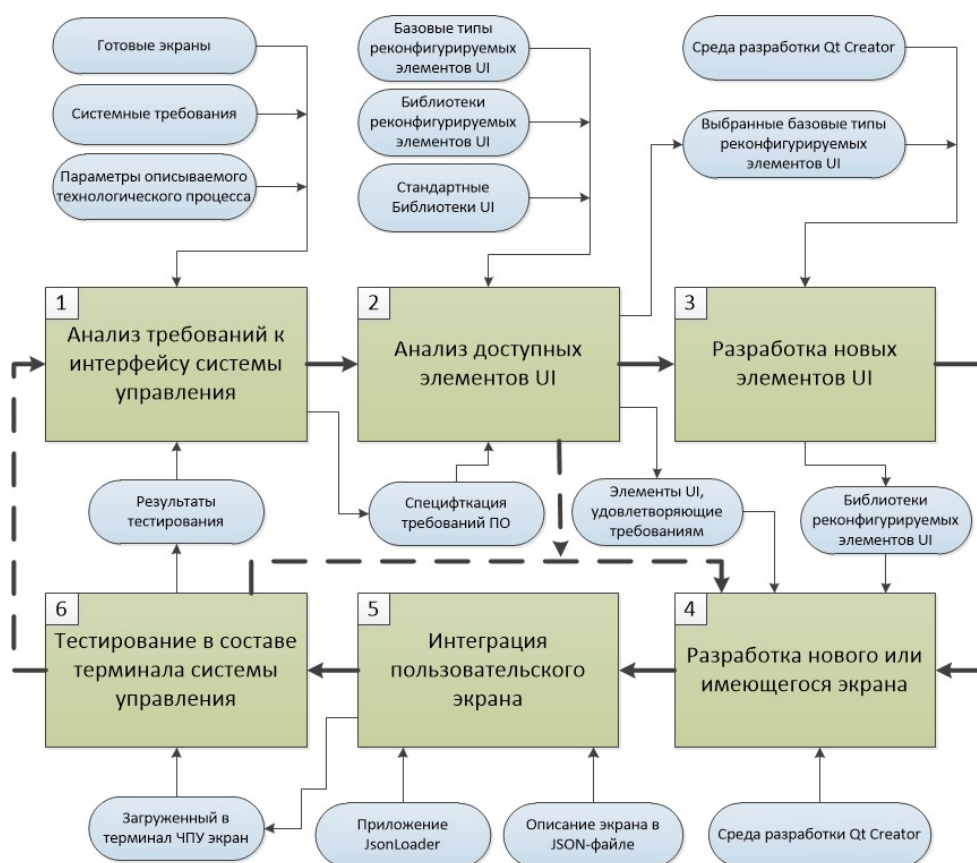
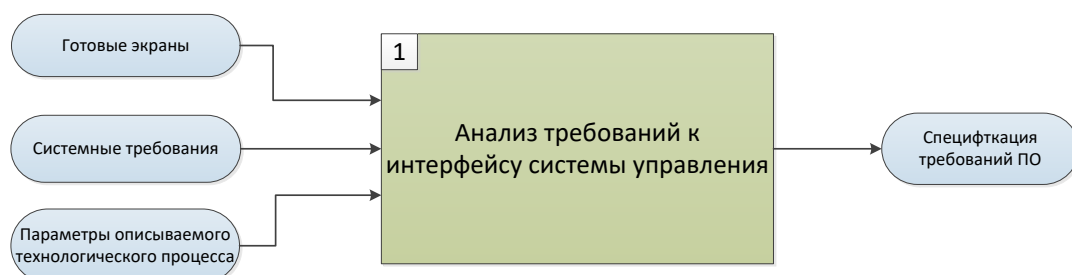


Рисунок 10 – Методика построения удалённых терминалов

1. Необходимо провести анализ требований к интерфейсу системы управления:
  - узнать какие параметры технологического процесса должны отображаться в мобильном терминале;
  - узнать системные требования устройства, на котором будет работать терминал, включая размер экрана и его разрешение;

- посмотреть в библиотеке уже реализованных экранов, есть ли подходящее решение.
2. Затем необходимо:  
Изучить наличие доступных элементов: проверяем наличие базовых типов, проводим поиск в библиотеке уже реализованных реконфигурируемых компонент;
  3. Если устраивает набор компонент, то двигаемся к пункту 4, если не устраивает, то создаём новые компоненты при помощи среды Qt Creator из базовых примитивов, либо из реализованных разработчиками мобильного терминала. На выходе должна получиться либо новая библиотека компонент, либо должна быть расширена уже имеющаяся;
  4. Как только мы удовлетворились имеющимися или вновь созданными компонентами, переходим к компоновке экрана этими самыми элементами. Для этого используем среду Qt Creator, либо конфигурируем это с помощью json.
  5. Далее проводим интеграцию экрана в мобильный терминал при помощи ScreenLoader'a (загружаем файл настроек на мобильное устройство);
  6. Далее проводим ручное тестирование полученного терминала. Если тестирование прошло успешно, сохраняем экран, если нет, то возвращаемся к п.1 или п.4.

Прежде чем начать проектирование нового интерфейса мобильного терминала необходимо выяснить каковы требования заказчика или спецификация управляемого РТК (рис.11).



*Рисунок 11 – Анализ требований к интерфейсу*

После того, как заказчик предоставит свои требования к экрану, необходимо узнать, не решалась ли данная задача ранее, для этого необходимо поискать в библиотеке компонентов уже готовый экран, который, сможет удовлетворить требованиям.

Чем больше элементов управления, тем мощнее должно быть устройство, на котором будет запускаться мобильный терминал.

На выходе из этого этапа должна получиться спецификация требований к разрабатываемому программному обеспечению.

После того, как были получены требования в виде спецификации, необходимо понять какими средствами для реализации будущих экранов располагают разработчики. Для этого необходимо знать какие базовые примитивы доступны в стандартной библиотеке компонентов, например это могут быть прямоугольники, круги, эллипсы, треугольники и др (рис.12).

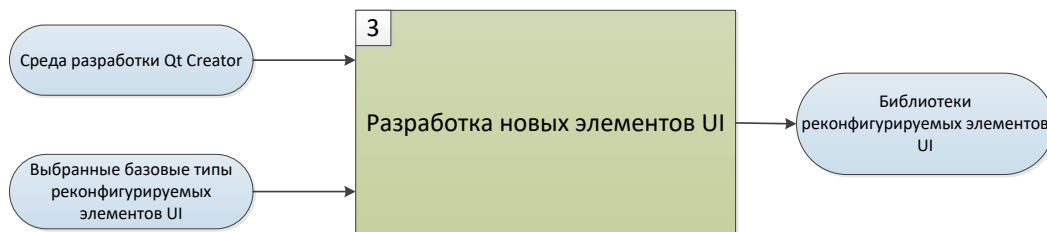
Так же необходимо знать какие базовые элементы, которых не было в стандартной библиотеке были реализованы разработчиками. Например это может быть шестиугольник, которого нет в стандартной библиотеке.



*Рисунок 12 – Анализ имеющихся функций управления*

После того как на основании требований будет проведен первичный анализ доступных элементов, необходимо выбрать базовые типы реконфигурируемых элементов, а так же элементы UI, удовлетворяющие требованиям спецификации.

Если вдруг по какой-то причине не были найдены компоненты, удовлетворяющие требованиям спецификации, то придётся их разрабатывать самим. Реализовать задуманные компоненты необходимо будет при помощи среды разработки компонентов. Для данной работы такой средой является Qt Creator, позволяющая создавать новые элементы на декларативном языке QML. Новые компоненты необходимо разрабатывать, используя предыдущие наработки, тем самым не нарушать принцип DRY – Don't repeat yourself – принцип разработки программного обеспечения, нацеленный на снижение повторения информации различного рода, особенно в системах со множеством слоёв абстрагирования. Принцип DRY формулируется как: «Каждая часть знания должна иметь единственное, непротиворечивое и авторитетное представление в рамках системы»[4].

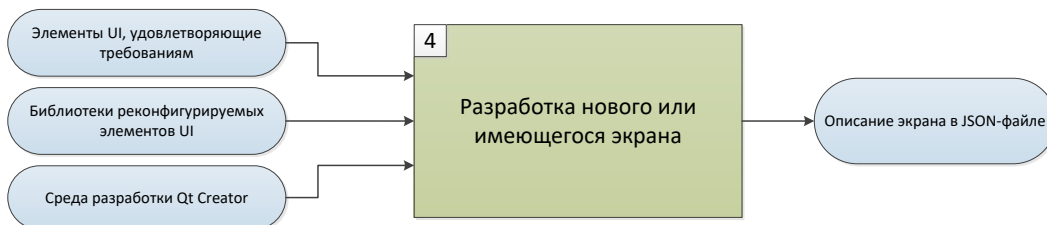


*Рисунок 13 – Реализация новых функций управления*

На выходе, после завершения данного этапа (рис.13) должны появиться новые элементы, которые должны быть скомпонованы в библиотеки реконфигурируемых элементов. Так же важно, следуя принципу DRY сделать компоненты настраиваемыми, это можно сделать, выделив базовые абстракции, и сформировать на их основе независимый код, который будет иметь возможность параметризации.

Например, не нужно жестко задавать цвет и размеры элементов, необходимо сделать эти параметры доступными пользователям, разрабатываемой библиотеки.

Получив элементы UI, удовлетворяющие требованиям спецификации, а так же имея библиотеки реконфигурируемых элементов, а так же пользуясь средой разработки Qt Creator, позволяющей произвести тестовую проверку экранов терминала, без загрузки их на мобильное устройство, что является довольно затратным по времени мероприятием, на локальном компьютере, и если надо произвести отладку.

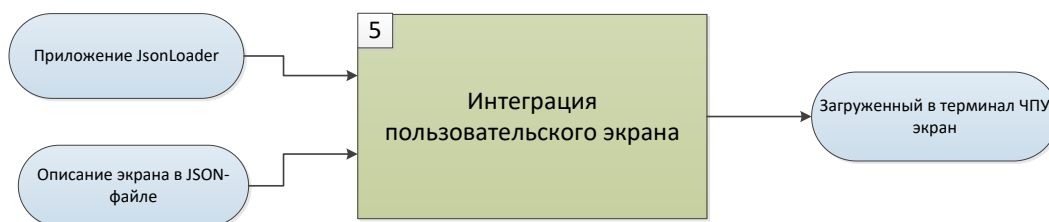


*Рисунок 14 – Реализация нового пользовательского экрана*

Как только разработчик убедится в штатной работе всех элементов, необходимо сохранить результаты в JSON-файл (рис.14). В данном файле должны быть указаны координаты, размеры, цвета и функции элементов.

Имея в распоряжении описание экрана настроек в формате JSON, которые должны быть проверены на предыдущем этапе – не должно быть синтаксических и логических ошибок, при помощи приложения JsonLoader необходимо загрузить данные настройки на мобильное устройство (рис.15).

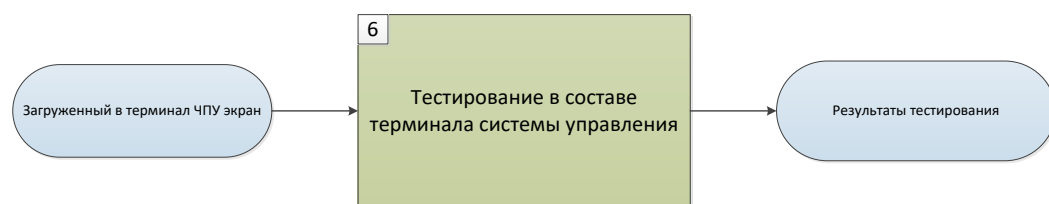




*Рисунок 15 – Интеграция пользовательского экрана*

Результатом пройденных этапов является сконфигурированный пользовательский интерфейс, загруженный в мобильный терминал оператора РТК.

По завершению загрузки экрана на мобильное устройство необходимо провести интеграционное тестирование самими разработчиками. Далее можно отдать устройство для тестирования специальным командам тестировщиков (рис.16). Отслеживание результатов тестирования можно осуществлять через баг-трекер. Примером баг-трекинг-системы может служить программное обеспечение Bugzilla, либо можно осуществлять отслеживание баг-репортов при помощи ПО Redmine или любого другого.



*Рисунок 16 – Отладка функций управления*

По прохождению интеграционного тестирования обязательно должны получиться результаты, которые если будут признаны удовлетворительными, то разработку мобильного терминала можно будет считать успешно завершённой, либо могут быть выявлены недостатки и тогда на их основе должны быть выявлены новые требования к программному обеспечению.

**Заключение.** Разработана методика решения терминальной задачи на базе современных технологий с использованием кроссплатформенной среды Qt и формата данных JSON. Создано единое архитектурное решение для реализации терминальной задачи для мобильных и стационарных платформ, позволяющее осуществлять управление РТК, производить их диагностику обучение операторов.

## Литература

1. Евстафиева С.В., Лукьянов А.В., Мартинова Л.И., Пушков Р.Л. Разработка аппаратных компонентов системы ЧПУ с использованием современных САПР // Автоматизация в промышленности, №9, 2014. с.35-39.



2. Козак Н. В., Мартинова Л. И., Савинов К. А., Дубровин И. А. Построение гетерогенной распределенной компьютерной системы управления для высокотехнологичных децентрализованных производств на основе web-технологий // Мехатроника, автоматизация, управление. 2011. № 11. с. 44-48.
3. Ibryam B. Principles of Container-Based Application Design //Redhat Consulting Whitepaper. – 2017.
4. Lin B. et al. Comparison between JSON and XML in Applications Based on AJAX //2012 International Conference on Computer Science and Service System. – IEEE, 2012. – С. 1174-1177.