

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 537.633.9

ПРИНЦИП ИНДУКЦИОННО-БАЛАНСНОГО МЕТАЛЛОДЕТЕКТИРОВАНИЯ

В.И. АРЖАНОВСКИЙ
(Представлено: В.М. ЧЕРТКОВ)

Представлены области применения металлодетекторов, рассмотрены и подробно описаны принципы работы, отображены достоинства и недостатки каждого из выделенных видов. Подробно рассмотрен индукционно-балансный металлодетектор.

Металлодетектор – электронный прибор, позволяющий обнаруживать металлические предметы в нейтральной или слабопроводящей среде за счет их проводимости. Металлоискатель обнаруживает металл в грунте, воде, стенах, в древесине, под одеждой и в багаже, в пищевых продуктах, в организме человека и животных и т. д. С каждым годом расширяется область использования металлоискателей в самых различных сферах. Для военных металлоискатель – это прежде всего миноискатель, применяется в охранных структурах и криминалистике, предприятиями по производству пищевых продуктов, в строительстве и в процессе ремонтных работ, в процессе обработки древесины, при сортировке мусора на мусороперерабатывающих предприятиях, при добыче полезных ископаемых, в археологии. [1]

Существует несколько принципов работы металлодетекторов. Рассмотрим основные из них:

1. Приборы типа «прием – передача». В основе их лежат две катушки индуктивности – приемная и передающая, расположенные так, чтобы сигнал, излучаемый передающей катушкой, не просачивался в приемную катушку. Когда вблизи прибора появляется металлический предмет, то сигнал передающей катушки переизлучается им во всех направлениях и попадает в приемную катушку, усиливается и подается на блок индикации. Достоинством такой системы является относительно простая схемотехника, широкие возможности для определения типа обнаруженного объекта. Недостатки заключаются в сложности изготовления датчика, влиянии минерализации грунта, относительно невысокой чувствительности.

2. Индукционные металлоискатели. Представляют собой разновидность приборов типа «прием–передача», однако в отличие от последних содержат не две, а только одну катушку, которая одновременно является и передающей и приемной. Основной трудностью при создании подобных приборов является выделение весьма малого отраженного (наведенного) сигнала на фоне мощного передаваемого (излучаемого). Достоинством является простота конструкции датчика.

3. Приборы – измерители частоты. В их основе лежит LC-генератор. При приближении металла к контуру его частота изменяется. Это изменение фиксируется методом смешивания частоты генератора с эталонной и измерения частоты биений, подачей сигнала с генератора на систему ФАПЧ и измерением напряжения в цепи обратной связи. Положительным моментом является простота конструкции датчика, простая схемотехника. Недостатками являются худшие возможности дискриминации обнаруженных объектов, малая чувствительность.

4. Приборы, фиксирующие изменение добротности колебательного контура, входящего в состав LC-генератора. При приближении металлического предмета к катушке добротность контура уменьшается, и амплитуда колебаний на выходе LC-генератора также уменьшается. У этого типа приборов простая конструкция, малая потребляемая мощность, однако низкая температурная стабильность.

5. Импульсные металлоискатели – принцип работы основан на возбуждении в зоне расположения металлического объекта импульсных вихревых токов и измерении вторичного электромагнитного поля, которое наводят эти токи. В данном случае, возбуждающий сигнал передается в катушку датчика не постоянно, а периодически, в виде импульсов. В проводящих объектах наводятся затухающие вихревые токи, которые возбуждают затухающее электромагнитное поле. Поле, в свою очередь, наводит в катушке датчика затухающий ток. Соответственно, в зависимости от проводящих свойств и размера объекта, сигнал меняет свою форму и длительность. Такие металлодетекторы нечувствительны к минерализованному грунту, просты конструкции датчика, однако у них повышенное потребление энергии, слабые возможности дискриминации.

В профессиональных металлоискателях могут совмещаться несколько способов обнаружения объектов [2].

Рассмотрим подробно принцип работы индукционно-балансного металлодетектора.

Датчик состоит из двух катушек индуктивности, направление намоток которых совпадают. Одна катушка запитана синусоидальным напряжением (передающая катушка – на рисунках изображена красным цветом), а с выводов второй катушки (приемная катушка – на рисунках изображена синим цветом) снимается сигнал. Результат на выходе будет зависеть от относительного положения одной катушки относительно другой. Первый случай – на рисунке 1 передающая и приемная катушки лежат на одной оси и в одной плоскости.

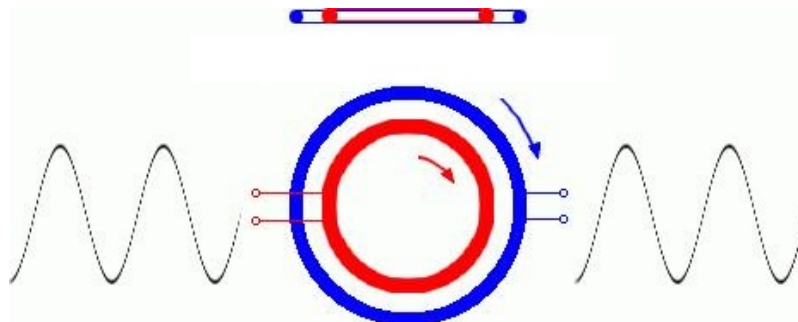


Рис. Ошибка! Текст указанного стиля в документе отсутствует.. Передающая и приемная катушки лежат в одной плоскости

На выходе приемной катушки амплитуда сигнала будет чуть меньше входного, фазы сигнала на входе и на выходе совпадают – фактически это обычный трансформатор, только без сердечника.

Если расположить одну катушку рядом с другой, как показано на рисунке 2, то правая часть витков передающей катушки наведет в левой части приемной катушки ток, так что на ее выходе напряжение будет противофазно входному, а его амплитуда будет меньше, так как у обеих катушек только небольшие площади проводников взаимодействуют между собой.

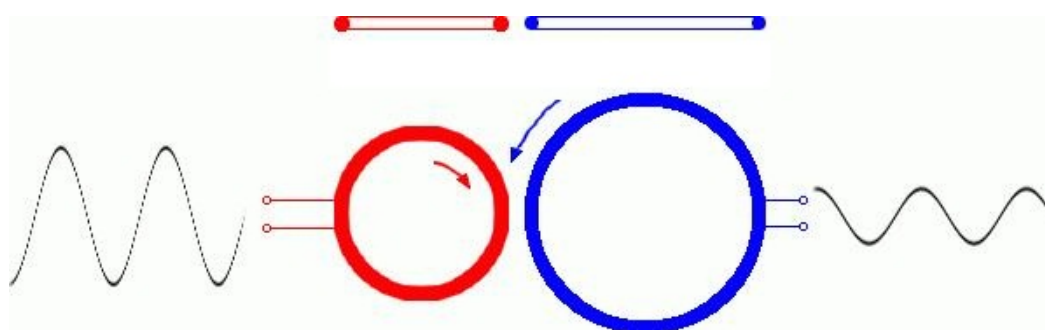


Рис. 1. Расположение одной катушки рядом с другой

Итак, при перемещении передающей катушки за пределы приемной на выходе поменялась фаза сигнала и его амплитуда. Но фаза резко поменяться не может. На рисунке 3 показано геометрическое положение катушек одна относительно другой, когда на выходе приемной катушки будет отсутствовать всякий сигнал (либо он будет ничтожно мал).

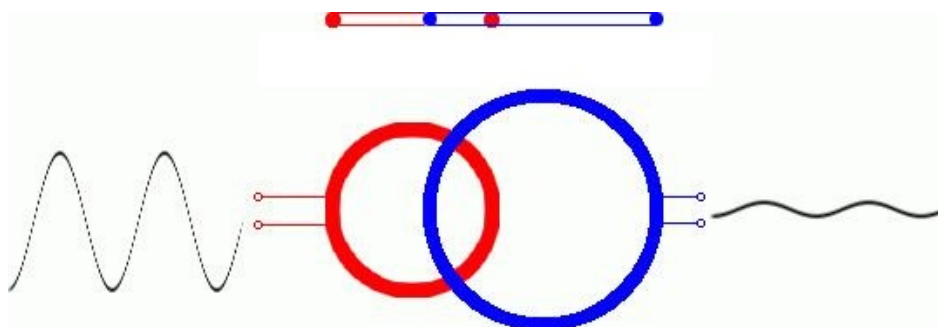


Рис. 2. Геометрическое положение катушек одна относительно другой, когда на выходе приемной катушки будет отсутствовать всякий сигнал

В данном случае на приемную катушку наводятся токи разных направлений, которые в ней взаимно компенсируются. Если теперь в магнитное поле попадет какой-либо предмет, то он исказит магнитное поле, и на выводах приемной катушки появится сигнал разбаланса, который после преобразования и зафиксирует контроллер [3].

Заключение. Изучив основные методы металлодетектирования можно сделать вывод, что наиболее удобным и вполне реализуемым является металлодетектор, работающий по принципу индукционного баланса.

ЛИТЕРАТУРА

1. Металлоискатели – электронные индукционные приборы [Электронный ресурс]. – Режим доступа: http://www.akmetrade.ru/Products/article_100/. – Дата доступа: 23.06.2015.
2. Металлоискатель по принципу [Электронный ресурс]. – Режим доступа: http://www.chipinfo.ru/literature/books/metal_detector/chapter6/part1.html. – Дата доступа: 20.06.2015.
3. История металлоискателей [Электронный ресурс]. – Режим доступа: http://zpostbox.ru/istoriya_metallodetektorov.html. – Дата доступа: 20.06.2015.

УДК 537.633.9

**СЕЛЕКТИВНЫЙ МЕТАЛЛОДЕТЕКТОР, РАБОТАЮЩИЙ ПО ПРИНЦИПУ
БАЛАНСА ИНДУКЦИИ С ПРЯМОЙ ОБРАБОТКОЙ
НА МИКРОКОНТРОЛЛЕРЕ STM32**

В.И. АРЖАНОВСКИЙ
(Представлено: В.М. ЧЕРТКОВ)

Приведены результаты научной работы по разработке устройства металлодетектирования. В результате проектирования разработаны структурная и функциональная схемы, произведен выбор и обоснование элементов схемы устройства, разработаны принципиальная схема устройства, исходный код программы микроконтроллера и конструктивное исполнение устройства.

Актуальность разработки обусловлена постоянным ростом рынка реализации готовой продукции, отсутствием аналогов отечественного производства, а также устройств зарубежного производства в целевой ценовой категории. Разрабатываемое устройство металлодетектирования обладает достаточно низкой себестоимостью для удовлетворения спроса широкого круга потребителей, и, вместе с тем, достаточной точностью для постоянного применения пользователем, уже имеющим опыт работы с аналогичными устройствами зарубежного производства.

Структурная схема устройства приведена на рисунке 1.



Рис. 3. Структурная схема металлодетектора

Назначение блоков:

1. Модуль питания – стабилизация напряжения питания до 3,3 В и 5 В, в зависимости от требуемого напряжения для каждого блока.
2. ЭСППЗУ – электрически стираемое перепрограммируемое постоянное запоминающее устройство, один из видов энергонезависимой памяти.
3. Устройство звуковой индикации – выводит звук с определенными параметрами, в зависимости от условий измерения.
4. Функциональные клавиши – с помощью этого блока происходит настройка прибора и листинг меню.
5. Устройство визуальной индикации – отображает наглядную информацию, текст меню, поисковые данные.
6. Выходной каскад – формирует сигнал для передающей катушки.
7. Передающая катушка – образует электромагнитную волну для распространения в пространство.
8. Приемная катушка – принимает электромагнитные колебания, отраженные от исследуемых объектов.
9. Входной усилительный каскад – усиливает сигнал с приемной катушки для дальнейшей обработки микроконтроллером.
10. Микроконтроллер – формирует выходной сигнал, обрабатывает входной сигнал, формирует вывод информации.

Для разработки принципиальной схемы металлодетектора и исходного кода программы микроконтроллера была рассмотрена внутренняя структура микроконтроллера и разработана методика его взаимодействия с остальными блоками устройства. Функциональная схема структуры микроконтроллера изображена на рисунке 2.



Рис. 2. Функциональная схема базового блока металлодетектора

Помимо процессора, производящего вычисления и управляющего работой контроллера, в данном устройстве использованы следующие периферийные модули:

– внешний программатор – ST-LINK/V2 программатор/отладчик для микроконтроллеров серии STM32. К контроллеру отладчик подключается посредством стандартного JTAG/SWD интерфейса, к компьютеру, через USB порт;

– JTAG/SW отладка – собственная встроенная отладочная система ЦПУ ARM. Порт JTAG является стандартным интерфейсом для подключения к ЦПУ, а также загрузки кода программы во внутреннее ОЗУ или Flash память. Порт JTAG также поддерживает базовые функции управления исполнением про-

граммы (пошаговое выполнение, установка контрольных точек и др.) и делает возможным просмотр содержимого ячеек памяти;

– ЦПУ Cortex–M3 – выполняет инструкции, производит вычисления в своем АЛУ (арифметико-логическом устройстве). Его гарвардская архитектура позволяет одновременно загружать инструкции и осуществлять доступ к памяти – благодаря этому, а также трехступенчатому конвейеру, большинство инструкций выполняются за 1 такт. Ядро Cortex–M3 поддерживает набор инструкций Thumb–2, который содержит как 32-битные, так и 16-битные инструкции для сокращения объема кода за счет менее дальнобойных переходов; имеет 13 регистров общего назначения, снижая потребность в частом доступе к памяти [1];

– КВВП – контроллер векторизованных вложенных прерываний. КВВП предоставляет стандартную структуру прерываний для всех Cortex-микроконтроллеров и способы их обработки;

– каналы ПДП – каналы прямого доступа к памяти. Каждый канал может использоваться для передачи данных между регистрами любого из УВВ и запоминающими устройствами 8-, 16- или 32-битными словами. Каждое из УВВ может выполнять роль потокового контроллера ПДП, при необходимости отправляя данные или посылая запрос на их получение. Арбитр внутренней шины и матрица шин минимизируют потребность в арбитраже доступа к шинам со стороны ЦПУ и каналов ПДП. Это означает, что блок ПДП является универсальным, простым в применении и реально автоматизирует передачу потоков и данных внутри микроконтроллера;

– матрица шин – набор высокоскоростных шин, через которые системная шина и шина данных подключаются к внешнему микроконтроллеру. Она образует несколько параллельных соединений между шинами Cortex и другими внешними шинными мастерами. Здесь шины соединены так, что устройства могут взаимодействовать напрямую, не через ядро;

– ФАПЧ – блок фазовой автоподстройки частоты. Выходная частота блока ФАПЧ зависит от заданного целочисленного значения коэффициента умножения частоты, которое хранится в регистре RCC_PLL_configuration. Если используется генератор на частоту 8 МГц, для генерации максимальной тактовой частоты 72 МГц необходимо задать коэффициент умножения 9;

– интегрированная флэш-память – память программ, которая представляет собой электрически стираемое ПЗУ. Она предназначена для хранения команд, управляющих работой микроконтроллера. Память программ также часто используется для хранения таблиц констант, не меняющихся во время работы программы;

– СОЗУ – статическое запоминающее устройство, используемое для хранения переменных;

– ЭСППЗУ – электрически стираемое перепрограммируемое ПЗУ, один из видов энергонезависимой памяти;

– таймер SysTick – предназначен для использования в качестве стандартного таймера во всех Cortex-микроконтроллерах. Таймер SysTick может использоваться для формирования шкалы времени в ОСРВ или для генерации периодических прерываний для обработки запланированных задач;

– ЦАП – устройство для преобразования цифрового кода в аналоговый сигнал;

– АЦП – устройство, преобразующее входной аналоговый сигнал в цифровой;

– блок питания – для работы микроконтроллеров STM32 их необходимо питать одним напряжением в диапазоне от 2,0...3,6 В. Для питания ядра Cortex напряжением 1,8 В в МК интегрирован стабилизатор напряжения. Однако к STM32 может быть подано еще два опциональных напряжения питания. В домене с отдельным питанием расположены часы реального времени и небольшое число регистров, что позволяет организовать их резервное питание и обеспечить работоспособность даже при нахождении МК в режиме полного отключения;

– генератор – в МК STM32 имеются внутренние RC-генераторы, способные синхронизировать встроенную схему ФАПЧ. Благодаря их совместной работе, МК могут синхронизироваться частотой до 72 МГц. Внутренние генераторы, по сравнению с кварцевыми, не отличаются такой же высокой точностью или стабильностью, поэтому, во многих применениях может потребоваться использование как минимум одного внешнего кварцевого генератора;

– приемный усилитель – состоит из двух каскадов. Оба каскада выполнены на ОУ. Первый обладает высоким входным сопротивлением благодаря последовательной ООС по напряжению. Это позволяет исключить потери полезного сигнала вследствие шунтирования колебательного контура входным сопротивлением усилителя. Второй каскад формирует искусственную среднюю точку для питания первого;

– усилитель мощности – построен на мощных ключах, усиливает сигнал и передает его на передающую катушку.

В ходе проектирования была создана макетная плата с помощью программного пакета для проектировки и ручной разводки печатных плат Sprint-Layout 6.0. Внешний вид одного из слоев платы в программе отображен на рисунке 3.

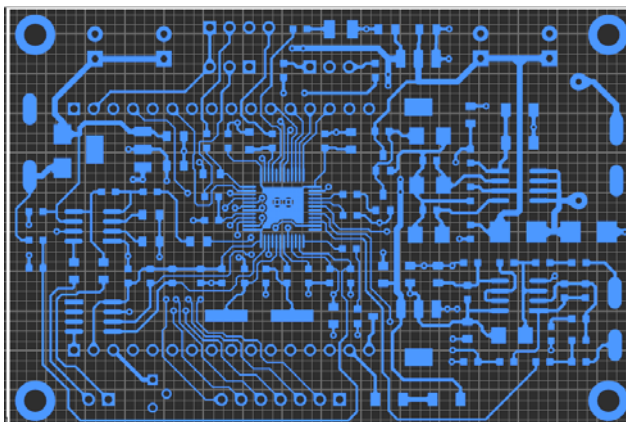


Рис. 3. Дорожки верхнего слоя печатной платы

На рисунке 4 изображена плата, на которой размещены и припаяны все элементы.

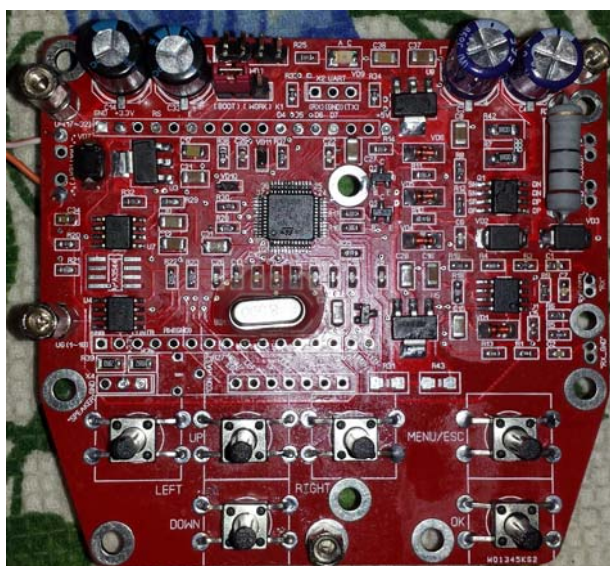


Рис. 4. Готовая плата металлодетектора

После полной сборки макетной платы прибора, установки дисплея, динамика был запрограммирован микроконтроллер и устройство успешно запустилось.

Заключение. В данной научной работе произведена разработка устройства металлодетектирования. В результате проектирования проведен анализ существующих аналогов и методов металлодетектирования, разработаны структурная и функциональная схемы, произведен выбор и обоснование элементов схемы устройства, разработана принципиальная схема устройства, исходный код программы микроконтроллера. Осуществлено конструктивное исполнение устройства и проверено на работоспособность. Результат подтвердил изначально поставленные гипотезы.

ЛИТЕРАТУРА

1. Ознакомительное руководство по ARM-микроконтроллерам Cortex-M3 [Электронный ресурс]. – Режим доступа: http://www.gaw.ru/html.cgi/txt/doc/micros/arm/cortex_arh/index.htm.
2. STM32 – Datasheet – STMicroelectronics [Электронный ресурс]. – Режим доступа: <http://www.alldatasheet.com/datasheet-pdf/pdf/352432/STMICROELECTRONICS/STM32-SK-IAR.html>.

УДК 004.021

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО ПЕРЕДВИЖЕНИЯ С ДОСТАВКОЙ ГРУЗОВ

Н.А. ГУРЕЦКИЙ

(Представлено: С.А. ОЛЬШАНИКОВ)

Представлен практический способ отыскания оптимального пути передвижения с доставкой грузов. Цель работы – написание программы для расчета наиболее выгодной с точки зрения времени и затрат на выполнение работ очередности посещения адресов. Задача решалась путем рекурсивного перебора всех возможных вариантов посещения адресов и нахождения оптимального маршрута передвижения. Программа написана на языке программирования C++ с использованием Win32 API.

В транспортной логистике, особенно для курьерских компаний, распространена задача определения оптимальной последовательности посещения адресов. Строительные компании также заинтересованы в определении наименьшего пути, который нужно пройти на стройке (перевезти груз, закопать ямы и т.п.). Причем, зачастую, переместить груз из одной точки в другую за один раз не получается и приходится возвращаться. Тогда любой путь и конечные пункты можно представить в виде матрицы чисел, где положительные элементы – это откуда нужно что-то взять и сколько, отрицательные – куда положить и сколько, нулевые – те, которым «ничего не нужно».

В реализации данной программы заложен алгоритм построения кратчайшего по расстоянию маршрута, проходящего через заданные пункты. Алгоритм основан на полном переборе возможных вариантов маршрутов.

Для выявления оптимального пути необходимо изначально собрать растровые данные. Растровые данные – двумерная сетка числовых значений, представляющих некоторую измеренную характеристику, где пространственные данные поверхности спроектированы на плоскость сетки. Значения в каждой дискретной точке сетки, называемой пикселями, обычно интерпретируются как среднее над областью, спроектированной на каждую точку сетки (рис. 1). Другие интерпретации, конечно, возможны в зависимости от источника поступления данных о поверхности (сенсор, процесс ассимиляции или другие). Процесс сбора растровых данных – процедура простая, интуитивно понятная. Поскольку, исключая дискретизацию, такое представление данных аналогично построению географических карт с использованием линий уровня.

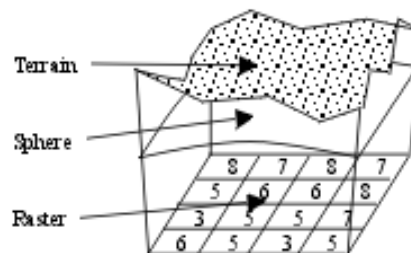


Рис. 4. Пример высот, представленных в виде растровых данных

На сегодняшний день известны несколько различных подходов к проблеме нахождения оптимальных путей через пространство. Мы попробуем составить дерево всевозможных решений. Подход, принятый в этой работе, должен упростить проблему поиска по дереву. Подобные, часто более простые версии этого подхода, используются разработчиками компьютерных игр для перемещения модулей по небольшим картам.

В самом начале перемещения для землеперемещающей машины область поверхности не имеет границ, то есть может перемещаться фактически бесконечно во всех направлениях. Ясно, что мы должны ограничить себя несколько меньшей областью, чтобы выполнить поиск пути на компьютере. Кроме того, мы хотели бы, чтобы эта область была как можно меньше без того, чтобы подвергать сомнению оптимальность пути. К тому же мы хотим уменьшить использование памяти. Ограничение области для движения машины будет иметь положительный эффект, связанный с уменьшением числа возможных путей, которые мы потенциально должны исследовать при поиске оптимального.

Дискретизируем бесконечное число позиций в непрерывном пространстве на конечное число точек. Чтобы облегчить представление, при дискретизации будем назначать эти точки в регулярной сетке. Область пространства, называемая ячейкой, закрывает каждую точку. Все ячейки не пересекаются и объединение всех ячеек должно закрыть все пространство. Движение нашего транспортного средства ограничено перемещением по «прямым линиям» между вершинами. Сетка может иметь любой вид, подходящий к заданному пространству и приложению при анализе. Два примера плоских сеток даны на рисунках 2 и 3.

Для нашего приложения использована регулярная «квадратная» плоская сетка. Это удобно, так как затем можно выбирать ячейки, которые точно совпадают с пикселями наших растровых данных.

Следующий шаг должен ограничить число возможных перемещений из данной ячейки. Если ячейка положительная, то мы ищем отрицательную ячейку, чтобы переместить груз. Иначе, если ячейка отрицательная, либо равна нулю, то мы просто перемещаемся в положительную ячейку. При этом каждый раз вычисляется расстояние до конечной клетки, которое помещается в результат.

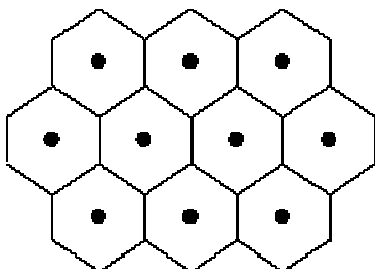


Рис. 5. Регулярная гексагональная плоская сетка

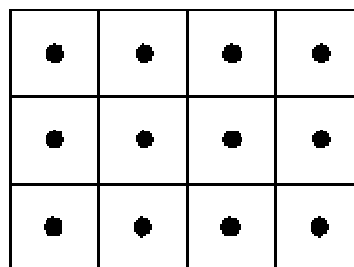


Рис. 6. Регулярная квадратная плоская сетка

Для определения оптимальности маршрута будем использовать следующие критерии:

1. Общая длина пройденного пути.
2. Длина, пройденная пустым (из отрицательной ячейки, либо ячейки равной нулю к положительной).
3. Длина груженым (из положительной ячейки в отрицательную).

В качестве языка программирования выбран язык C++, т.к. он является одним из самых быстрых, в плане вычислительных операций, языков.

Для составления дерева решений использована рекурсия. Конечно, все дерево решений не хранится, т.к. это огромные затраты оперативной памяти. Сохраняем лишь 20 наиболее оптимальных маршрутов. Стоит отметить, что при рекурсии идет огромное потребление оперативной памяти, но сокращается время работы. Поскольку всевозможных вариантов очень много (зависит от растровых данных), то целесообразно пустить эту функцию в отдельный поток, чтобы избежать зависания основного потока приложения.

На данный момент программа работает следующим образом: вводится размер матрицы и загружается текстовый файл с исходными растровыми данными. Растровые данные являются целыми числами. В результате работы программа выдает отчет, в котором сформирована наиболее оптимальная последовательность посещения адресов-ячеек. Оптимальный путь может быть и не один, поэтому пользователь может просмотреть отчет о 20 наиболее оптимальных маршрутах. Перемещения через точки можно наглядно отследить через интерфейс, написанный на Windows API [1],[2].

В результате данной работы решена задача определения оптимального маршрута при перевозке грузов и разработано актуальное программное обеспечение для ее решения, имеющее перспективы коммерческого использования. На данный момент ведутся работы по увеличению функционала программы.

ЛИТЕРАТУРА

1. Windows API. [Электронный ресурс] – Режим доступа https://ru.wikipedia.org/wiki/Windows_API . Дата обращения 10.04.2015.
2. Windows API. [Электронный ресурс] – Режим доступа <https://msdn.microsoft.com/en-us/library/cc433218>. Дата обращения 11.04.2015.

УДК 004.021

ОПТИМИЗАЦИЯ ПРОИЗВОДСТВЕННОЙ ЛОГИСТИКИ В СТРОИТЕЛЬСТВЕ НА ПРИМЕРЕ РАБОТЫ ЗЕМЛЕПЕРЕМЕЩАЮЩИХ МАШИН

Н.А. ГУРЕЦКИЙ

(Представлено: С.А. ОЛЬШАНИКОВ)

Затронута важная задача оптимизации при проектировании организационной части строительного производства. Понятно, что решение сложной оптимизационной задачи с множеством переменных невозможно без использования программных средств. Задача решена путем рекурсивного перебора всех возможных вариантов перемещения и нахождения оптимального маршрута. Если опти-

мальных маршрутов несколько, то можно выбрать любой из 20 предложенных, чтобы рабочие могли воспользоваться той, которая, по их мнению, будет наиболее удобной.

Строительное производство состоит из процессов, конечным результатом выполнения которых является строительная продукция, под которой следует подразумевать как отдельные части строящихся и реконструируемых объектов, так и законченные здания и сооружения. Строительная продукция включает в себя введенные в эксплуатацию промышленные предприятия и цехи, жилые дома, здания общественного назначения и другие, вновь построенные и реконструированные объекты.

Строительство является одной из важнейших и менее проработанных с точки зрения оптимизации отраслей промышленности. Земляные работы, в частности, работы по вертикальной планировке площадки, имеют высокие энергозатраты и составляют 15-20% от общей трудоемкости возведения объекта. Исходя из этого, оптимизация работ по вертикальной планировке площадки является важным вопросом, входящим в общий перечень оптимизации строительных процессов.

Планировку площадки производят при помощи землеремещающих механизмов. К землеремещающим механизмам относятся бульдозеры, грейдеры и скреперы. Несмотря на особенности работы каждого из механизмов, смысл планировки площадки состоит в перемещении объемов грунта из квадратов площадки, где имеется насыпь в квадраты с выемкой. Главный вопрос оптимизации работы землеремещающих машин состоит в том, чтобы определить в какой последовательности и из какого в какой квадрат необходимо перемещать грунт, что бы путь машины был минимальным.

Тогда любой путь и конечные пункты можно представить в виде матрицы чисел, где положительные элементы – это откуда нужно что-то взять и сколько, отрицательные – куда положить и сколько, нулевые – те, которым «ничего не нужно».

В реализации данной программы заложен алгоритм построения кратчайшего по расстоянию маршрута, проходящего через заданные пункты. Алгоритм основан на полном переборе возможных вариантов маршрутов и нахождением кратчайшего и оптимального пути. Если таких маршрутов несколько, то программа выдаст все эти маршруты, чтобы рабочие могли воспользоваться той, которая, по их мнению, будет наиболее удобной.

Интерфейс программы интуитивно понятен и прост в использовании (рис.1). Пользователь задает размер матрицы и указывает текстовый файл на компьютере с исходными данными.

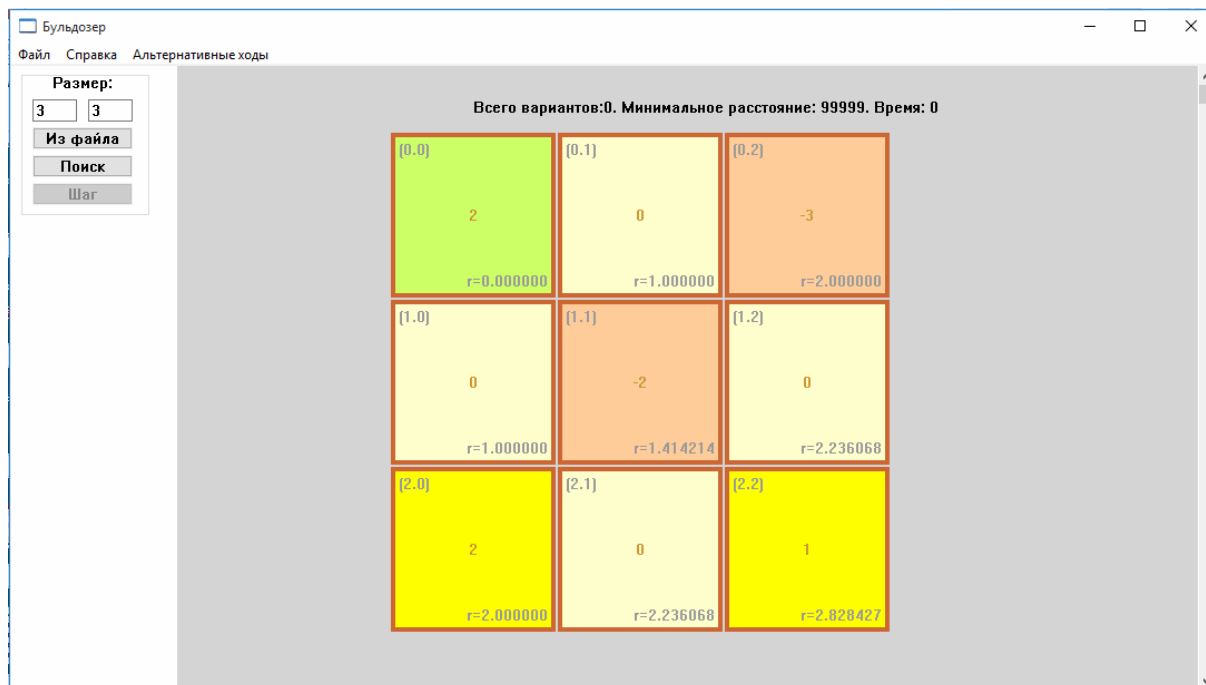


Рис. 1. Ввод данных

Далее пользователю необходимо лишь нажать кнопку «Поиск» и дождаться завершения подсчетов. В результате работы программа выдает список с наилучшими очередностями посещения адресов (рис. 3). Пользователь может проследить как меняются значения ячеек пошагово, нажав на кнопку «Шаг».

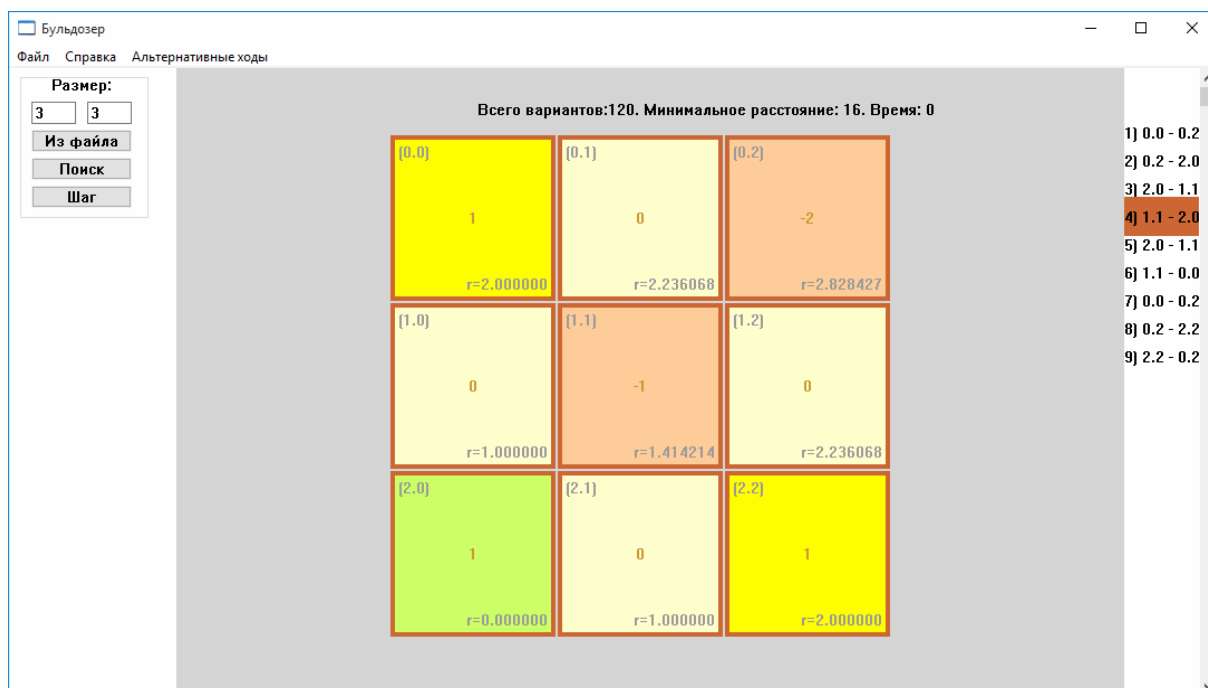


Рис. 2. Результаты работы программы и пошаговая инструкция

Может возникнуть ситуация, когда оптимальных путей несколько. Тогда пользователь сам может выбрать один из 20 маршрутов и также пошагово разобрать их (Рис. 3).

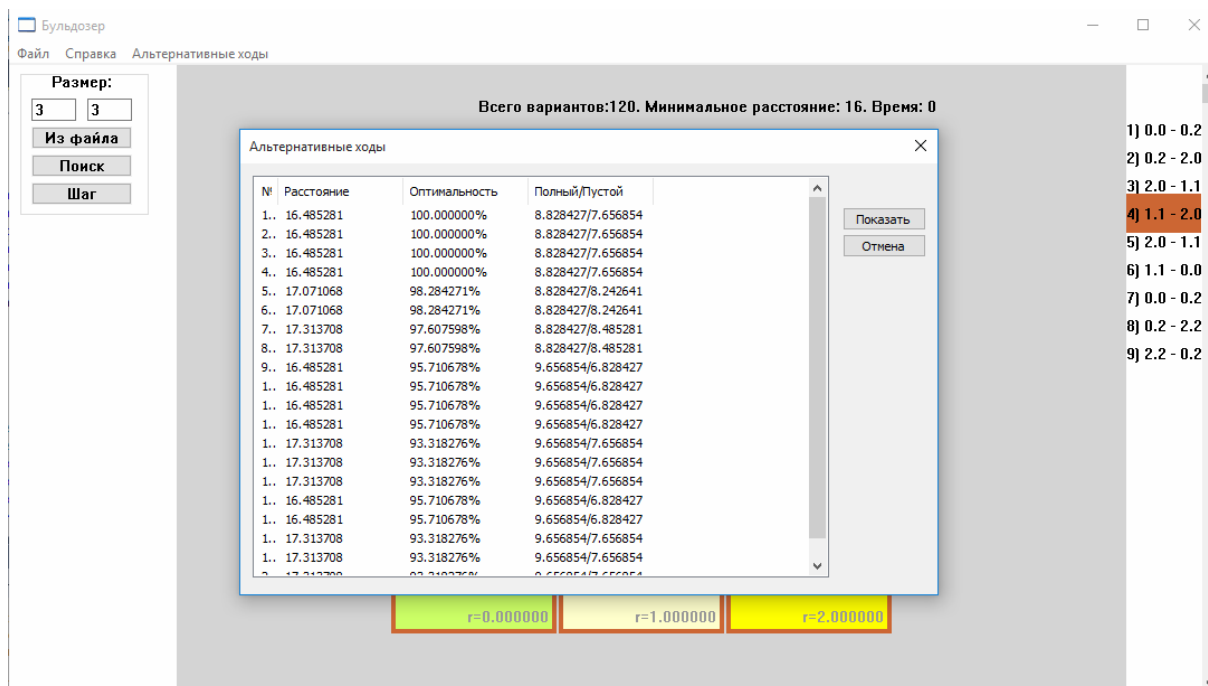


Рис. 3. Альтернативные маршруты

На данный момент ведутся работы по увеличению функционала программы, а также внедряется возможность получить данное ПО в индивидуальном порядке строительными компаниями и логистическими центрами.

ЛИТЕРАТУРА

1. Windows API [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/en-us/library/cc433218>. – Дата обращения: 11.04.2015.

УДК 004.658.001.63

**ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ
СИСТЕМЫ УПРАВЛЕНИЯ АФИШЕЙ КИНОТЕАТРОВ РЕСПУБЛИКИ БЕЛАРУСЬ****В.Ю. ДЕМЕНТЬЕВ***(Представлено: М.В. ДЕКАНОВА)*

Объектом разработки является система управления афишей кинотеатров Республики Беларусь. Цель работы – создание информационной системы для удобного управления афишей кинотеатров Республики Беларусь. Рассмотрены вопросы, связанные с процессом проектирования базы данных системы управления афишей кинотеатров Республики Беларусь.

Любое приложение, будь то web-приложение, desktop или мобильное, которое нацелено на предоставление актуальной информации, нуждается в администрировании, т.е. управлении контентом в удобной для администратора форме [4], используя понятный графический интерфейс. Обычно, для таких потребностей используются административные панели, которые позволяют обращаться к базе данных посредством web-форм, изменять содержимое приложений в визуальных редакторах и т.д. Поэтому, основное преимущество разрабатываемой системы управления в том, что администратору не нужно редактировать базу данных напрямую, а из-за необходимости хранения множества различных сущностей и большого объема данных, проектируемая база данных должна быть наиболее простой, нормализованной и эффективной.

Для определения сущностей, хранимых в базе данных, необходимо выделить функции, которые будут доступны в системе управления. В первую очередь стоит отметить, что система будет разделена на две части – глобальная система управления (для управления кинотеатрами, администраторами и т.п.) и система управления афишей кинотеатра (для непосредственного управления афишей кинотеатра). Во-вторых, из-за разделения системы будет выделено две роли – глобальный администратор и администратор кинотеатра. Таким образом, администратору кинотеатра будет доступен небольшой ряд функций по работе с афишей кинотеатра и редактированию информации о кинотеатре:

- добавление/редактирование/удаление фильмов;
- добавление/редактирование/клонирование/удаление цен;
- добавление/удаление расписания сеансов;
- редактирование информации о кинотеатре;

Глобальный администратор будет иметь намного больше прав:

- добавление/редактирование/удаление фильмов для любого кинотеатра;
- добавление/редактирование/клонирование/удаление цен для любого кинотеатра;
- добавление/удаление расписания сеансов для любого кинотеатра;
- редактирование информации о любом кинотеатре;
- редактирование информации любого администратора;
- добавление новых кинотеатров в систему;
- регистрация новых администраторов для кинотеатров;
- просмотр лога изменений любого администратора кинотеатра;
- «черный» список администраторов кинотеатров;
- просмотр различной статистики по кинотеатрам (количество фильмов, сеансов, цен и т.д.).

С учетом перечисленных функций, выделим следующие сущности проектируемой базы данных:

- администратор (user) – пользователь системы управления, обладающий определенными правами редактирования данных кинотеатра;
 - кинотеатр (cinema) – заведение, в котором проводятся показы фильмов;
 - город (city) – город, в котором располагается кинотеатр;
 - фильм (film) – продукт кинематографа, показы которого продает кинотеатр;
 - формат (format) – формат показа фильма (2D или 3D);
 - сеанс (session) – один показ фильма. Информация о сеансе включает в себя стоимость просмотра фильма, дату, время показа;
 - цена (cost) – стоимость одного сеанса;
 - архив (archive) – список фильмов, которые завершили свой показ в кинотеатре;
 - лог (log) – запись об изменении в системе, которая внесена администратором кинотеатра.

Чтобы построить схему реляционной базы данных необходимо определить совокупность отношений, которые составляют базу данных [2]. Важно, чтобы выполнялось условие целостности отношений: в любом отношении должны отсутствовать записи с одним и тем же значением первичного ключа, т.е. любая запись любого отношения должна быть отличной от любой другой записи этого отношения.

В проектируемой базе данных все таблицы будут иметь первичные ключи и, связаны между собой связью один ко многим [2].

Схема спроектированной базы данных, нормализованной до 3-й нормальной формы [2], представлена на рисунке.

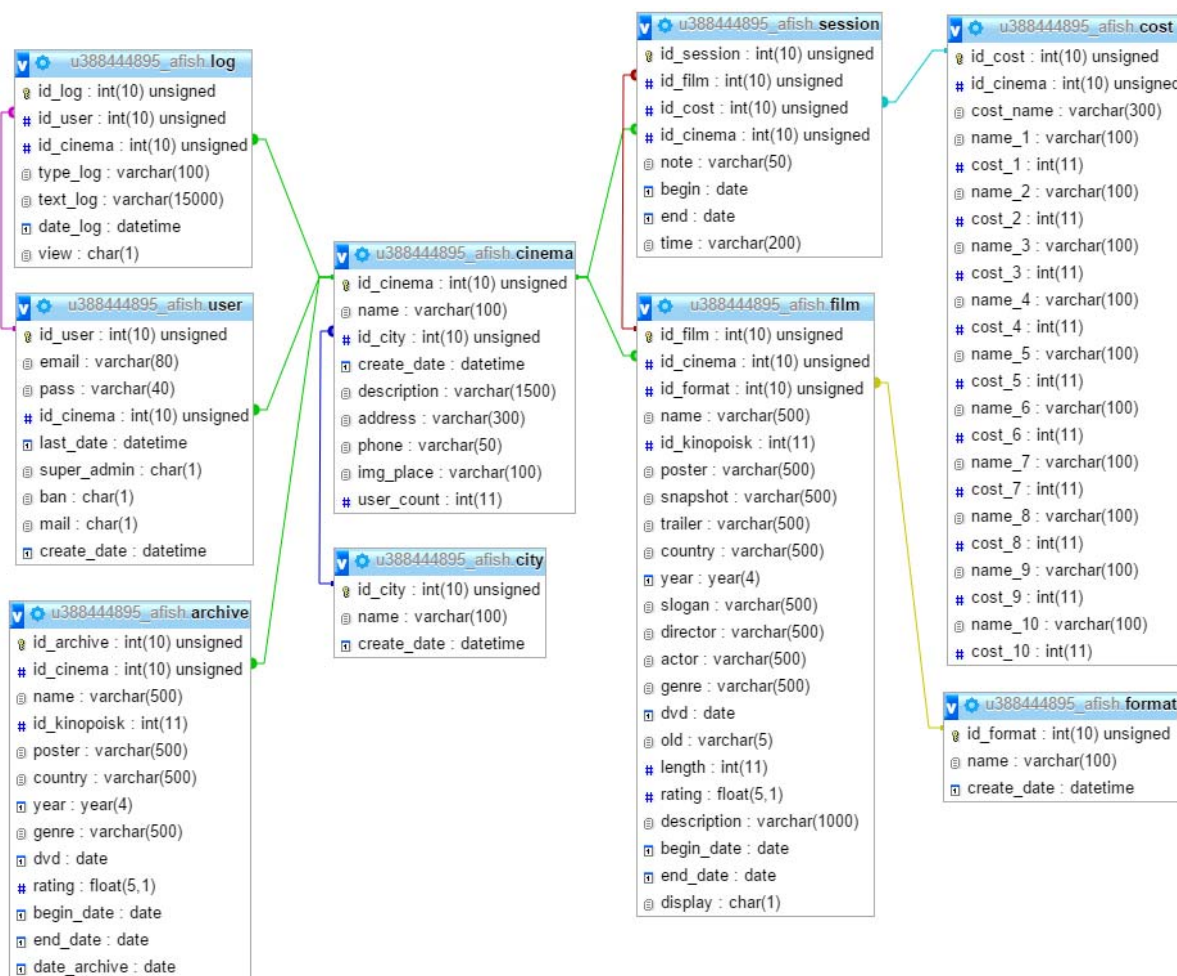


Рис. Схема спроектированной базы данных

В качестве СУБД была выбрана MySQL. СУБД MySQL обладает высокой скоростью работы по сравнению с другими системами управления. Также преимуществом является то, что это многопользовательская система, она не налагает ограничений на количество пользователей, одновременно работающих с базой данных [3].

MySQL может хранить огромное количество записей в своих таблицах. Этим обеспечивается возможность создания крупных проектов. Немаловажным показателем является также простая и эффективная система безопасности MySQL [1].

Для удобного формирования архива фильмов разработан триггер на удаление фильма из кинотеатра. Суть триггера заключается в том, что если в базу данных приходит запрос на удаления записи из таблицы фильмов, то автоматически будет выполнен запрос на добавление данной записи в таблицу архива, а уже после этого – удаление записи из таблицы фильмов.

В результате проделанной работы была спроектирована база данных, при помощи которой можно создать систему управления афишей кинотеатров для удобной работы администратора. Также использование базы данных делает удобным хранение и управление данными афиши.

ЛИТЕРАТУРА

1. Кузнецов, М.В. MySQL 5 / М.В. Кузнецов, И.В. Симдянов. – СПб. : БХВ-Петербург, 2010. – 1024 с.
2. Малыгина, М. Базы данных: основы, проектирование, использование / М. Малыгина. – М. : ВНУ, 2004. – 512 с.
3. Бейли, Л. Изучаем PHP и MySQL / Л. Бейли, М. Моррисон. – М. : Эксмо, 2012, – 800 с.
4. Плюсы и минусы CMS [Электронный ресурс] / Mywebblog.ru Блог Вебмастера – Режим доступа: <http://mywebblog.ru/sozдание/plyusy-i-minusy-cms.html>. – Дата доступа: 29.09.2015.

УДК 004.451.83

**ПРОЕКТИРОВАНИЕ КЛИЕНТСКОЙ ЧАСТИ СИСТЕМЫ УПРАВЛЕНИЯ
АФИШЕЙ КИНОТЕАТРОВ РЕСПУБЛИКИ БЕЛАРУСЬ****В.Ю. ДЕМЕНТЬЕВ***(Представлено: М.В. ДЕКАНОВА)*

Объектом разработки является система управления афишей кинотеатров Республики Беларусь. Цель работы – создание системы для удобного управления афишей кинотеатров Республики Беларусь. Рассмотрены вопросы, связанные с процессом проектирования клиентской части системы управления.

Целью разработки клиентской части системы управления является в первую очередь удобство работы администратора с записями из базы данных и исключение возможных ошибок при работе с базой данных. Следовательно, особое внимание необходимо уделить разработке графического пользовательского интерфейса.

Если рассматривать системы управления в целом, то можно выделить несколько минусов. Как правило все CMS разрабатываются под какие-то общие потребности (добавление статей, модерирование пользователей и др.) и возникают определенные трудности при необходимости добавления каких-либо специфических и уникальных для таких CMS модулей [4]. В результате чего вся система становится более громоздкой. Некоторые возможности системы управления не используются, т.к. они не нужны для текущих потребностей пользователя, но на них тратятся определенные ресурсы. Кроме этого стиль таких CMS недостаточно современен. На данный момент Google задает тенденцию в так называемом Material Design [3], схожим со стилем минимализм. В проектируемой клиентской части системы управления будет использован данный стиль, т.к. он все больше набирает популярность и выглядит достаточно строго и аккуратно. Нет необходимости в красочных изображениях, из-за которых пользователь может теряться в системе и не воспринимать или упускать важную информацию. Вся панель администратора должна быть выполнена в едином приятном для глаза стиле. Также стоит отметить, что многие системы управления не всегда удобны. Для выполнения основных задач бизнес-процесса необходимо выполнить неочевидные шаги для пользователя. Разрабатываемая система должна быть совершенно понятна и прозрачна, чтобы любой администратор кинотеатра мог интуитивно выполнить любое действие путем нажатия не более двух кнопок. Отдельное внимание будет уделено эргономике. Удобное размещение элементов также позволит облегчить работу администратора и снизить когнитивную нагрузку.

В качестве среды разработки был выбран JetBrains PhpStorm 8. Данная среда имеет отличный интерфейс и поддержку самых последних версий PHP. Кроме этого, PhpStorm предоставляет отличные возможности для удобной разработки на языках JavaScript, HTML и CSS [2].

За основу разработки дизайна была взята стилистика CSS библиотеки Skeleton [1]. Данная библиотека предоставляет минимальные возможности для начала разработки собственного проекта, поэтому все основные стили будут писаться собственноручно.

Для отображения иконок в приложении будет использоваться библиотека Font Awesome [1]. Это векторные иконки, которые интегрированы в шрифт. Их преимущество в том, что они способны изменять размер без потери качества, а также к ним можно применять различные CSS стили, например, изменение цвета.

Для удобной работы с JavaScript будет использоваться библиотека JQuery [2]. Она предоставляет простой доступ к DOM-элементам страницы. Большинство POST/GET запросов при работе с данными в панели управления будут обрабатываться посредством технологии AJAX [2]. Таким образом, это уменьшит время ожидания получения ответов от сервера, т.к. подгружаться на странице будут только необходимые данные, остальные данные затрагиваться не будут.

Все страницы будут сверстаны по единому шаблону: шапка – основной контент – подвал.

Прежде чем начать работу с системой управления, необходимо пройти процедуру авторизации. Это выполняется посредством формы входа, которая состоит из двух полей для ввода e-mail адреса и пароля, а также кнопки «Войти».

Далее будут описаны страницы доступные глобальному администратору.

В шапке сайта располагается меню, в котором находятся следующие пункты:

- Статистика;
- Управление;
- Лог изменений;
- Выход.

В свою очередь пункт меню «Управление» может быть раскрыт в виде выпадающего списка, который включает в себя следующие подпункты:

- управление кинотеатрами;
- управление пользователями;
- управление форматами фильмов;
- управление списком городов.

На странице статистики расположена различная информация по количеству сущностей в системе, статистика по кинотеатрам и т.п.

На странице «Лог изменений» расположен список изменений, которые могут внести в систему администраторы кинотеатров. Глобальный администратор сможет следить за всеми изменениями и принимать меры в случае нарушения каких-либо правил.

Страницы управления кинотеатрами и пользователями похожи между собой. Пример данной страницы приведен на рисунке 1.

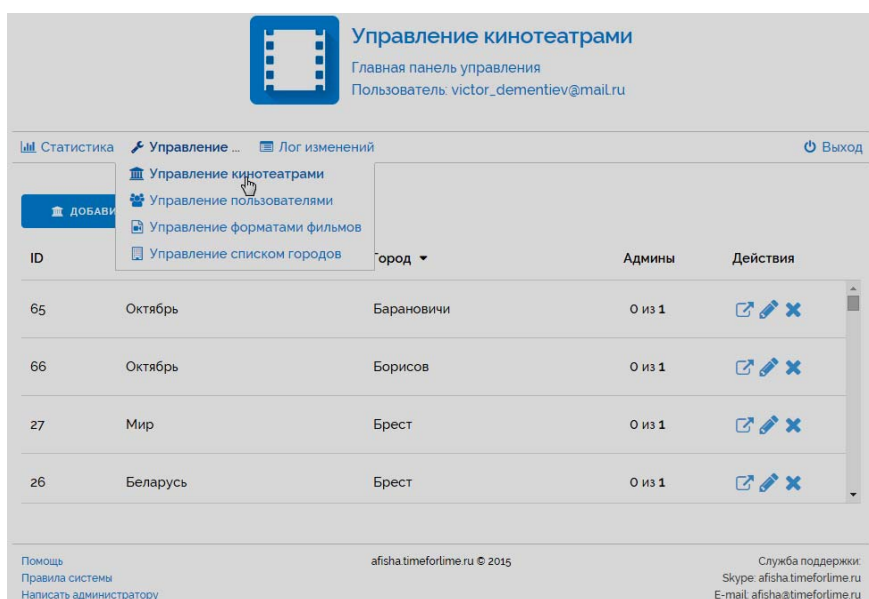


Рис. 1. Страница управления кинотеатрами

Данные страницы состоят из списка сущностей, некоторой информации о них и действий, которые для них возможны. Используя данную страницу, администратор может добавить новый кинотеатр в систему, отредактировать или удалить существующий, либо войти в любой из кинотеатров от лица администратора кинотеатра и выполнять любые действия доступные обычному администратору кинотеатра. На странице управления пользователь администратор также может добавлять новых администраторов для кинотеатров, удалять или редактировать существующих, добавлять их в «черный» список.

Страницы управления форматами фильмов и списком городов также похожи. На них отображается список сущностей, с возможностью редактирования и удаления, а также дополнительное поле для быстрого добавления новой сущности. Все действия выполняются по средствам технологии AJAX [2].

Так как у администратора кинотеатра другие функции, у него в панели управления располагаются иные пункты меню:

- Афиша;
- Добавить;
- Архив;
- В панель управления;
- Настройки;
- Выход.

При этом пункт меню «Добавить» включает в себя следующие подпункты, которые раскрываются в виде выпадающего списка:

- Добавить фильм;
- Добавить сеансы;
- Добавить цены.

На странице «Афиша» располагается список текущих фильмов, которые добавлены в кинотеатре, а также его сеансы и возможные действия. На данной странице администратор может сделать фильм невидимым для конечных пользователей, перейти к редактированию фильма или удалить его. После удаления фильм помещается в архив и весь список удаленных фильмов доступен на соответствующей странице. В архиве администратор может удалить фильм полностью из системы.

Пункт меню «В панель управления» отображается только, если в кинотеатр вошел глобальный администратор через страницу «Управление кинотеатрами». Она возвращает администратора обратно к списку кинотеатров. Для обычного администратора данный пункт недоступен.

На странице настроек администратор может изменять данные о кинотеатре – телефон, адрес или описание; сменить пароль.

Страницы добавления и редактирования фильма очень похожи и состоят из одинаковых полей. На них есть возможность загрузки постера фильма и выбора формата фильма. Кроме этого на данных страницах указываются стандартные для фильма данные – название, страна, год, описание и т.п.

На странице добавления цен располагается список уже добавленных цен в кинотеатре и форма для добавления в систему новой цены. При этом цены могут быть составные, т.е. одна запись может содержать до 10 цен. Таким образом, если в кинотеатре существует разделение зала на сектора, с разной стоимостью, администратор в пределах одной записи может указать стоимость каждого сектора. На той же странице администратор может удалить, отредактировать или создать клон записи о стоимости.

На странице добавления сеансов располагаются выпадающие списки с доступными фильмами и ценами, которые необходимо выбрать для добавления сеанса. Там же необходимо указать период и время сеанса.

Кроме описанных пунктов меню, в системе в подвале каждой страницы доступны следующие пункты меню:

- Помощь;
- Правила системы;
- Написать администратору.

На странице помощи располагается список возможных проблем и методы их решения в виде видеозаписи, а также примеры использования всех доступных администратору кинотеатра функций. На странице правил системы располагается список правил, из-за невыполнения которых администратор может попасть в «черный» список системы. А при нажатии ссылки «Написать администратору» пользователю отобразится всплывающее окно с формой для отправки сообщения администратору.

Внешний вид страницы «Афиша» приведен на рисунке 2.

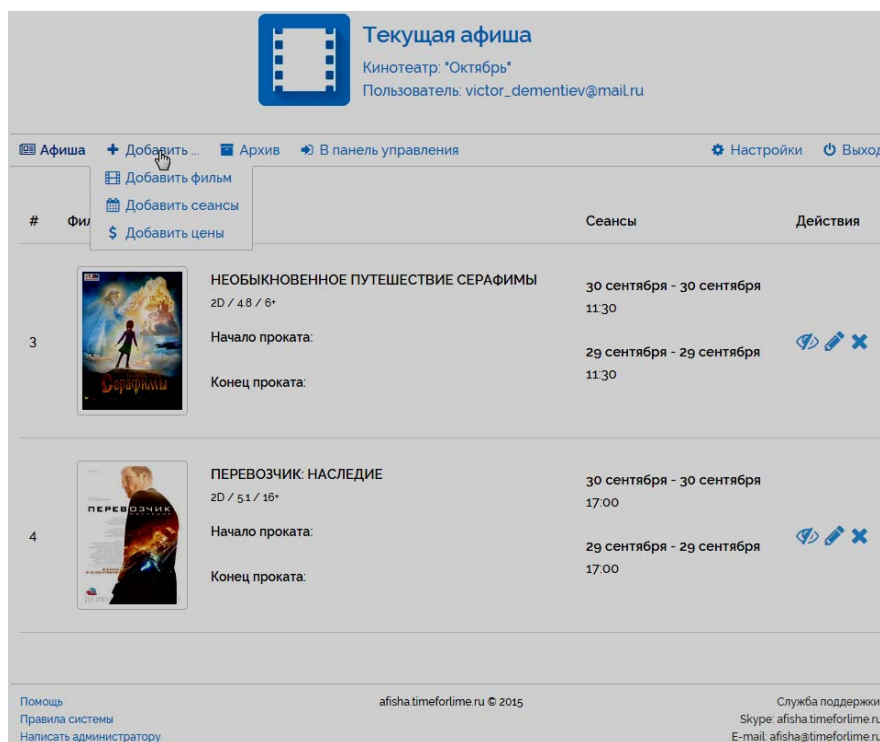


Рис. 2. Страница управления кинотеатрами

Таким образом, спроектирована система управления афишей кинотеатров Республики Беларусь, которая позволит облегчить процесс управления афишей кинотеатра. Разработан интерфейс, который является понятным пользователю без обучения работы с системой. Присутствует очень подробная страница помощи. В случае выполнения неверных действий пользователю будет показано сообщение об ошибке, а в случае корректного выполнения – сообщение об успешном завершении операции.

ЛИТЕРАТУРА

1. Макфарланд, Д. Большая книга CSS3 / Д. Макфарланд. – 3-е изд. – СПб. : Питер, 2014 – 608 с.
2. Бенедетти, Р. Изучаем работу с JQuery / Р. Бенедетти, Р. Крэнли. – СПб. : Питер, 2012. – 512 с.
3. Спецификация Google – Material design [Электронный ресурс] / Google. Material design. – Режим доступа: <http://www.google.com/design/spec/>. – Дата доступа: 29.09.2015.
4. Плюсы и минусы CMS [Электронный ресурс] / Mywebblog.ru Блог Вебмастера. – Режим доступа: <http://mywebblog.ru/sozдание/plyusy-i-minusy-cms.html>. – Дата доступа: 29.09.2015.

УДК 005

ВНЕДРЕНИЕ МЕТОДОЛОГИИ DEV OPS. ВИРТУАЛИЗАЦИЯ ОКРУЖЕНИЯ РАЗРАБОТЧИКА

Е.П. ЖИДЕЦКИЙ

(Представлено: Е.Р. СУХАРЕВ)

В рамках внедрения методологии DevOps в нашей компании был проведен ряд организационных изменений, пересмотрены зоны ответственности работников технического сектора, а также внедрены новые инструменты и методы в процесс создания и сопровождения программных продуктов. Одним из таких методов стала виртуализация окружения разработчика с использованием инструментов Vagrant и Ansible.

Термином «DevOps» обычно называют возникшее профессиональное движение, которое выступает за совместные рабочие отношения между разработчиками и ИТ-подразделением, в результате получая более быстрое выполнение планируемых работ, одновременно увеличивая надежность, стабильность, устойчивость и безопасность production-среды [1]. DevOps закономерно наследует наработки гибких методологий и определяет ряд принципов, общий смысл которых сводится к стиранию границ между разработчиками, тестировщиками, администраторами, специалистами по контролю качества и другими представителями операционного сектора, а также мотивации к реализации дерзких рискованных затей и получении максимально быстрой обратной связи от пользователей.

Для достижения поставленных целей требуется максимальная автоматизация рабочего процесса: тестирование, непрерывная интеграция, непрерывная доставка обновлений пользователям, проверка качества кода, развертывание production-окружений, мониторинг. Для реализации всех этих процессов существует целая масса готовых решений, обладающих своими плюсами и минусами. В соответствии с концепцией DevOps, зоны ответственности между различными отделами технического сектора компании размываются, что при выборе инструмента выводит критерий простоты освоения на единый уровень с функциональностью: DevOps-профессионал должен иметь возможность и уметь одинаково эффективно писать код, писать тесты к нему, разворачивать у себя окружение аналогичное тому, которое будет на production-стенде, изменять настройки этого окружения, постоянно находясь под присмотром автоматических средств контроля качества кода и покрытия тестами, и при этом превентивная годовая подготовка в специальном центре не предусматривается.

Проблема различных окружений у разработчика, тестировщика и на production-стенде известна всем. Еще лучше известны ее последствия: невозпроизводимые ошибки, фраза «Не знаю, у меня все работает», неожиданные падения production-стендов при отличной жизнеспособности сервиса под такими же нагрузками на тестовых площадках и др. Элементарно создание идентичных конфигураций на 5–20 машинах, что в наш век облачных сервисов даже не среднее количество, без автоматизации процесса превращается в работу, посылную только прожженному администратору старой закалки.

На самом деле решение всех этих проблем уже давно живет рядом с нами, и имя ему – виртуальная машина. Давно не секрет, что сети основной массы средних и крупных компаний целиком построены

на виртуальных машинах. У этого подхода есть масса преимуществ, основные из которых это цена и гибкость. С использованием виртуальных машин становится проще распределять ресурсы для различных узлов сети и контролировать уровни доступа, разворачивать новые узлы, клонируя виртуальные образы существующих машин.

В окружении разработчика виртуализация нашла свое применение сравнительно недавно. Да и сейчас, несмотря на массу известных положительных моментов, опытные разработчики не спешат перестраивать свой привычный подход к работе. Тем не менее, прогрессивные представители нашей профессии уже разработали несколько инструментов для упрощения работы с виртуальными машинами и ряд рекомендаций по их использованию, проверенных в «боевых» условиях.

Наиболее распространенным средством для управления виртуализацией окружения разработчика является Vagrant. Vagrant – это менеджер виртуальных машин, работает с VirtualBox, VmWare и несколькими другими менее популярными системами виртуализации [2]. Обладает простым консольным интерфейсом, и главное – работает по сценарию, описанному в конфигурационном файле в синтаксисе Ruby. Это значит, что один DevOps-специалист может написать сценарий на этом довольно распространенном языке программирования, поместить файл со сценарием под управление системы контроля версий, и вся команда разработчиков, работающая с ним над проектом, сможет ощутить на себе всю прелесть виртуализации, выполнив всего одну простую консольную команду: *vagrant up*. Если разработчик может поднять у себя виртуальную машину с окружением для разрабатываемого проекта, то он может поднять эту же машину и на тестовом стенде и на production-сервере, а это уже шаг от простого разработчика к DevOps-специалисту. Объективно, такой подход дает следующие преимущества:

- повторяемость окружения. У программиста, у тестировщиков и на production-стенде все одинаково;

- чистота host-машины. Работа на различных проектах требует различных окружений, если все ставить прямо на свою машину, то скоро там будет «зоопарк» из различных СУБД, версий интерпретаторов, сервисов и серверов. При использовании виртуальных машин, на host-машине остаются только любимая IDE и несколько легких UI-клиентов к сервисам, работающим на виртуальных машинах;

- простота подключения новых людей к проекту. В больших проектах всегда множество внешних зависимостей, новый человек сможет начать работать и видеть результаты своих действий, не изучая множество документации по установке и инструкций в wiki-проекта по конфигурированию экосистемы;

- простота переключения между проектами. Переключение между проектами сводится к следующей последовательности действий:

- 1) *vagrant suspend* в текущем проекте;
- 2) открытие нового проекта в IDE;
- 3) *vagrant up* в новом проекте;

- легко можно передать от тестировщика разработчику на исправление трудно воспроизводимую ошибку, просто заморозив виртуальную машину на моменте с ошибкой и передав образ.

Виртуальная машина при работе с Vagrant представляет собой некий начальный образ (box в терминах Vagrant), который разворачивается и конфигурируется. Большая коллекция базовых образов систем доступна в репозитории Vagrant [3]. Конфигурирование представлено в самом широком смысле этого слова: пробросы портов, настройка общих ресурсов, установка и настройка приложений (provisioning в терминах Vagrant). Установка и настройка приложений может выполняться как на низком уровне: исполнение shell-скриптов – так и с применением инструментов управления конфигурациями: Chef, Puppet, Ansible.

В нашей компании мы выбрали Ansible, как самый простой и достаточно функциональный инструмент управления конфигурациями [4]. Ansible обладает практически нулевым порогом вхождения для программиста. Все управление целевым сервером, или группой серверов, осуществляется через SSH протокол, не требуется открытие дополнительных портов и установка дополнительного программного обеспечения на обслуживаемых серверах.

Сценарии в Ansible описываются на языке YAML с использованием синтаксиса выражений шаблонизатора Jinja2. В сценарии (playbook в терминах Ansible) указывается конфигурация и множество серверов, к которым она применяется. Сами сервера представлены в так называемых inventory-файлах: файлы в ini-формате в которых указаны IP или DNS-имена серверов с группировкой по предназначению: сервера баз данных, сервера приложений и т.д. Как правило на каждое окружение: тестовые, production – создается свой inventory-файл. Конфигурация состоит из списка команд, за выполнение каждой команды отвечает конкретный модуль со своим списком параметров. Технически, модуль в Ansible – это одна команда в shell-оболочке. Основное различие заключается в том, что сценарий Ansible обладает свойством идемпотентности: если ранее конкретное изменение применялось, то оно не будет применено повторно –

а это значит, что можно спокойно вносить улучшения в сценарии и раз за разом запускать их на одном и том же множестве машин. Ansible «из коробки» поддерживает множество модулей, есть возможность писать собственные модули на языке Python. Чем выше версия Ansible, тем больше модулей он поддерживает, по этой причине рекомендуется всегда работать на последней стабильной версии.

Ansible позволяет писать сценарии в одном `playbook`-файле, в таком случае файл сценария представляет собой последовательность модулей, построенную по аналогии с обычным `shell`-скриптом. Сценарии, написанные таким образом, имеют большой размер, трудны в поддержке и абсолютно не пригодны для повторного использования. Есть еще более простой способ исполнения команды на требуемом множестве серверов – `AdHoc` [5]. Данный способ вообще не требует файла сценария, и может выполнить команду на серверах, содержащихся в `inventory`-файле, на лету из командной строки. Этот стиль является категорически плохим тоном и не рекомендуется к применению в «боевом» окружении, поскольку позволяет «сгоряча» накатить на множество серверов невоспроизводимое изменение. Любое изменение, не зафиксированное в сценарии под системой контроля версий, считается не воспроизводимым, поскольку не будет гарантированно применено к серверам, добавленным в систему в будущем, а значит, приведет к рассинхронизации системы и появлению потенциальных проблем поддержки.

В соответствии с лучшими практиками рекомендуется разделять наборы изменений на роли [6]. Под этим термином в Ansible, крайне не интуитивно, подразумевается некоторое установленное и настроенное средство: `JRE`, `nginx`-сервер, `Tomcat`, `MongoDB` и др. Создание роли предполагает описание процесса установки и настройки конкретного средства, определение множества возможных параметров, оказывающих влияние на процесс, определение зависимостей от других ролей. Формат описания роли строго регламентирован, используется определенная соглашением структура файловой системы.

Для распространения ролей существует специальный репозиторий `Ansible Galaxy` [7]. Стоит заметить, что модерация публикуемых в этом репозитории ролей оставляет желать лучшего. Существует система пользовательских отзывов, но и она не дает гарантии качества. Кроме этого, Ansible не является кросс-дистрибутивным средством, что создает дополнительные трудности в поиске готовой роли для применения в собственной среде. Методом проб и ошибок, мы пришли к собственноручной реализации ролей с минимально требуемой для повторного использования конфигурируемостью и распространению их через корпоративный репозиторий.

После освоения подхода по оформлению конфигураций в роли, файл сценария превращается в простой список ролей и серверов, к которым их требуется применить. При этом конкретные сервера для каждого окружения задаются в собственном `inventory`-файле. Конфигурацию такого формата легко расширять и поддерживать, она становится похожа на хороший объектно-ориентированный код.

В настоящее время практически во всех проектах нашей компании внедрена система виртуализации окружения разработчика с применением описанных выше инструментов и подходов. В результате этого, из оборота практически исчезли инструкции для новых разработчиков, а среднее время начала работы над новым проектом сократилось с 26 до 3 часов. Степень соответствия рабочего процесса методологии `DevOps` включает в себя множество параметров и почти не поддается объективной оценке, но положительная динамика очевидна, а значит, выбран правильный ориентир для развития.

ЛИТЕРАТУРА

1. Шарма, С. Освоение концепции `DevOps` в интересах непрерывной поддержки инноваций / С. Шарма // `IBM developerWorks` [Электронный ресурс]. – 2014. – Режим доступа: <https://www.ibm.com/developerworks/ru/library/d-devops-continuous-innovation>. – Дата доступа: 25.09.2015.
2. `Providers` [Electronic resource] / `Vagrant docs`. – 2015. – Mode of access: <https://docs.vagrantup.com/v2/providers>. – Date of access: 25.09.2015.
3. `Vagrant boxes index` [Electronic resource] / `Vagrantbox.es`. – 2015. – Mode of access: <http://www.vagrantbox.es>. – Date of access: 25.09.2015.
4. Поляков, В. Ansible / В. Поляков // `Habrahabr` [Электронный ресурс]. – 2013. – Режим доступа: <http://habrahabr.ru/post/195048>. – Дата доступа: 26.09.2015.
5. `Introduction To Ad-Hoc Commands` [Electronic resource] / `Ansible documentation`. – 2015. – Mode of access: http://docs.ansible.com/ansible/intro_adhoc.html. – Date of access: 26.09.2015.
6. `Best Practices` [Electronic resource] / `Ansible documentation`. – 2015. – Mode of access: http://docs.ansible.com/ansible/playbooks_best_practices.html. – Date of access: 26.09.2015.
7. `Welcome to Ansible Galaxy` [Electronic resource] / `Ansible Galaxy`. – 2015. – Mode of access: <https://galaxy.ansible.com/intro>. – Date of access: 26.09.2015.

УДК 004.652.4

**ХРАНЕНИЕ ЭЛЕКТРОННЫХ ПАСПОРТОВ ТОВАРОВ
В РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ С ВЕРСИОНИРОВАНИЕМ ИЗМЕНЕНИЙ****Е.П. ЖИДЕЦКИЙ***(Представлено: Е.Р. СУХАРЕВ)*

Реляционные базы данных по определению плохо подходят для хранения данных с изменяющейся структурой, но иногда такая необходимость возникает. Существуют различные методы решения этой задачи в зависимости от дополнительных требований. Рассмотрен один из вариантов ее решения.

Для хранения данных с изменяющейся структурой в большинстве случаев лучшим решением будет использование документно-ориентированной базы данных [1]. Такой подход упростит логику DAO-слоя приложения, даст высокую скорость записи и чтения данных. Но иногда характер сохраняемой информации, а также бизнес логика работы с ней требуют реляционного подхода.

Примером такой модели данных может быть информация о товарах для системы управления мастер-данными. В таких системах особенно важным является удобство заполнения каталогов и редактирования отдельных товаров, а также скорость доступа к отдельным атрибутам конкретного товара. Кроме того для различных пользователей структура товара может различаться, это обусловлено различными внешними и внутренними факторами: требования рынка, государственный стандарт электронного паспорта товара, характер внутренних бизнес процессов компании и др. Для описания информации о товаре существует единый мировой стандарт GS1, включающий в себя более 500 атрибутов – что является избыточным для большинства пользователей [2]. Дополнительным требованием было хранение истории изменений товаров для возможности определения момента и автора изменения любого атрибута товара, а также отката изменений товара до определенного момента. Эти требования делали модель реляционной и ограничивали множество кандидатов на роль хранилища данных.

Для анализа модели данных использовался стандарт GS1, который давал информацию о максимально детальном описании товара, за исключением атрибутов, используемых во внутренних бизнес процессах компаний пользователей. В ходе анализа было определено множество типов данных, покрывающее все атрибуты:

- число. Обычная числовая характеристика: процентное содержание алкоголя, штрих код, остаток на складе;
- строка. Обычная строковая характеристика: код GTIN, GLN владельца;
- дата. Обычная календарная характеристика: дата создания, дата отгрузки, дата окончания срока годности;
- логическое да/нет. Различные флажки: подлежит ли возврату, указана ли цена на упаковке, подлежит ли переработке;
- значение из словаря. Выбор из списка вариантов: страна производитель, тип штрих кода, канал торговли;
- многоязычная строка. Строковая характеристика, представленная на разных языках: наименование на упаковке, описание, рекомендации;
- число с единицей измерения. Физические величины: метрики, цена с указанием валюты;
- файл. Различные присоединенные файлы: изображение товара, сертификат.

Каждый атрибут, кроме логических, может быть представлен как единичным значением, так и массивом значений соответствующего типа данных.

Для получения возможности хранить товары с гибкой структурой атрибутов при минимальной избыточности было решено построить реляционное хранилище по следующим принципам:

- отдельные таблицы для атрибутов каждого типа данных: ссылка на товар, имя атрибута, индекс, значение (в зависимости от типа данных);
- таблица для ключевых данных товара: версия, автор, хеш (высчитывается из ключевых атрибутов) для быстрого поиска;
- отдельные таблицы для сохранения истории атрибутов каждого типа данных;
- таблица для сохранения истории товаров, для группировки истории атрибутов.

Такая структура базы данных позволила убрать влияние размера истории изменений товара на скорость доступа к его актуальному состоянию, поместив данные об истории изменений в отдельный набор таблиц, а при использовании стратегии вертикального шардинга – даже на отдельный сервер [3]. Также в структуре было зафиксировано максимальное количество статичной информации о модели дан-

ных без потери гибкости в добавлении в модель новых атрибутов. Схема базы данных представлена на рисунке.

Для работы с данными был разработан ряд алгоритмов, позволяющий выполнять базовые операции максимально эффективно. Операции чтения, создания и удаления имели только одну особенность: необходимость работы с несколькими таблицами для доступа к данным по всем требуемым типам атрибутов – что не было проблемой благодаря использованию механизма транзакций и пакетного исполнения запросов JDBC [4].

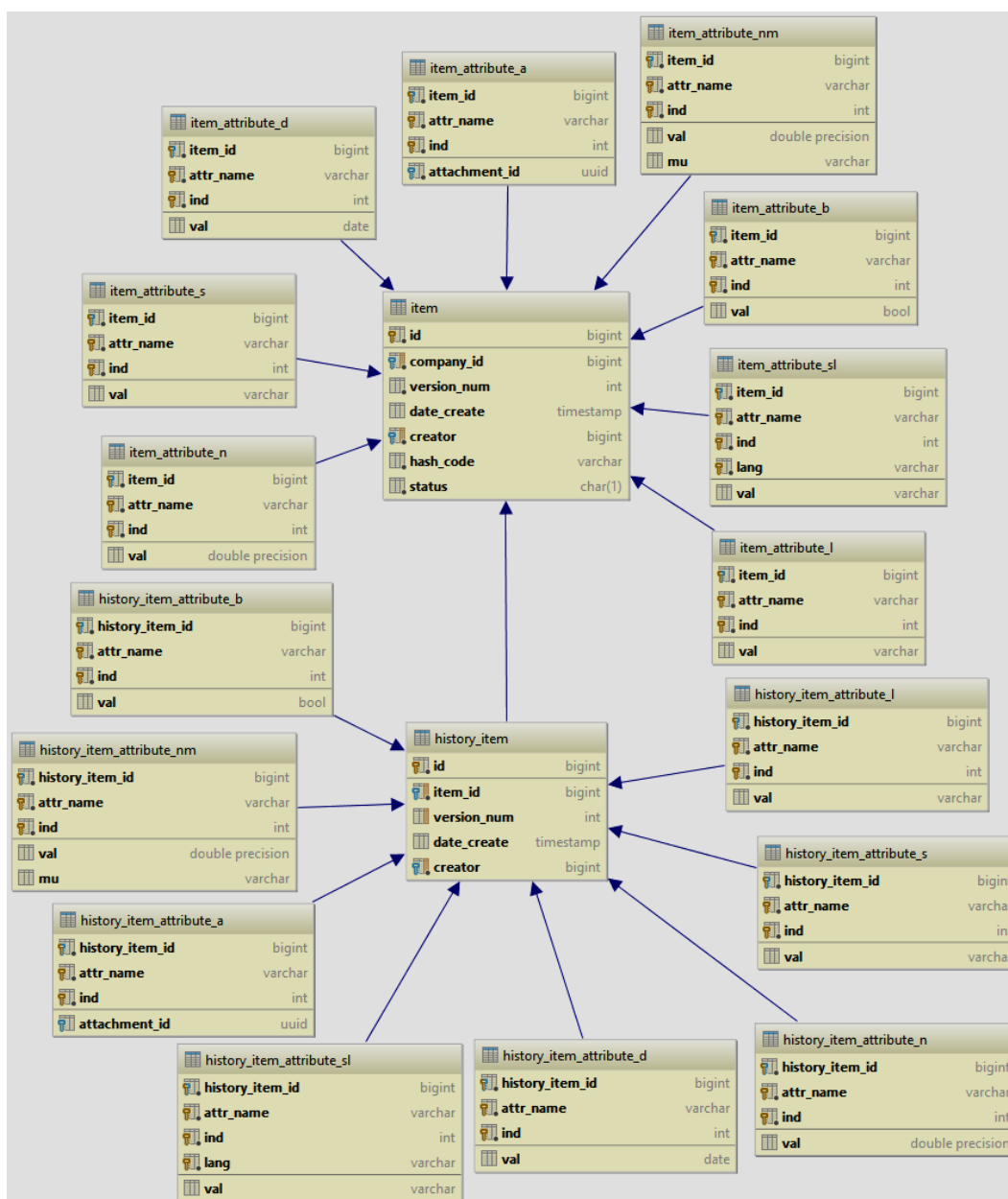


Рис. Схема базы данных с информацией о товарах и версиях изменений

Наиболее сложной операцией было обновление данных, поскольку только в этой операции требовалось вести версию изменений. Классические алгоритмы ведения разностных изменений не подходили, поскольку они рассчитаны на накопление изменений от некоторого фиксированного базового состояния. При таком подходе существует прямая зависимость между вычислительной сложностью получения актуального состояния товара и длиной истории его изменений, а значит, чем больше было изменений в товаре, тем дольше будет выполняться доступ к его актуальному состоянию. Кроме этого, при классическом подходе урезание истории редактирования до определенной глубины является довольно сложной операцией, в которой изменения в не актуальной части истории перед удалением последова-

тельно накатываются на базовое состояние и меняют его. Эти факты привели к необходимости разработки собственного «обратного» алгоритма сохранения изменений. Его ключевые особенности:

- в базе всегда хранится актуальная версия товара;
- в актуальной версии товара не сохраняются атрибуты с пустым значением;
- восстановление ранней версии товара представляет собой итерационное применение изменений от актуальной версии до требуемой;
- вычислительная сложность получения ранней версии прямо пропорциональна количеству версий между актуальной и требуемой;
- урезание истории выполняется путем простого удаления данных об устаревших изменениях.

Перед обновлением данных загружается текущее состояние товара из базы. Это необходимо для формирования разностных данных между существующим состоянием и новым, а также предотвращает множество проблем при одновременном редактировании товара разными пользователями: позволяет определить конфликт версий и провести слияние. В историю сохраняются полученные разностные данные в инвертированном виде, что позволяет, применяя сохраненные в базе изменения, переходить от текущего состояния на предыдущее. Для каждого изменившегося значения атрибута, который однозначно идентифицируется в товаре при помощи составного ключа из имени атрибута и индекса, применяется алгоритм в зависимости от характера изменения атрибута. Алгоритм включает в себя изменение атрибута в актуальном состоянии, и в истории предыдущей версии товара. Возможные варианты представлены в таблице.

Таблица

Работа с хранилищем при обновлении товара в зависимости от характера изменения атрибута

Характер изменения	Признак	Действие	
		Актуальная версия	Предыдущая версия
Добавление	В текущей версии товара у атрибута отсутствовало значение, а в новой – присутствует	Вставить новое значение	Вставить NULL
Удаление	В текущей версии товара у атрибута присутствовало значение, а в новой – отсутствует	Удалить значение	Вставить старое значение
Изменение	Значение атрибута присутствует в текущей и новой версии товара, но оно различно	Изменить на новое значение	Вставить старое значение

С целью оптимизации работы с базой, описанный выше алгоритм был несколько усложнен: изменения атрибутов конкретного товара группировались по изменяемым таблицам. Это позволило применить пакетную обработку запросов JDBC и увеличить производительность операций [4].

При нагрузочном тестировании, в базу было внесено более 9 000 000 товаров по 30 заполненным атрибутам в каждом. Шардинг базы данных не применялся. В качестве СУБД выступал PostgreSQL 9.3, развернутый на машине средней мощности: Intel Core i5 2,2GHz 4 CPU, 8 GB RAM – и настроенный в соответствии с лучшими практиками [5]. Размер хранилища данных составил 40 Гб. Средняя скорость выборки по ключевым атрибутам – 78 запросов/сек, по не ключевым – 11 запросов/секунду. Т.о. система, с хранилищем, построенным на описанных выше принципах, показала высокую производительность при нагрузках, а также надежность при одновременной работе множества пользователей с одним и тем же набором товаров.

Описанный в статье подход к хранению и работе с данными динамичной структуры в реляционном хранилище показал свою эффективность на тестовом стенде и в настоящее время работает в production-окружении. Он может применяться для хранения объектов из широкого спектра предметных областей, где характер данных не позволяет работать в документно-ориентированной среде.

ЛИТЕРАТУРА

1. Seguin, K. Когда использовать MongoDB / K. Seguin // Маленькая книга о MongoDB [Электронный ресурс]. – 2015. – Режим доступа: <http://jsman.ru/mongo-book/Glava-5-Kogda-ispolzovat-MongoDB.html>. – Дата доступа: 27.09.2015.
2. Standards [Electronic resource] / GS1 The Global Language of Business. – 2015. – Mode of access: <http://www.gs1.org/standards>. – Date of access: 27.09.2015.
3. Шардинг и репликация [Электронный ресурс] / HighLoad.com. – 2009. – Режим доступа: <http://ruhighload.com/index.php/2009/05/06/шардинг-партиционирование-репликац>. – Дата доступа: 28.09.2015.
4. JDBC – Batch Processing [Electronic resource] / Tutorialspoint. – 2015. – Mode of access: <http://www.tutorialspoint.com/jdbc/jdbc-batch-processing.htm>. – Date of access: 28.09.2015.
5. Installation and Administration Best practices [Electronic resource] / PostgreSQL Wiki. – 2015. – Mode of access: http://wiki.postgresql.org/wiki/Installation_and_Administration_Best_practices. – Date of access: 28.09.2015.

УДК 004.051

ЦЕЛЕСООБРАЗНОСТЬ РАЗРАБОТКИ ПРИЛОЖЕНИЙ ДЛЯ ПЛАТФОРМЫ ANDROID

М.А. ИВАНОВ

(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

Рассмотрена целесообразность разработки приложений для платформы Android, а также возможность заработка на приложениях.

Введение. Мобильные телефоны, а сейчас уже и смартфоны настолько прочно вошли в повседневную жизнь, что сейчас уже сложно представить, как можно прожить неделю без интернета или телефона. Наверное, есть люди, не пользующиеся этим, но таких людей подавляющее меньшинство.

Разработчики программ и приложений для смартфонов, хорошо понимая существующую тенденцию, разрабатывают огромное количество программных средств.

Но существует множество операционных систем и их версий. Приложение, написанное для одной системы, не будет работать на другой, если не предусмотреть совместимость с ней. Поэтому, как правило, приложение пишется для какой-то определенной платформы, и, если оно приобрело популярность, портируется на остальные.

Поэтому есть смысл изучить популярность платформ и их версий перед тем, как заниматься разработкой, а также ознакомиться с возможными способами заработка от приложений.

Основной раздел. Согласно данным сайта www.3dnews.ru [3] по итогам 2014 года в целом поставки смартфонов достигли 1,30 млрд штук против 1,02 млрд в 2013-м. На Android-устройства пришлось 1,06 млрд в общем объеме поставок, что соответствует 81,5% рынка. Спрос на iOS-устройства достиг 192,7 млн единиц, доля iPhone составила 14,8%. То есть в годовом исчислении суммарная доля двух названных операционных систем (ОС) также достигла 96,3%. Платформа Windows Phone по итогам года заняла 2,7% мирового рынка, BlackBerry – 0,4%.

На рисунке 1 приведен график популярности мобильных операционных систем за отчетный период 2010 – 2014 гг.

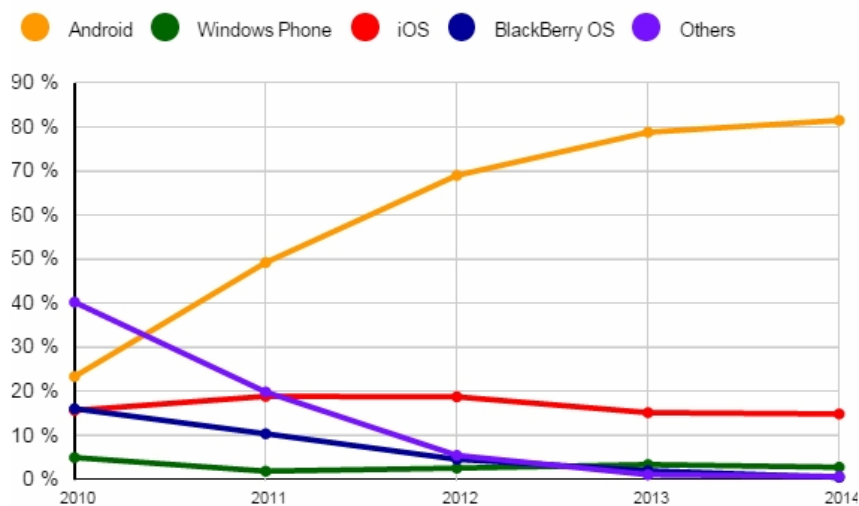


Рис. 1. График популярности мобильных ОС

Компания Google регулярно отслеживает информацию о том, насколько популярными являются различные версии операционной системы Android.

Круговая диаграмма на рисунке 2 демонстрирует, что наиболее популярной операционной системой этого семейства является Android 4.4 KitKat, которая установлена на 39,8% устройств. Месяцем ранее на долю этой версии ОС приходилось 41,4% устройств. Jelly Bean идет на втором месте с 39,2% (месяцем ранее у этой версии Android было 40,7%).

Данные собирались в течение недели, которая закончилась 4 мая 2015 г., учитывались все активные устройства, которые зашли за это время в Google Play [1].

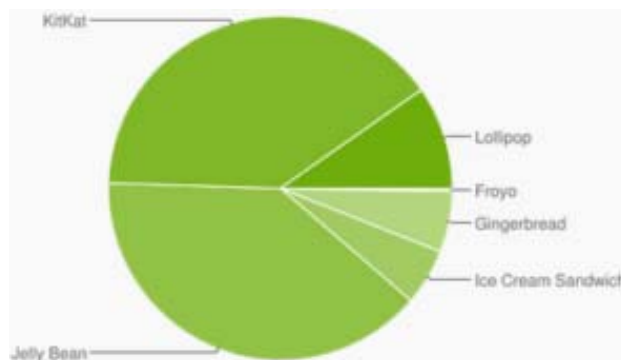


Рис. 2. Диаграмма популярности ОС семейства Android

Приложение, разработанное под Android, всегда найдет своего пользователя. И чем проще, актуальнее оно будет, чем меньше рекламы будет в приложении, тем выше вероятность того, что им будут пользоваться больший процент людей.

В сервисе Google Play заработать можно тремя способами:

- рекламные баннеры в бесплатном приложении;
- продажа приложений пользователям (основной вид заработка);
- реализация платных функций в бесплатных приложениях [2].

Наличие рекламы в приложении очень влияет на его популярность, количество скачиваний, и соответственно на окупаемость приложения, т.к. чем больше людей будет им пользоваться, тем больше прибыли оно будет приносить.

Как и любая реклама, реклама в Android-приложениях показывается ради прибыли. Разработчики бесплатных игр и приложений встраивают специальный код, который отвечает за показ рекламы. Иными словами: встраивая рекламу – разработчики получают прибыль.

Но иногда реклама бывает слишком агрессивной (показывается каждые несколько секунд или после каждого уровня игры). Порой кнопка закрытия рекламы намеренно скрыта, что вынуждает пользователя нажимать на рекламу, и, соответственно, пользователь после этого перенаправляется на сайт рекламодателя. Именно из-за таких способов и ухищрений реклама отталкивает пользовательскую аудиторию.

Для того чтобы приложение имело спрос и покупалось пользователями, оно должно быть уникальным в своем роде. Также, если пользователь готов платить за него деньги, то приложение должно быть реализовано на довольно высоком уровне, не содержать дефектов, полностью выполнять поставленные перед ним задачи. Как правило, стоимость приложений относительно невысокая, что привлекает пользователей. С каждой продажи приложения в сервис Google Play идет процент от стоимости приложения.

Платные функции в бесплатных приложениях – также довольно популярный способ заработка на приложении. Обычно в таких случаях приложение выполняет поставленную перед ним задачу бесплатно, но требует внесения денежных средств за дополнительную функциональность. Особенно часто такая ситуация наблюдается в мобильных играх. В качестве примера можно рассмотреть системы по учету финансов. Например, здесь бесплатными функциями могут быть добавление, редактирование, удаление, отображение записей о доходах и расходах. Но функции построения графиков, анализа данных могут стоить денег.

Заключение. Таким образом, разработка приложений для платформы Android целесообразна, т.к. платформа занимает ведущие позиции на рынке мобильных приложений. При создании приложения нужно учитывать также популярность версий данной операционной системы. Если есть желание получать доход от приложения, то есть смысл размещать в нем небольшие, ненавязчивые рекламные баннеры.

ЛИТЕРАТУРА

1. Специализированный российский информационно-аналитический сайт с самыми актуальными новостями из сферы IT [Электронный ресурс] / Статистика Google о популярности различных версий Android. – Режим доступа: <http://www.ixbt.com/news/it/188256>. – Дата доступа: 29.09.2015.
2. Способы заработка в Интернете [Электронный ресурс] / Как зарабатывать тысячи долларов на Android Market. – Режим доступа: <http://max1net.com/kak-zarabatyvat-tysyachi-dollarov-na-android-market/>. – Дата доступа: 29.09.2015.
3. Новости IT-индустрии [Электронный ресурс] / Суммарная доля Android и iOS на рынке смартфонов. Режим доступа: <http://www.3dnews.ru/910000/>. – Дата доступа: 29.09.2015.

УДК 004.514

КАК ДОЛЖНЫ ВЫГЛЯДЕТЬ ANDROID-ПРИЛОЖЕНИЯ

М.А. ИВАНОВ

(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

Рассмотрены нововведения дизайна Android-приложений, рекомендации по которому недавно выпущены компанией Google, и которых должны придерживаться разработчики приложений для данной платформы.

Введение. Платформа Android не имела жестких рекомендаций, как должны выглядеть и работать приложения. Существовал набор UI-рекомендаций, но они в основном концентрировались на мелочах вроде значков, виджетов и меню [3].

С самого запуска платформы были сотни различных идей интерфейсов и внешний вид приложений был очень разнообразным. Теперь, когда платформа достаточно усовершенствована и количество приложений резко возросло, происходит формирование пользовательского интерфейса Android. Некоторые функции интерфейса стали общими, а некоторые из них даже нашли свой путь в библиотеках Android SDK.

Основной раздел. Релиз Android Lollipop (5.0) принес с собой крупнейший набор пользовательских улучшений платформы, чем когда-либо. Эти изменения естественно влияют на то, как Android приложения будут выглядеть в будущем. Некоторые усовершенствования могут быть портированы обратно для более ранних версий, но не все из них.

Основное изменение, пришедшее с новой версией системы Material design, – новая составляющая пользовательского интерфейса, которая основана на простоте, яркости, понятности и функциональности.

Материальный дизайн – дизайн программного обеспечения и приложений операционной системы Android от компании Google впервые представлен на конференции Google I/O 25 июня 2014 г. Идея дизайна заключается в приложениях, которые открываются и сворачиваются как карточки, используя эффекты теней. По идее дизайнеров Google у приложений не должно быть острых углов, карточки должны переключаться между собой плавно и практически незаметно [1].

Материальный дизайн используется полноценно на операционных системах Android 5.0, Android 5.1, Android M и на некоторых приложениях Андроид 4.1, 4.2, 4.3 и 4.4.

В качестве примеров приложений в новом дизайне можно отметить мобильные приложения: Gmail, YouTube, Google+, Hangouts, Google Play, Telegram, Википедия, Ask.fm.

Material Design зиждется на четырех основных принципах (рис.):

1. Тактильные поверхности.

В Material Design интерфейс складывается из осязаемых слоев так называемой «цифровой бумаги». Эти слои расположены на разной высоте и отбрасывают тени друг на друга, что помогает пользователям лучше понимать анатомию интерфейса и принцип взаимодействия с ним.

2. Полиграфический дизайн.

Если считать слои кусками «цифровой бумаги», то в том, что касается «цифровых чернил» (всего того, что изображается на «цифровой бумаге»), используется подход из традиционного графического дизайна, например, журнального и плакатного.

3. Осмысленная анимация.

В реальном мире предметы не возникают из ниоткуда и не исчезают в никуда – такое бывает только в кино. Поэтому в Material Design всегда думают о том, как с помощью анимации в слоях и в «цифровых чернилах» давать пользователям подсказки о работе интерфейса.

4. Адаптивный дизайн.

Речь идет о том, как применяются предыдущие три концепции на разных устройствах с разными разрешениями и размерами экранов [2].

Касательно первого принципа есть ряд рекомендаций:

- глубина должна иметь смысл;
- необходимо заботиться о логистике;
- учитывать, что кнопка должна использоваться только для ключевого действия в приложении;

- если у какого-то объекта есть много форм и он содержит в себе много разного контента, то карточка подходит, а если нет, то лучше сделать это обычным текстом или текстовым списком;
- диалоговые окна нужны только, чтобы задать вопрос пользователю;
- имеет смысл использовать раскрытые списки.



Рис. Основные принципы материального дизайна

Выводы касательно полиграфического дизайна: цвет стал чем-то дополнительным, играет более заметную роль; контент на странице должен быть размещен свободно, базовая сетка в 8dp и отступ слева в 72dp – практически правило; фотографии и иллюстрации следует использовать в качестве выразительных средств.

В Material Design осмысленная анимация используется, чтобы показать, что именно только что произошло. Не стоит оставлять анимацию на самый конец – она может служить ключевым фактором пользовательского опыта и ее нужно продумывать заранее. Слишком много анимации – тоже плохо, анимация всегда должна быть осмысленной.

Google очень часто использует выдвижную навигацию в своих приложениях, ее можно видеть в разных примерах. Но у Google очень много задач, которые можно решить с ее помощью. Если разрабатываемое приложение имеет похожие задачи (разместить помощь, настройки, логин/логаут, информацию о пользователе и т.д.), то есть смысл использовать выдвижную навигацию; если создается простой инструмент, то ее использовать не стоит. Нужно помнить о том, что есть возможность менять размер панели инструментов динамически, делать ее двойного и тройного размера. Плавающая кнопка может быть где угодно: снизу, сверху, справа, слева. Кнопка может перемещаться с места на место в зависимости от задач. И следует учитывать тот момент, что приложение должно одинаково функционировать и отображаться на всей линейке Android-устройств, в горизонтальной и вертикальной ориентациях.

Заключение. С выходом материального дизайна произошла «тихая революция». Дизайн становится не просто и не только «рисованием», но и обретает собственную логику и законы поведения. Если раньше разработчики приложений и дизайнеры не придерживались рекомендаций по внешнему виду приложений, так как они, по сути, отсутствовали, то сейчас анонсированные компанией изменения в большой степени влияют на внешний вид, а также удобство (и, соответственно, популярность) использования приложений.

ЛИТЕРАТУРА

1. Википедия – свободная общедоступная мультиязычная универсальная Интернет-энциклопедия [Электронный ресурс] / Material Design. – Режим доступа: https://ru.wikipedia.org/wiki/Material_design/. – Дата доступа: 29.09.2015.
2. Хабрахабр [Электронный ресурс] / Material Design. – Режим доступа: <http://habrahabr.ru/company/redmadrobot/blog/252773/>. – Дата доступа: 29.09.2015.
3. Хабрахабр [Электронный ресурс] / Как должны выглядеть Android приложения? – Режим доступа: <http://habrahabr.ru/post/149610/http://habrahabr.ru/company/redmadrobot/blog/252773/>. – Дата доступа: 29.09.2015.

УДК 440.8.028

**СИСТЕМЫ МОНИТОРИНГА ПОКАЗАНИЙ ПРОБЕГА
АВТОТРАНСПОРТА****П.А. КАЛИНИН****(Представлено: В.М. ЧЕРТКОВ)**

Представлены системы мониторинга пробега на основании штатного одометра и системы спутниковой навигации GPS/ГЛОНАСС. Рассмотрены конструктивные исполнения одометров и их реальные погрешности в показаниях. Определены преимущества использования системы GPS/ГЛОНАСС.

Любая система мониторинга показаний пробега транспорта показывает меньше, чем показания штатного одометра. Различные значения пробега, полученные с одометров, установленных на автомобилях, и с помощью спутниковой системы мониторинга, могут стать причиной служебных проверок и конфликтных ситуаций в организации.

Конструкция одометров и их погрешности.

Для измерения пройденного пути на транспортном средстве используют специальный прибор – одометр. Бортовые одометры всех видов не относятся к классу точных приборов. Для каждого вида данных устройств установлены допустимые погрешности. Для оценки погрешности в показаниях приборов следует учитывать следующие моменты, заложенные на этапе их производства [1]:

1. Погрешность прибора установлена только для самих приборов. Все конструктивные изменения, а так же физический износ некоторых узлов и агрегатов автомобиля в его погрешность не включены.

2. По техническим требованиям ЕЭК ООН № 39 спидометры не могут занижать показания. Средняя погрешность спидометра по этим правилам (ГОСТ Р 41.39-99) может быть только положительной и не превышать истинную скорость движения более чем на 10% + 6 км/ч. Поэтому одометр, конструктивно связанный со спидометром, также дает завышенные показания.

Заводы-изготовители завышают показания скорости и пробега на 5–10%. Об этом ведется множество разговоров и бурные обсуждения на форумах автолюбителей. Возможно, что автопроизводители заботятся не только о безопасности водителей, но и вполне законно (опираясь на правила ЕЭК ООН № 39) уменьшают реальный гарантийный пробег на неизвестную величину, потому что отсутствуют требования к точности измерения пробега [2].

Одометры бывают различных конструкций: механические, электромеханические или электронные.

Механический одометр – весьма оценочное устройство, имеет собственную погрешность до 5%. В зависимости от условий эксплуатации, износа деталей и агрегатов и использовании неоригинальных запчастей суммарная погрешность прибора может достигать 15% [3].

Электромеханические одометры – основаны на показаниях электронного измерителя числа импульсов от датчика скорости, расположенного обычно на коробке передач. Эти приборы несколько точнее механических, ведь они избавились лишь от нескольких слабых мест механической части. Погрешность большинства из них находится в пределах 5–7% [4].

Электронные одометры – дальнейшее эволюционное развитие. Избавились от механического индикатора, который заменили жидкокристаллическим экраном. В то же время сам принцип *контроля пройденного пути* (обороты колеса) остается неизменным, и даже точная электроника находится в зависимости от технического состояния ходовой части автомобиля. Если провести дополнительную калибровку на тестовом участке пути (на заводе-изготовителе эта процедура не происходит), погрешность данных устройств редко превышает 5% [5].

Существенный минус электронных одометров – возможность изменения его микропрограммы. Для популярных автомобилей существует услуга по «модификации» бортового одометра, для завышения его показаний на любую величину (например: +20% к пробегу). Для выявления подобного вмешательства нужен контрольный заезд или обращение в службу ремонта [3].

Общие факторы, влияющие на любые одометры.

1. Радиус колеса может внести существенную погрешность в показания одометра. Разница в высоте протектора в 1 см, например, даст на 100 км пробега автомобиля разницу в пробеге в 1955 м: диаметр одного колеса 1 м, второго – 1,02 м. Первое совершит 31 830 оборотов, второе – 31 206. Каждый оборот – 3,1416 м, разница – 1955 м. И эту разницу мы получаем только при одном сантиметре. Например, разные шины 325/70 и 325/75 дадут сразу разницу в диаметре в 3,2 см. Поэтому одометр на автомобиле со стертým протектором покажет большее значение по сравнению с таким же автомобилем, но на

новых шинах. Еще важно знать, на какой радиус колес рассчитан одометр: если поставить другой размер колес, то будут совсем другие данные по скорости и пройденному пути.

2. Вес груза – при полной или чрезмерной загрузке автомобиля шина проминается по-разному, поэтому изменяется диаметр колеса.

3. Давление в шинах – шина проминается по-разному при штатном и нештатном давлении. На давление влияет температура, при прогретых или перегретых шинах оно выше.

4. Скольжение колес – при пробуксовках, скольжениях или же, наоборот, торможении на льду, автомобиль либо находится на месте при вращении колес, либо, наоборот, движется при блокировке колес.

Измерение пробега системой GPS/ГЛОНАСС мониторинга.

Система мониторинга транспорта на основе спутниковой навигации может определять пройденное расстояние тремя основными способами [3]:

1. Подключение к штатному датчику: данные о пробеге рассчитываются на основе данных получаемых с датчика скорости, установленного в автомобиле. Данный способ позволяет добиться полного соответствия данных измерений с одометром или тахографом. Иногда такой способ более предпочтителен даже более высокоточного навигационного способа, когда необходимо бухгалтерское соответствие путевых листов, являющихся первичным документом, и программы мониторинга.

2. По координатам точек маршрута: данные о пробеге рассчитываются как расстояние по прямой между координатами точек (долгота и широта), которые присылает прибор. Причем временной интервал между присланными точками может быть разный. Обычный интервал: 10–30 с. Также для повышения точности многие регистраторы присылают точки в случае изменения угла движения. Использование данного способа в современных системах мониторинга не рекомендуется из-за ограниченной точности.

3. Рассчитывается «Вояджером»: данные о пробеге рассчитываются на основе дополнительной информации получаемой с GPS-приемника. В данном способе терминал сам определяет моментальное значение скорости каждую секунду. Данный способ является максимально точным по отношению к двум предыдущим.

Общим недостатком использования любой навигационной системы является то, что при определенных условиях сигнал может не доходить до приемника, или приходиться со значительными искажениями или задержками. Например, практически невозможно определить свое точное местонахождение в подвале или в тоннеле. Хотя есть разработки, использующие акселерометры, которые позволяют весьма точно определить траекторию, скорость и пробег без доступных спутников, остается ждать их внедрения в автомобильные трекеры [4].

По официальным данным чистая погрешность модуля ГЛОНАСС/GPS находится в пределах 2–5 м (это порядка 1,5% в определении пробегов). Модуль ГЛОНАСС/GPS ГЕОС-1 обладает обязательным сертификатом как средство измерения. В настоящее время точность определения координат системой ГЛОНАСС несколько отстает от аналогичных показателей для GPS. Согласно данным СДКМ на 22 июля 2011 года ошибки навигационных определений ГЛОНАСС по долготы и широте составляли 4,46–7,38 м при использовании в среднем 7–8 видимых спутников (в зависимости от точки приема). В то же время ошибки GPS составляли 2,00–8,76 м при использовании в среднем 6–11 видимых спутников (в зависимости от точки приема). Рельеф местности также не оказывает сильного влияния на погрешность измерения пробега. Например, при уклоне по знаку в 15–20% угол наклона дороги составит 8,53–11,31°, а погрешность измерения пробега составит около 1,5%. И даже при очень крутом уклоне в 40% по знаку (угол наклона дороги составит 21,8°), погрешности измерения пробега составит всего около 6%. Но все современные трекеры умеют определять высоту и делать соответствующие поправки при вычислении пробега. Таким образом, общая погрешность систем мониторинга транспорта ГЛОНАСС/GPS при нормальных условиях составляет менее 3,5%.

Заключение. На основании сравнения систем мониторинга штатного одометра и системы спутниковой навигации GPS/ГЛОНАСС можно сделать вывод о том, что наиболее эффективной в использовании является система спутниковой навигации GPS/ГЛОНАСС. Данная спутниковая система может быть установлена на автомобиле или другие средства передвижения вследствие малой погрешности в показателях пройденного расстояния (не более 5%), в то время как у одометров погрешность может достигать порядка 15%.

ЛИТЕРАТУРА

1. Global Positioning System, Standard Positioning System Service, Signal Specification [Электронный ресурс]. – 2nd ed. – 1995. – P. 18. – Режим доступа: <http://www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf>.
2. NAVCEN: GPS SPS Signal Specifications [Электронный ресурс]. – 2nd ed. – 1995. – Режим доступа: [Uhttp://www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf](http://www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf).

3. ГЛОНАСС: принципы построения и функционирования / под ред. А.И. Перова, В.Н. Харисова. – 3-е изд., перераб. – М. : Радиотехника, 2005. – 688 с.
4. Сетевые спутниковые радионавигационные системы / В.С. Шебшаевич [и др.] ; под ред. В.С. Шебшаевича. – 2-е изд., перераб. и доп. – М. : Радио и связь, 1993. – 408 с.
5. Соловьев, Ю.А. Системы спутниковой навигации / Ю.А. Соловьев. – М. : ЭКО-ТРЕНДЗ, 2000. – 369 с.

УДК 440.8.028

УСТРОЙСТВО ОЦЕНКИ ПОГРЕШНОСТИ СПИДОМЕТРА И ОДОМЕТРА АВТОМОБИЛЯ

П.А. КАЛИНИН

(Представлено: В.М. ЧЕРТКОВ)

Представлена разработка структурной схемы устройства определения скорости и пробега. Рассмотрены основные функциональные блоки, на основании которых спроектирован и собран макет устройства. Приведены экспериментальные характеристики отклонения пробега от реального с использованием спроектированного GPS устройства и одометра.

Структурная схема данного устройства определения скорости и пробега разрабатывалась на основании анализа функций и задач, а именно минимизации погрешности показаний пройденного расстояния.

На рисунке 1 представлена структурная схема разрабатываемого устройства.



Рис. 1. Структурная схема устройства

В данной схеме можно выделить следующие функционально законченные блоки [1]:

- схема питания;
- модуль GPS CRIUS NEO-6;
- датчик импульсов привода колеса;
- блок микроконтроллеров;
- блок индикации.

Разработанная структурная схема устройства определения скорости и пробега на рисунке 1. Блоки, представленные на рисунке, выполняют следующие задачи:

1) схема питания.

Данный блок должен формировать нужный уровень выходного напряжения, достаточного для работы всех блоков проектируемого устройства;

2) блок модуля GPS CRIUS NEO-6.

Данный блок должен производить прием информации со спутника, а так же формировать сигналы, передаваемые в блок микроконтроллера;

3) блок микроконтроллеров.

Данный структурный блок производит управление всеми подконтрольными ему устройствами, обрабатывает данные, поступающие ему на вход от модуля GPS [2];

4) блок индикации;

5) блок формирователя импульсов.

Данный блок служит для формирования импульсов от вращения привода колеса. Данные поступают на блок микроконтроллера ATMEGA8A. Далее информация обрабатывается и передается по интерфейсу SPI в первый микроконтроллерный блок. Уже потом при вычислении скорости результаты сравниваются и выводятся на LCD.

На основании данной структурной схемы (см. рис. 1) была разработана принципиальная электрическая схема, а также спроектирован и собран макет устройства определения скорости и пробега, представленные на рисунке 2.

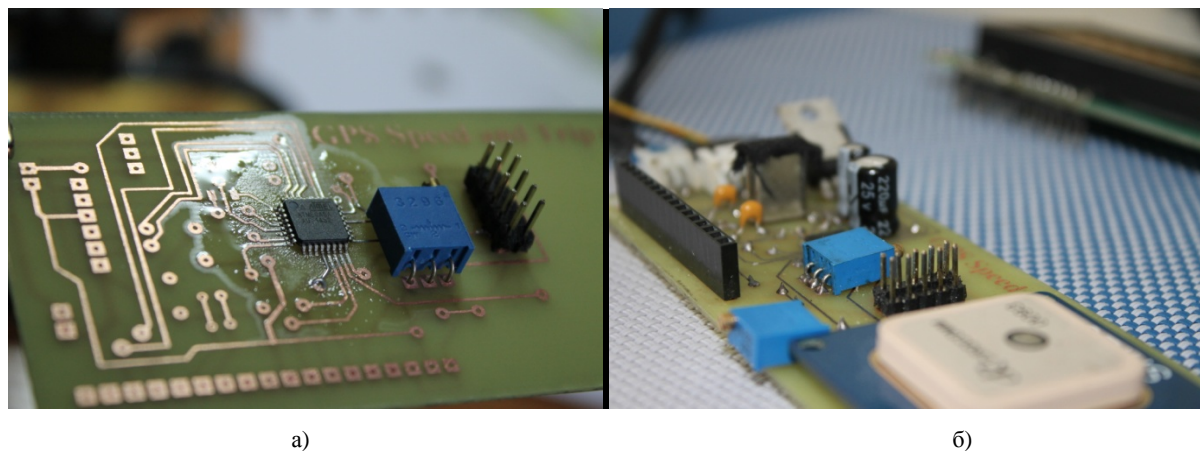


Рис. 2: а – печатная плата, б – макет собранного устройства

Снятие экспериментальных данных производили на отрезке трассы продолжительностью 75 км. Заезд выполнялся в два этапа. Прохождение данных участков производилось с одинаковой скоростью 100 км/ч. Расстояние разгона и время разгона не учитывалось на данных участках. Схема проведения эксперимента представлена на рисунке 3.



Рис. 3. Схема проведения эксперимента

Так как в экспериментальном автомобиле одометр электронный, то подключать датчик импульсов для подсчета значений пробега с точностью до метров не было смысла. Общее расстояние составило 75 км, после того как мы проехали 35 км, то дальнейшее отслеживание производили на том же автомобиле, но с другим размером шин. Потом выполнили замеры по нашему устройству с использованием модуля GPS. Ниже на рисунке 4 приведены графики зависимости отклонения реального пробега от измеренного.

Далее построена такая же характеристика, но с использованием модуля GPS. Участок точно также был измерен на двух автомобилях с разным размером шин. На рисунке 5 представлена характеристика отклонения пробега от реального с использованием нашего устройства.

В ходе анализа графиков можно сделать вывод, что пробег по одометру отклоняется линейно. Также отметим то, что при смене размера шин коэффициент отклонения вырос на определенную величину. В нашем эксперименте учувствовал автомобиль марки Renault Laguna 2. По техническим документам на данное техническое средство в данном автомобиле, как и во многих авто, отсутствует корректровка коэффициента поправки при смене размера шин.

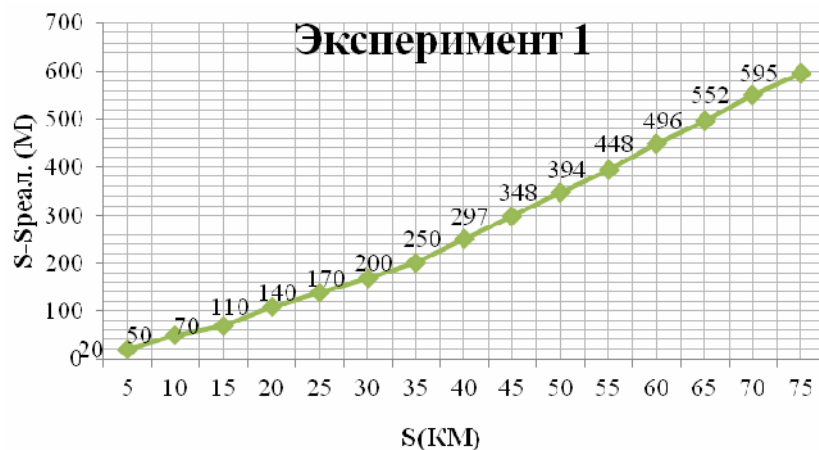


Рис. 4. Зависимость отклонения пробега от пройденного расстояния по одометру (датчику импульсов)

Из недостатков данной системы можно особо выделить зависимость измеряемых величин от технического состояния автомобиля, невозможность корректировки коэффициента отклонения пробега от реального [3].

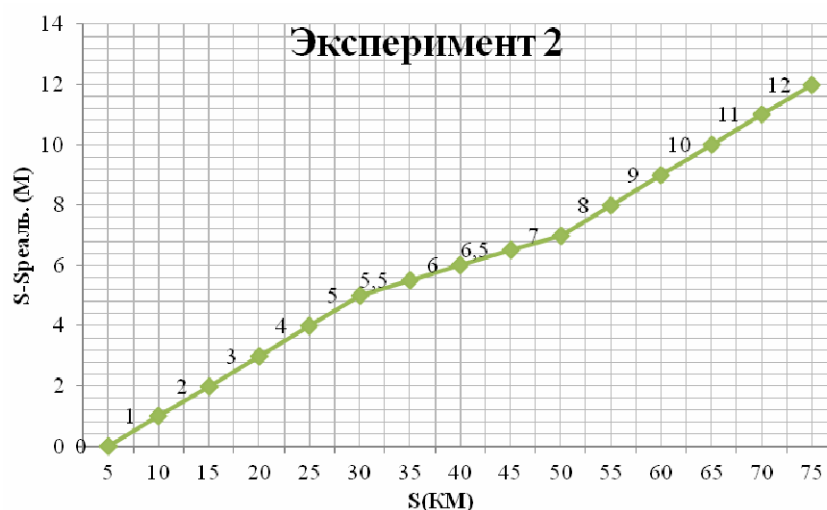


Рис. 5. Характеристика отклонения пробега от реального с использованием нашего GPS устройства

Данные характеристики были сняты с условиями приема от 4–6 спутников. На промежутке 30–45 км слежение осуществлялось по 11–12 спутникам. Среднее время для нахождения хотя бы трех спутников занимало около 2 мин. В данном случае следует отметить такие недостатки, как зависимость погрешности от количества найденных спутников; продолжительность некоторого времени для поиска спутников.

Заключение. Разработана структурная схема устройства, указаны и рассмотрены его основные блоки. На основании структурной схемы был собран макет, с помощью которого были проведены эксперименты по определению отклонений пробега и скорости от реальных данных.

В ходе проведения анализа экспериментально полученных данных можно сделать вывод, что отклонение от реального километража намного меньше при использовании спроектированного GPS устройства, чем у одометра.

ЛИТЕРАТУРА

1. Белов, А.В. Создаем устройство на микроконтроллерах / А.В. Белов. – М. : Наука и техника, 2007. – 513 с.

2. Сетевые спутниковые радионавигационные системы / В.С. Шебшаевич [и др.] ; под ред. В.С. Шебшаевича. – 2-е изд., перераб. и доп. – М. : Радио и связь, 1993. – 408 с.
3. Соловьев, Ю.А. Системы спутниковой навигации / Ю.А. Соловьев. – М. : ЭКО-ТРЕНДЗ, 2000. – 369 с.

УДК 004.057

МОДЕЛИРОВАНИЕ АНТЕНН В ПРОГРАММНОМ ПАКЕТЕ ANTENNA MAGUS

А.В. КАРАСЬ

(Представлено: канд. техн. наук, доц. В.Ф. ЯНУШКЕВИЧ)

Рассмотрено моделирование антенн с помощью программного пакета Antenna Magus. Сделан обзор основных функций и возможностей. Определены области ее использования.

Продукт Antenna Magus (рис. 1), разработанный компанией Magus (PTY) Ltd и предназначенный для проектирования и моделирования антенн различных типов, представляет собой базу данных различных антенн, из которой пользователь может выбрать подходящую параметризованную модель и экспортировать ее в пакет CST MICROWAVE STUDIO, где затем выполняется ее моделирование и оптимизация.

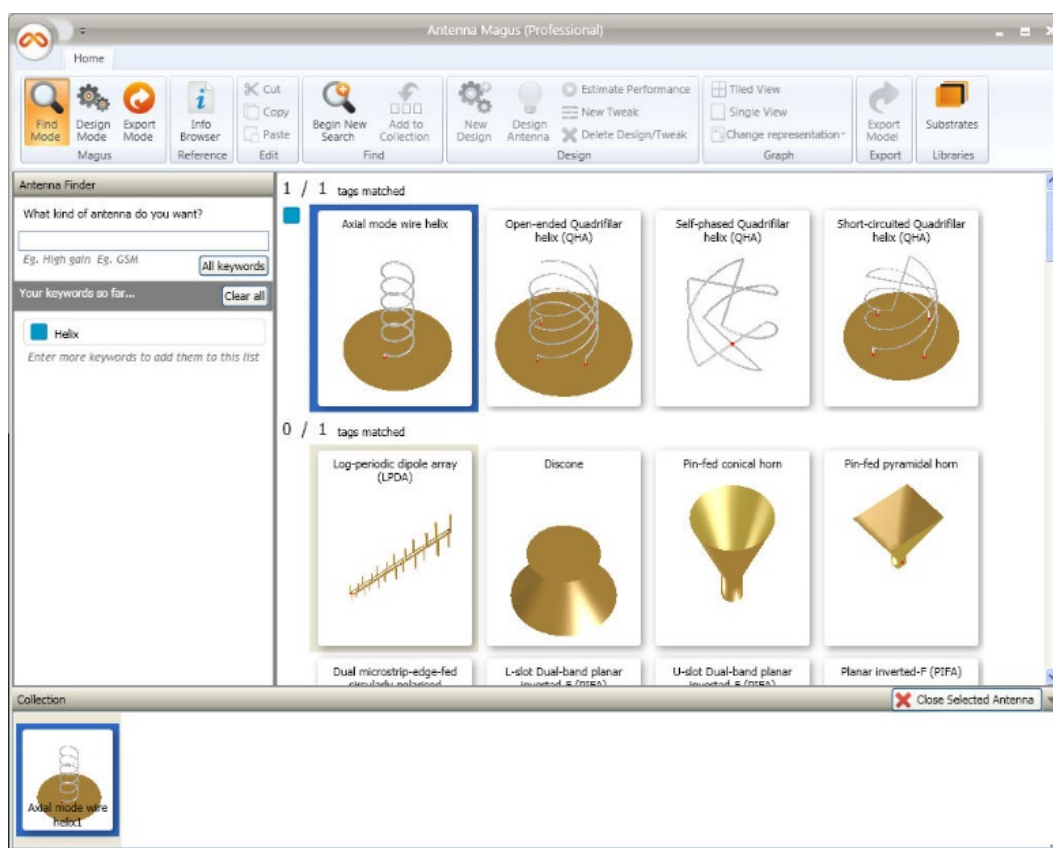


Рис. 1. Заглавная страница программного пакета Antenna Magus

Пакет ориентирован на широкие массы разработчиков антенн, специалистов по проблемам EMC, а также системных интеграторов, выполняющих оценку расположения антенн на крупных объектах. Программа Antenna Magus анализирует доступные цели проектирования и выполняет интеллектуальный выбор недостающих начальных параметров.

Впоследствии пользователь может вручную подстроить предложенные программой настройки (рис. 2). Для реализации технологии Smart Design все цели проектирования в Antenna Magus организованы в группы. Для каждой из этих групп пользователю доступен свой калькулятор параметров, который позволяет различными способами реализовать поставленную задачу, группируя начальные цели в про-

ектные требования. Технология Smart Design дает пользователю возможность задать лишь те параметры, которые известны ему, и полностью довериться Antenna Magus для получения конечных результатов за один этап проектирования (рис. 3).

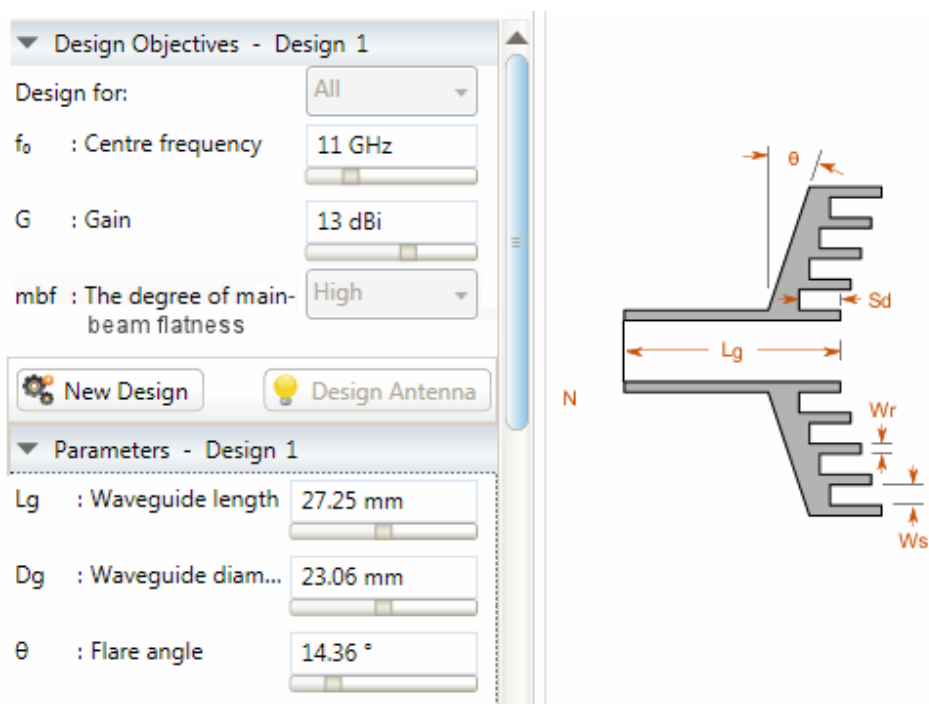


Рис. 2. Изменение параметров моделируемой антенны в программном пакете Antenna Magus

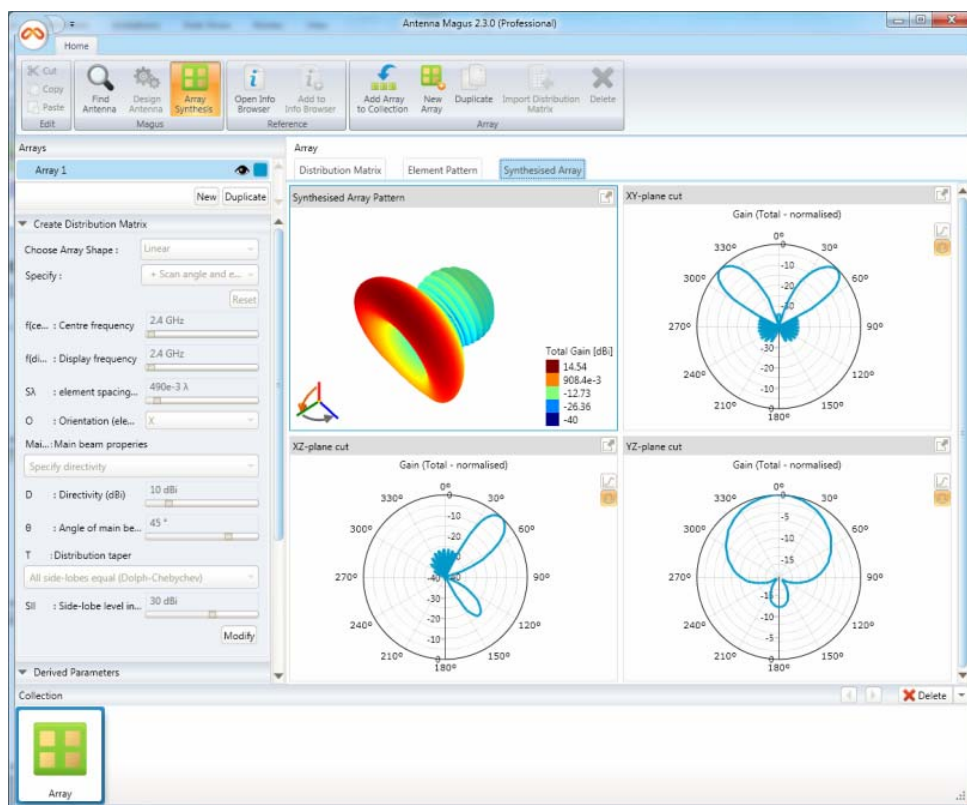


Рис. 3. Параметры моделируемой антенны

Несмотря на то, что для диэлектрических стержневых антенн доступно множество вариантов реализации, в большинстве случаев они представляют собой сужающийся стержень круглого или прямоугольного сечения. При этом фидерный волновод возбуждает часть входной мощности в виде поверхностной волны, распространяющейся вдоль стержня с минимальным отражением. Полученная волна начинает непрерывно излучать, постепенно преобразовываясь из ограниченной в волну в свободном пространстве. Также в Antenna Magus добавлен ряд практических моделей широкополосных планарных спиральных антенн с дополнительным поглощающим слоем. Слои поглощающего материала используются для получения однонаправленного излучения при сохранении много октавного рабочего диапазона спиральных излучателей.

ЛИТЕРАТУРА

1. Компания ООО «Евроинтех» [Электронный ресурс] / Евроинтех. – Режим доступа: <http://www.eurointech.ru>. – Дата доступа: 24.09.2015.

УДК 006.3/8

РАДИОЭЛЕКТРОННЫЕ УСТРОЙСТВА, ИСПОЛЬЗУЕМЫЕ В ПОЛОСЕ ЧАСТОТ 2300–2400 МГц ДЛЯ СИСТЕМ ТЕЛЕКОММУНИКАЦИЙ

А.В. КАРАСЬ

(Представлено: канд. техн. наук, доц. В.Ф. ЯНУШКЕВИЧ)

Представлены радиоэлектронные устройства, для полосы частот 2300–2400 МГц в системах телекоммуникаций. Рассмотрены характеристики и параметры данных устройств. Описаны основные стандарты, используемые в диапазоне 2300–2400 МГц.

В соответствии с решением Государственной комиссии по радиочастотам (ГКРЧ) от 28.11.2005 № 05-10-01-001 Федеральным государственным унитарным предприятием «Научно-исследовательский институт радио» (ФГУП НИИР) совместно с научно-исследовательскими учреждениями Министерства обороны Российской Федерации проведена разработка условий совместного использования полосы радиочастот 2300–2400 МГц радиоэлектронными средствами (РЭС) беспроводного доступа с РЭС, используемых для нужд государственного управления. В ходе работ был проведен анализ международного опыта регулирования использования радиочастотного спектра в полосе радиочастот 2300–2400 МГц, технических характеристик РЭС, применяемых и планируемых к применению на территории Российской Федерации, загрузки радиочастотного спектра, а также разработаны варианты и условия использования полосы радиочастот РЭС беспроводного доступа (табл.).

Таблица

Основные параметры стандарта LTE

Параметр	Размерность	Значение ТТХ	
		Абонентская станция	Базовая станция
Метод радиодоступа		TDMA, OFDMA, TDD доступ на основе временного и ортогонального частотного мультиплексирования	
Рабочий диапазон частот	МГц	2300-2400	
Классы излучения		5M00G7W, 8M75G7W, 10M0G7W, 2,0M0G7W, 5M00D7W, 8M75D7W, 10M0D7W, 20M0D7W	
Ширина полосы излучения, МГц, на уровне	-3 дБ	5/8,75/10/20	
	-30 дБ	5,5/9/11/22	
Максимальная мощность передатчика	дБм	24	46
Побочные излучения		В соответствии с действующими нормами ГКРЧ	
Избирательность по соседнему каналу		В соответствии с Рекомендацией МСЭ-Р М. 1457	
Максимальный коэффициент усиления антенны	дБ	7	20

Принимая во внимание результаты исследований и учитывая необходимость обеспечения электромагнитной совместимости действующих на территории Российской Федерации РЭС различного назначения и РЭС перспективных сетей радиосвязи, ГКРЧ решила:

1. Признать возможным использование полосы радиочастот 2300–2400 МГц РЭС беспроводного доступа.

2. Утвердить прилагаемые основные тактико-технические характеристики РЭС беспроводного доступа, использующих полосу радиочастот, указанную в пункте 1 настоящего решения ГКРЧ.

3. Выделение полос радиочастот и присвоение (назначение) радиочастот или радиочастотных каналов для РЭС беспроводного доступа в полосе радиочастот 2300–2400 МГц должно осуществляться по заявлениям юридических и физических лиц в установленном порядке при условии соответствия параметров заявляемых РЭС беспроводного доступа тактико-техническим характеристикам, приведенным в Приложении к настоящему решению ГКРЧ.

Стандарт LTE – Long-Term Evolution – или стандарт связи четвертого поколения считается перспективным направлением развития сетей (рис.). В радиотехнической части применены новые методы модуляции: OFDM на downlink и SC-FDMA на uplink. Технология MIMO, которая применялась и в HSPA+ (2×2 антенн), получила дальнейшее развитие (4×4 антенн и более, в зависимости от релиза 3GPP и поддержки терминалами).

Преимущество FDD – более низкая интерференция между соседними BS и более высокая скорость в Downlink на одинаковой с TDD ширине канала (скорость Downlink в TDD на канале 20 МГц соответствует скорости в FDD на канале 15 МГц). TDD с другой стороны, более эффективно использует ресурсы при асимметричном канале (каким и является канал в мобильной связи), поскольку в TDD возможно регулировать соотношение ресурсов для Downlink/Uplink. Поэтому, как FDD, так и TDD нашли свое место в современных сетях LTE. Кардинальные изменения коснулись и ядра сети. В LTE забыли про старый, добрый OKC7 и перешли на новый протокол – DIAMETER (на базе IP). Все сигнальные интерфейсы между элементами сети работают по протоколу DIAMETER, за исключением LEGACY 3G элементов, соединения с которыми происходят по SIGTRAN (опять же IP).

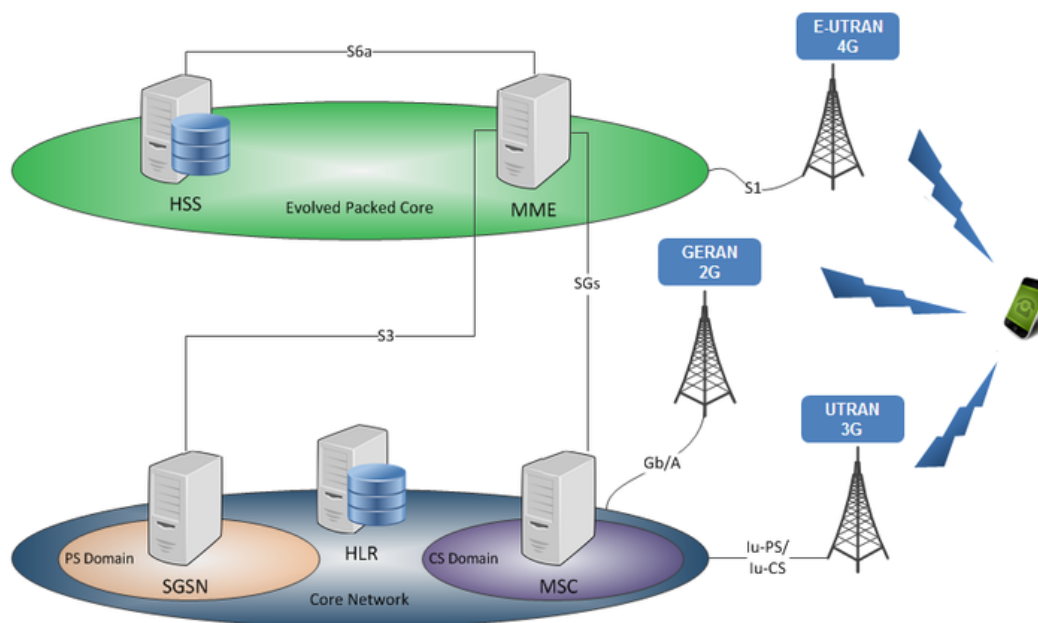


Рис. Структурная схема стандарта LTE

Основное отличие LTE от 3G – более высокая скорость передачи данных. В 3G предел – 42 Мбит/с, это пиковая скорость загрузки данных в самой современной модификации стандарта 3G – HSPA+. Реальная же пока составляет 2-3 Мбит/с. Тогда как пользователи московской LTE-сети «МегаФона» уже сейчас скачивают данные на скорости более 20 Мбит/с.

ЛИТЕРАТУРА

1. Что такое LTE и 4G от МегаФона [Электронный ресурс] / Хабрахабр. – М. – Режим доступа: <http://habrahabr.ru/company/megafon/blog/154153>. – Дата доступа: 24.09.2015.

УДК 004.023

ОБЗОР РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ БИОМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ

М.А. КВЕТИНСКИЙ
(Представлено: М.В. МАТЮШ)

Рассматривается система биометрической идентификации, ее режимы работы, функции, принцип ее работы, критерии для анализа поведения пользователя за компьютером, обзор и выбор подходящих сред разработки и инструментов проектирования

Разрабатываемая система предназначена для считывания, сохранения и анализа поведенческой биометрии пользователя персонального компьютера, а также основываясь на полученных при этом данных, вынесения решения, относительно того, какой это пользователь.

Она должна делать это с должной степенью точности с минимальным количеством ложных срабатываний.

Система должна иметь два режима работы:

- режим обучения, в котором система анализирует, собирает и сохраняет ключевые характеристики и параметры поведенческой биометрии;
- режим контроля, в котором система анализирует поведение пользователя и на основе наблюдений выносит решение, соответствует ли оно заложенному ранее образцу.

Для анализа поведения пользователя за компьютером используются особенности его поведенческой биометрии, называемые критериями. Устройствами взаимодействия с пользователем являются клавиатура и мышь. В то время как движения мыши крайне сложно поддаются анализу, действия пользователя за клавиатурой идеально подходят для анализа.

Состав выполняемых функций:

- отслеживание параметров поведенческой биометрии пользователя в реальном времени;
- сохранение параметров поведенческой биометрии пользователя в его профиль;
- анализ параметров поведенческой биометрии;
- вынесение решения о валидности пользователя на основании считанных в реальном времени параметров поведенческой биометрии пользователя.

Всего можно выделить три основных критерия поведения пользователя, получаемых от клавиатуры:

- время нажатия – время между моментом нажатия и моментом отпускания клавиши;
- время повторного нажатия – время между нажатиями двух клавиш;
- время набора последовательности – время набора определенной последовательности символов.

Они представлены на рисунке.



Рис. Критерии поведения пользователя, получаемые от клавиатуры

Каждый из этих критериев имеет свой вес, в общей сумме складывающихся в коэффициент доверия пользователю. Благодаря тонкой настройке «веса» каждого критерия, можно добиться максимальной точности определения валидности пользователя.

Выбор средств разработки данного проекта осуществлялся из типа разрабатываемого приложения. Так как требовалось разработать систему для персонального компьютера и, учитывая, что подавляющее большинство пользователей используют операционную систему Windows, было вынесено решение работать, ориентируясь именно на эту ОС.

В качестве среды разработки рассматривались QT CREATOR и Microsoft Visual Studio.

Основная задача QT CREATOR – упростить разработку приложения с помощью фреймворка QT на разных платформах. Поэтому среди возможностей, присущих любой среде разработки, есть и специфичные, такие как отладка приложений на QML и отображение в отладчике данных из контейнеров QT, встроенный дизайнер интерфейсов как на QML, так и на QtWidgets.

В QT CREATOR реализовано автодополнение, в том числе ключевых слов, введенных в стандарте C++11, подсветка кода (ее определение аналогично таковому в Kate, что позволяет создавать свои виды подсветок или использовать уже готовые).

Также начиная с версии 2.4 есть возможность задания стиля выравнивания, отступов и постановки скобок.

Реализован ряд возможностей при работе с сигнатурами методов, а именно:

- автогенерация пустого тела метода после его обновления;
- возможность автоматически изменить сигнатуру метода в определении, если она была изменена в объявлении и наоборот;
- возможность автоматически поменять порядок следования аргументов.

Другая же среда разработки, Microsoft Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня.

Кроме того, Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Используемый в разработке программного обеспечения инструментарий не ограничивается только лишь системой управления проектом и интегрированной средой разработки. Для моделирования и проектирования разработаны специализированные инструменты, о которых обязательно нужно знать разработчикам и уметь применять.

Наилучшим инструментом проектирования является Sparx Enterprise Architect.

Enterprise Architect – достаточно мощное программное обеспечение, предназначенное для работы с UML.

Это приложение разработано с целью модернизация системы, создания и управления бизнес проектами, построение действующих моделей бизнес-планов и так далее. В настоящее время программа поддерживает огромное количество языков программирования, таких как ActionScript, C, C++, C# and VB .NET, Java, Visual Basic 6, Python, PHP, XSD и WSDL.

Использовать ее функционал можно в самых разных сферах – бухгалтерской системе, веб-разработке, медицине, исследовательских и научных работах, электротехнике. Причем программу можно использовать не только как средство анализа и моделирования процессов с помощью UML-диаграмм, но и для обучения. Это подтверждается популярностью приложения во многих иностранных колледжах.

Помимо прочего, стоит отметить, что в Enterprise Architect хорошо продумано взаимодействие диаграмм с программным кодом, имеется встроенная автоматическая генерация кода, с гибкой настройкой параметров, а также имеется набор инструментов для более детального редактирования кода прямо в окне приложения [1].

Данный программный продукт обеспечивает:

- создание профиля параметров поведенческой биометрии пользователя;
- идентификацию пользователя, основываясь на сравнении полученных от него параметров поведенческой биометрии и эталонных параметров, содержащихся в его профиле.

Учитывая тот факт, что аналогов данного программного продукта в свободном доступе не существует, нельзя провести его сравнение, однако на момент окончания разработки, он обладает такими качествами как:

- простота в использовании;
- легкость в освоении;
- наглядность.

Разрабатываемое приложение является системой отслеживания и анализа поведенческой биометрии.

На всемирном рынке в свободном доступе существуют десятки систем, работающих с биометрическими параметрами, но лишь единицы из них рассматривают их с поведенческой точки зрения. Чаще всего рассматриваются отпечатки пальцев, сетчатка глаза, анализ голоса и лица пользователя, что неприменимо к нашему случаю.

ЛИТЕРАТУРА

1. Интернет-портал Devprom [Электронный ресурс] / Инструменты проектирования: Enterprise Architect. – Режим доступа: <http://devprom.ru/news-Enterprise-Architect>. Дата доступа: 15.09.15.

УДК 004.023

АНАЛИЗ СУЩЕСТВУЮЩИХ СРЕДСТВ БИОМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ

М.А. КВЕТИНСКИЙ
(Представлено: М.В. МАТЮШ)

Рассмотрены общие понятия, методы и технологии биометрической идентификации, как это работает и для чего нужно. Приведен обзор нескольких разрабатываемых и уже готовых проектов.

Биометрия – это методы автоматической идентификации человека и подтверждения личности человека, основанные на физиологических или поведенческих характеристиках. Примерами физиологических характеристик являются отпечатки пальцев, форма руки, характеристика лица, радужная оболочка глаза. К поведенческим характеристикам относятся особенности или характерные черты, либо приобретенные или появившиеся со временем, то есть динамика подписи, идентификация голоса, динамика нажатия на клавиши.

Все биометрические системы работают практически по одинаковой схеме. Во-первых, система запоминает образец биометрической характеристики (это и называется процессом записи). Во время записи некоторые биометрические системы могут попросить сделать несколько образцов для того, чтобы составить наиболее точное изображение биометрической характеристики. Затем полученная информация обрабатывается и преобразовывается в математический код.

Идентификация по любой биометрической системе проходит четыре стадии:

- запись – физический или поведенческий образец запоминается системой;
- выделение - уникальная информация выносится из образца и составляется биометрический образец;
- сравнение – сохраненный образец сравнивается с представленным;
- совпадение/несовпадение – система решает, совпадают ли биометрические образцы, и выносит решение.

Подавляющее большинство людей считают, что в памяти компьютера хранится образец отпечатка пальца, голоса человека или картинка радужной оболочки его глаза. Но на самом деле в большинстве современных систем это не так. В специальной базе данных хранится цифровой код длиной до 1000 бит, который ассоциируется с конкретным человеком, имеющим право доступа. Сканер или любое другое устройство, используемое в системе, считывает определенный биологический параметр человека. Далее он обрабатывает полученное изображение или звук, преобразовывая их в цифровой код. Именно этот ключ и сравнивается с содержимым специальной базы данных для идентификации личности.

Биометрические данные можно разделить на два основных класса:

- физиологические – относятся к форме тела. В качестве примера можно привести: отпечатки пальцев, распознавание лица, ДНК, ладонь руки, сетчатка глаза, запах, голос;
- поведенческие – связаны с поведением человека. Например, походка и речь. Иногда для этого класса биометрии используется термин англ. *behaviometrics*.

Идентификация по отпечаткам пальцев – самая распространенная, надежная и эффективная биометрическая технология. Благодаря универсальности этой технологии она может применяться практически в любой сфере и для решения любой задачи, где необходима достоверная идентификация пользова-

телей. В основе метода лежит уникальность рисунка папиллярных узоров на пальцах. Отпечаток, полученный с помощью специального сканера, датчика или сенсора, преобразуется в цифровой код и сравнивается с ранее введенным эталоном. Надежность данного способа идентификации личности, состоит в невозможности создания идентичного отпечатка. Наиболее совершенную технологию идентификации по отпечаткам пальцев реализуют оптические сканеры.

Технология распознавания радужной оболочки глаза была разработана для того, чтобы свести на нет навязчивость сканирования сетчатки глаза, при котором используются инфракрасные лучи или яркий свет. Ученые также провели ряд исследований, которые показали, что сетчатка глаза человека может меняться со временем, в то время как радужная оболочка глаза остается неизменной. И самое главное, что невозможно найти два абсолютно идентичных рисунка радужной оболочки глаза, даже у близнецов.

Для получения индивидуальной записи о радужной оболочке глаза черно-белая камера делает 30 записей в секунду. Еле различимый свет освещает радужную оболочку, и это позволяет видеокамере сфокусироваться на радужке. Одна из записей затем оцифровывается и сохраняется в базе данных зарегистрированных пользователей. Вся процедура занимает несколько секунд, и она может быть полностью компьютеризирована при помощи голосовых указаний и автофокусировки.

В аэропортах, например, имя пассажира и номер рейса сопоставляются с изображением радужной оболочки, никакие другие данные не требуются. Размер созданного файла, 512 байт с разрешением 640 x 480, позволяет сохранить большое количество таких файлов на жестком диске компьютера.

Очки и контактные линзы, даже цветные, никак не повлияют на процесс получения изображения. Также нужно отметить, что произведенные операции на глазах, удаление катаракты или вживление имплантатов роговицы не изменяют характеристики радужной оболочки, ее невозможно изменить или модифицировать. Слепой человек также может быть идентифицирован при помощи радужной оболочки глаза. Пока у глаза есть радужная оболочка, ее хозяина можно идентифицировать.

Голосовая биометрия, позволяющая измерять голос каждого человека, незаменима при удаленном обслуживании клиентов, когда основным средством взаимодействия является голос, в первую очередь, в автоматических голосовых меню и контакт-центрах.

Традиционные способы аутентификации клиента при удаленном обслуживании проверяют знания клиента (для этого клиента просят ввести какой-то пароль или ответить на вопросы безопасности - адрес, номер счета, девичью фамилию матери и пр.) Как показывают современные исследования в области безопасности, злоумышленники относительно легко могут добыть персональные данные практически любого человека и таким образом получить доступ, например, к его банковскому счету. Голосовая биометрия решает эту проблему, позволяя при удаленном телефонном обслуживании проверить действительно личность клиента а не его знания. При использовании голосовой биометрии клиенту при звонке в IVR или в контакт-центр достаточно произнести парольную фразу или просто поговорить с оператором (рассказать о цели звонка) - голос звонящего будет автоматически проверен.

Классическая верификация (идентификация) человека по почерку подразумевает сличение анализируемого изображения с оригиналом. Именно такую процедуру проделывает например оператор банка при оформлении документов. Очевидно, что точность такой процедуры, с точки зрения вероятности принятия неправильного решения (см. FAR & FRR) невысокая. Кроме этого, на разброс значений вероятности принятия правильного решения оказывает и субъективный фактор.

Принципиально новые возможности верификации по почерку открываются при использовании автоматических методов анализа почерка и принятия решения. Данные методы позволяют исключить субъективный фактор и значительно снизить вероятность ошибок при принятии решения (FAR & FRR).

Одним из факторов, которые определяют преимущество автоматических методов идентификации путем анализа почерка по сравнению с классическими методами верификации, является возможность использования динамических характеристик почерка. Автоматические методы идентификации позволяют принимать решение не только путем сличения изображения верифицируемого и контрольного образца, но и путем анализа траектории и динамики начертания подписи или любого другого ключевого слова [1].

В свободном доступе можно найти информацию лишь о некоторых проектах.

В Курском государственном университете под руководством Леонида Крыжевича, кандидата технических наук, доцента кафедры математического анализа и прикладной математики разрабатывается программа с подобным функционалом. По словам ученого, путем сложных математических преобразований, основанных на теории вероятности и статистике, коллективу КГУ удалось разработать программный продукт, который идентифицирует пользователя компьютера с точностью до 96%. Для успешной работы программы, по словам Крыжевича, достаточно набора четырех-шести символов. Ожидается, что демонстрационный программный продукт появится на рынке уже в следующем году [2].

DARPA Active Authentication Program: Behavioral Biometrics также схожий по функционалу продукт. В 2013 году на RSA Conference Asia Pacific были представлены некоторые его концепты и результаты его тестирования. Также были приведены математические формулы расчета и анализа «доверия» анализируемому пользователю. Данный программный продукт опирается на большое количество разнообразных критериев оценки и поведенческих параметров. По результатам тестирования, программа может определить неверного пользователя в течении 10 секунд использования компьютера (6 взаимодействий, в среднем 3 нажатия клавиш) или же менее чем за 3,5 минут использования мыши [3].

SilentSense, мобильное приложение, считывающее действия пользователя за устройством и с высокой точностью определяющее, является ли он владельцем телефона. Считывая нажатия на экран, различные движения, и показания акселерометра, программа за 10 действий может определить валидность пользователя с вероятностью ошибки меньше 1%. К сожалению, данный программный продукт не подходит для персональных компьютеров, но использованные в нем математические функции для расчета вероятности и «доверия» к пользователю, лежащие в свободном доступе, могут прилично поспособствовать разработке системы идентификации [4].

Разрабатываемое нами приложение предназначено для идентификации пользователя на базе поведенческой биометрии. Оно наглядно показывает работоспособность системы идентификации пользователя на базе поведенческой биометрии, позволяет создать профиль пользователя с параметрами поведенческой биометрии и в дальнейшем использовать его для анализа.

ЛИТЕРАТУРА

1. Свободная энциклопедия – Википедия [Электронный ресурс] / Распространенные технологии биометрии - Режим доступа: https://en.wikipedia.org/wiki/Biometrics#Adaptive_biometric_systems. Дата доступа: 10.09.15.
2. Интернет-портал Хакер [Электронный ресурс]/ Человека можно идентифицировать после 4-6 нажатий клавиш - Режим доступа: <https://xaker.ru/2014/12/18/keyboard-id/> . Дата доступа: 12.09.15.
3. Официальный сайт RSA Conference [Электронный ресурс]/ DARPA Active Authentication Program - Режим доступа: <http://www.rsaconference.com/>. Дата доступа: 12.09.15.
4. Свободная энциклопедия - Cornell University Library [Электронный ресурс]/ SilentSense - Режим доступа: <http://arxiv.org/pdf/1309.0073.pdf> Дата доступа: 12.09.15.

УДК 004.492

АНАЛИЗ ПРИНЦИПОВ ВНЕДРЕНИЯ, НАЗНАЧЕНИЯ, УПРАВЛЕНИЯ ПРОГРАММНЫХ ЗАКЛАДОК. ХАРАКТЕРНЫЕ ПРИЗНАКИ И КЛАССИФИКАЦИЯ ПРОГРАММНЫХ ЗАКЛАДОК

А.Е. РАМАШКА

(Представлено: канд. техн. наук, доц. К.Я. РАХАНОВ)

Рассмотрена возможная классификация программных закладок по принципам внедрения, назначения и управления. Проанализированы характерные признаки программных закладок.

Программная закладка – это внесенные в программное обеспечение функциональные объекты, которые при определенных условиях (входных данных) инициируют выполнение не описанных в документации функций, позволяющих осуществлять несанкционированные воздействия на информацию (недокументированные возможности программного обеспечения) [1].

В настоящее время проблема программных закладок является актуальной и требует немедленного решения. Таким образом, возникает потребность в методах и программных продуктах, способных успешно противодействовать подобным угрозам. Современное предприятие представляет собой сложноорганизованную систему со своими управляющими органами и объектами информатизации. Поскольку любое современное рабочее место уже трудно представить без компьютера, которые практически всегда объединены в общую вычислительную сеть, необходимо полностью обезопасить их от возможных утечек секретных данных и конфиденциальной информации.

Как правило, многие крупные компания скрывают факт взлома и получения конфиденциальной информации злоумышленниками. Но даже без этой большой части всеобщей статистики, можно провести анализ известных кибер-атак, которые, к слову были произведены с помощью программных закладок.

Самым ярким примером действия программной закладки является военный конфликт в Персидском заливе. Тогда при проведении операции «Буря в пустыне» система ПВО Ирака оказалась заблоки-

рованной по неизвестной причине. Несмотря на отсутствие исчерпывающей информации, высказывалось предположение, что ЭВМ, входящие в состав комплекса технических средств системы ПВО, закупленные Ираком у Франции, содержали специальные управляемые «электронные закладки», блокировавшие работу вычислительной системы путем воздействия извне защищаемого объекта [2].

Периодически эксперты обнаруживают недокументированные возможности в ПО для аппаратных комплексов. Например, в информационном бюллетене CERT CA-2002-32 объявлено о существовании программной закладки (backdoor) в операционной системе AOS (Alcatel Operating System) версии 5.1.1, применяемой для управления коммутаторами Alcatel OmniSwitch 7700/7800. Эта закладка запускает telnet-сервис на порту 6778/TCP, что может позволить удаленному злоумышленнику неавторизованно управлять коммутатором [3].

Есть случаи, когда уволенные программисты похищали корпоративные данные через оставленный собой же «черный ход» (backdoor) в программе или нарушали работоспособность [4].

Присутствие программных закладок обнаруживается по следующим признакам:

1. Наличие в исходном коде исполняемого файла алгоритмического описания функций, назначение которых явно не декларировано разработчиком;
2. Изменение структуры исполняемого файла;
3. Изменение цифровой подписи исполняемого файла;
4. Обращение к области памяти другого процесса, изменение и уничтожение данных в области памяти другого процесса;
5. Обращение к внешним устройствам хранения информации и файловой системе в целом;
6. Инициализация сетевых подключений;
7. Аномальная сетевая активность;
8. Аномальное потребление аппаратных ресурсов.

Таким образом, можно выделить основные классификационные признаки программных закладок:

1. Метод внедрения: программные закладки могут внедряться на этапе разработки и при уже использовании готового продукта путем инъекций кода в исполняемые файлы либо обновления программного обеспечения до новой версии;
2. Назначение: программные закладки предназначаются для копирования, изменения, уничтожения конфиденциальной информации или внесения изменений в алгоритмы работы программного обеспечения. Так же возможен сценарий использования программных закладок для нарушения работоспособности информационных систем;
3. Метод управления: срабатывание программной закладки может быть привязано к какому либо событию в системе, в которую она внедрена, либо срабатывание может быть вызвано удаленно, если иницируются сетевые подключения.

На основании вышеперечисленных признаков, можно предложить классификацию программных закладок:

1. Классификация по методу внедрения:
 - 1.1. Внедрение на этапе проектирования программного обеспечения;
 - 1.2. Внедрение в готовое программное обеспечение.
2. Классификация по назначению:
 - 2.1. Копирование конфиденциальной информации;
 - 2.2. Изменение конфиденциальной информации;
 - 2.3. Уничтожение конфиденциальной информации;
 - 2.4. Изменение алгоритмов работы программного обеспечения;
 - 2.5. Нарушение работоспособности системы в целом.
3. Классификация по методу управления:
 - 3.1. Автоматическое срабатывание;
 - 3.2. Удаленный вызов срабатывания.

Вывод. Используя предложенную классификацию и программных закладок, возможно создание эффективной системы борьбы с программными закладками и не декларированными возможностями программного обеспечения. Детальная классификация позволяет разработать точные алгоритмы для обнаружения и минимизации воздействия программных закладок.

ЛИТЕРАТУРА

1. Шабанов, И. Технологии и бизнес [Электронный ресурс] / И. Шабанов. – 2011. – Режим доступа: http://miratech.ua/sites/default/files/documents/press_about_us/pau20110427article1.pdf. – Дата доступа: 17.09.2015.

2. Лукашкин, К.А. Проблемы безопасности программного обеспечения военной техники [Электронный ресурс] / К.А. Лукашкин // Военное дело. – 2001. – Режим доступа: http://www.soldiering.ru/psychology/safe_soft_militaryequip.php. – Дата доступа: 17.09.2015.
3. Ian A. Finlay. Backdoor in Alcatel OmniSwitch AOS [Электронный ресурс] / Finlay // Carnegie Mellon University. – 2015. – Режим доступа: <http://www.cert.org/historical/advisories/CA-2002-32.cfm>. – Дата доступа: 20.09.2015.
4. Группа компаний InfoWatch [Электронный ресурс] / InfoWatch. – 2015. – Режим доступа: https://www.infowatch.ru/sites/default/files/files/products/appercut/Appercut_Company_Brochure_Rus.pdf. – Дата доступа: 16.09.2015.

УДК 004.42

ФУНКЦИОНАЛЬНАЯ СТРУКТУРА СИСТЕМЫ БОРЬБЫ С АНОМАЛЬНОЙ АКТИВНОСТЬЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.Е. РАМАШКА

(Представлено: канд. техн. наук, доц. К.Я. РАХАНОВ)

Рассмотрена возможная функциональная структура системы для борьбы с аномальной активностью программного обеспечения с применением облачных технологий.

В последнее время, в связи с ростом скорости внедрения информационных технологий в экономику и промышленность, наиболее остро встала проблема защиты конфиденциальных и секретных данных, представляющих определенную ценность, будь то определенные технологии производства или же данные о клиентах.

С точки зрения информационных технологий особую опасность представляют сами средства вычислительной техники в первую очередь компьютеризированные рабочие места сотрудников предприятия [1]. Одним из способов получения несанкционированного доступа к информации является внедрение в программное обеспечение не декларированных возможностей, которые порождают аномальную активность программного обеспечения.

Автоматизированная система должна представлять собой программный комплекс для выявления аномальной активности программного обеспечения, а также устранения возможности изменения информации и несанкционированного доступа к ней.

Система обнаружения аномальной активности должна удовлетворять следующим требованиям:

- 1) анализировать поведение исследуемого процесса (приложения);
- 2) возможность сигнатурного анализа исполняемого файла исследуемого процесса;
- 3) возможность анализа сетевых соединений, открываемых процессом;
- 4) возможность анализа сетевого трафика, генерируемого приложением;
- 5) реализация принятия решений оператором и в автоматическом режиме;
- 6) отвечать требованиям отказоустойчивости (сбой работы элемента системы, атаки на злоумышленников на ресурсы системы);
- 7) сохранение высокой производительности при пиковых нагрузках;
- 8) балансировка нагрузки между несколькими модулями обнаружения угроз;
- 9) низкая нагрузка на аппаратные мощности и ресурсы вычислительной машины;
- 10) поддержка облачных вычислений.

Исходя из вышеперечисленных требований система состоит из компонентов:

- 1) подсистема получения информации о сетевой активности приложений;
- 2) подсистема перехвата сетевых пакетов;
- 3) подсистема контроля целостности системы;
- 4) подсистема обнаружения угроз;
- 5) подсистема автоматического реагирования;
- 6) подсистема распределения нагрузки;
- 7) подсистема удаленного управления;
- 8) подсистема администрирования;
- 9) подсистема авторизации.

Подсистема получения информации о сетевой активности предназначена для постоянного мониторинга всех процессов на компьютере пользователя, а также получения данных о сетевых подключениях, инициированных этими процессами.

Подсистема перехвата сетевых пакетов предназначена для сбора данных о всех сетевых пакетах, генерируемых или получаемых процессами на компьютере пользователя.

Подсистема контроля целостности системы предназначена для мониторинга состояния системы. В случае аварийного или несанкционированного завершения работы одного из компонентов системы, данная подсистема оперативно восстанавливает работоспособность измененного файла.

Подсистема обнаружения угроз – ядро всей системы, которое отвечает за анализ событий, собранных подсистемой сетевой активности приложений и подсистемой перехвата сетевых пакетов.

Подсистема автоматического реагирования отвечает за действия, предпринимаемые по результатам анализа подсистемы обнаружения угроз.

Подсистема распределения нагрузки отвечает за распределение задач анализа и принятия решений равномерно между всеми компьютерами в сети в ситуации, когда серверу не хватает аппаратных ресурсов либо выхода его из строя. Данная подсистема автоматически, при нагрузке, многократно превышающей ту, которую может выдержать основной анализатор, распределяет ее по простаивающим клиентским машинам. Балансировка происходит следующим образом: если загрузка центрального процессора сервера больше 80%, подается команда передачи управления всем управляющим службам на клиентских машинах. Управляющая служба вычисляет процент использования центрального процессора и количество свободной оперативной памяти и записывает данное значение в базу данных. Эмпирическим путем была получена формула вычисления количества пользователей, которое в данный момент возможно анализировать:

$$N = RT \cdot \frac{PU_e}{PU_c} \cdot \frac{RA_e}{RA_c}, \quad (1)$$

где PU_e и RA_e – экспериментально полученные значения использования процессора в процентах и количество свободной оперативной памяти, а PU_c и RA_c – текущие значения тех же параметров, RT – количество потоков выполнения центрального процессора.

Подсистема удаленного управления отвечает за автоматическое выполнение команд по блокировке приложений или сетевого трафика со стороны подсистемы автоматического реагирования.

Подсистема администрирования предназначена для возможности функционирования системы в полуавтоматическом режиме, путем выполнения команд администратором систем или офицером безопасности по блокировке приложений или сетевого трафика в зависимости от возникшего события.

Подсистемы обнаружения угроз и принятия решения, выполняясь на удаленном хосте, реализуют концепцию облачных вычислений, пример которых представлен в работе [2], тем самым значительно снижая нагрузку и потребление аппаратных ресурсов, возникающие при анализе.

Таким образом, функциональную структуру системы можно представить в виде схемы, представленной на рисунке.

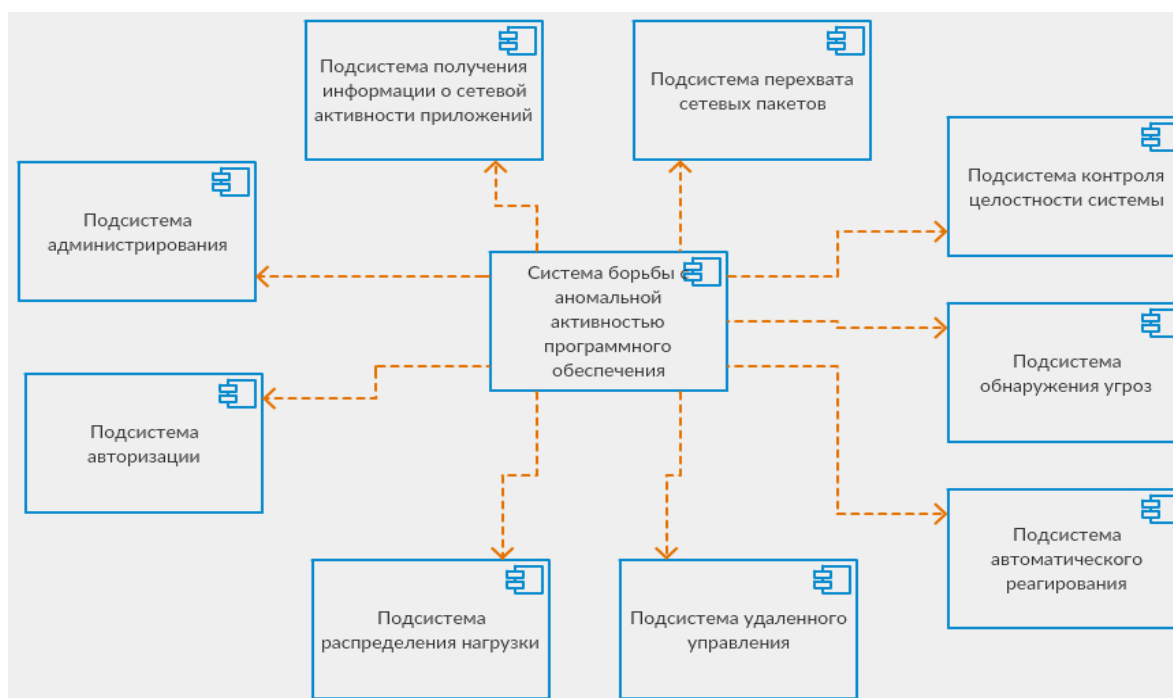


Рис. Функциональная структура автоматизированной системы

Представленная структура описывает базовый функциональный набор и может расширяться в зависимости от специализации.

Выводы. Предложена функциональная структура системы борьбы с аномальной активностью программного обеспечения, которая отличается повышенной надежностью и производительностью за счет применения распределенных вычислений и облачных технологий. Главной отличительной особенностью данной структуры является модульность, благодаря которой, при минимальных затратах, возможно интегрирование дополнительных модулей анализа и реагирования.

ЛИТЕРАТУРА

1. Министерство внутренних дел Республики Беларусь [Электронный ресурс] / Первое российское исследование скрытых угроз, 2011. Режим доступа: <http://mvd.gov.by/main.aspx?guid=55533> – Дата доступа: 16.09.2015.
2. Орлов С. Облачные сервисы: безопасность и надежность / Журнал сетевых решений №12, 2012 – 10с.

УДК 681.586.773:624.072.233.5

ПРИМЕНЕНИЕ НЕРАЗРУШАЮЩЕЙ ДЕФЕКТΟΣКОПИИ ДЛЯ КОНТРОЛЯ КАЧЕСТВА ЖЕЛЕЗНОДОРОЖНЫХ РЕЛЬСОВ

П.В. СТЕПАНОВ

(Представлено: канд. техн. наук, доц. Д.А. ДОВГЯЛО)

Рассмотрены наиболее распространенные дефекты, проявляющиеся в процессе производства и эксплуатации железнодорожных рельсов, установлены причины их возникновения. Дана классификация методов неразрушающего контроля, применяемых для бесконтактного или контактного диагностирования скрытых дефектов. Приведено описание оборудования, используемого при ультразвуковой дефектоскопии.

Введение. Неразрушающий контроль (НК) – это контроль, который не разрушает (именно такое определение дано в ГОСТ 16504-81[1]). Под словом «контроль» подразумевается «измерение значений рабочих параметров и свойств объекта и их проверка на соответствие допустимым величинам». «Неразрушающий» означает «не требующий демонтажа или остановки работы объекта», «не подразумевающий непосредственного вмешательства в исследуемую среду». Методы, с помощью которых реализуется НК, называются методами неразрушающего контроля.

Обзор существующих методов НК. На сегодняшний день существует большое разнообразие методов неразрушающего контроля. К ним относятся:

- магнитные методы НК;
- электрические методы НК;
- вихретоковые методы НК;
- радиоволновые методы НК;
- тепловые методы НК;
- оптические методы НК;
- радиационные методы НК;
- акустические методы НК;
- методы НК проникающими веществами.

Данные методы получили развитие в различных отраслях промышленности, использующих НК для диагностики различного рода конструкций, узлов и механизмов. Разновидностью акустических методов НК являются методы, реализованные на основе использования ультразвуковых волн. Ультразвук нашел широкое применение как в научных исследованиях, так и в промышленной технологии после практической реализации надежных способов его возбуждения. В последнее время наряду с традиционной дефектоскопией возрос интерес к ультразвуку как средству контроля физических и механических свойств материалов. При этом если упругие свойства твердых тел аналитически связаны с параметрами упругих волн и могут надежно определяться на основании акустических измерений, то прочностные

свойства, напротив, не имеют такой связи и могут контролироваться лишь на основе корреляционных связей, определяемых опытным путем [2].

Классификация дефектов рельсов. Причины их развития. Одной из областей, в которой применяются акустические методы контроля, является ультразвуковая рельсовая дефектоскопия. Ультразвуковая дефектоскопия рельсов – метод НК, выявляющий внутренние дефекты рельсов и их структурную неоднородность (трещины, неметаллические включения и пр.).

Причины зарождения и развития дефектов в рельсах можно условно разделить на две группы: эксплуатационные, например, неудовлетворительное состояние пути и подвижного состава и заводские – из-за наличия дефектов, допущенных при изготовлении рельсов.

Неудовлетворительное состояние пути и подвижного состава способствует ускорению и развитию дефектов, допущенных по заводским причинам. На участках с неудовлетворительным содержанием болтовых стыков воздействие поезда нагрузки в разы больше, чем при удовлетворительном их состоянии. Количество изломов рельсов в стыках в значительной степени зависит от срока службы рельсов. В первые годы эксплуатации рельсов изломы в стыках встречаются крайне редко. С увеличением срока службы этот дефект становится все более распространенным, а для старых рельсов – типичным. Резкое увеличение выхода рельса из строя по дефектам стыков наблюдается весной – во время оттаивания балласта, а также осенью – в период замерзания балласта. На поверхности катания головки рельса наиболее распространены следующие дефекты:

- отслоение и выкрашивание металла по поверхности катания;
- выкрашивание на боковой выкружке головки рельса;
- выбоксовины и закалочные трещины в местах буксования колес;
- выкрашивание и отслоение металла на поверхности в закаленном слое головки рельса.

Отслоение и выкрашивание металла по поверхности катания проявляются обычно при нарушении технологии изготовления рельсов. Такие повреждения успешно поддаются выявлению при визуальном осмотре.

Выкрашивание металла на рабочей боковой выкружке головки рельса происходит чаще всего из-за недостаточной контактной прочности рельса при загрязненности металла неметаллическими включениями. Одной из причин зарождения здесь микротрещин, кроме воздействия подвижного состава, могут быть удары при выгрузке рельсов и производстве путевых работ.

Выбоксовины и закалочные трещины в местах буксования колес образуются на рельсах, расположенных перед входными сигналами на станционных путях, на тормозных участках пути.

Выкрашивание и отслоение металла на поверхности закаленного слоя головки рельса возникают и развиваются из-за неудовлетворительного качества закалки рельсов. Аналогичные повреждения могут быть у концов рельсов из-за недоброкачественной наплавки.

Поперечные трещины усталости в головке рельса и изломы из-за них – наиболее опасный дефект контактно-усталостного происхождения. Он возникает и развивается внутри головки рельса. Существует две основные причины его зарождения и развития. Первая – наличие микроскопических надрывов (флокенов) внутри головки рельса, которые обычно располагаются на глубине не более 10 мм от поверхности катания рельса. Вторая причина – малоразвитые продольно-наклонные трещины. Они возникают при больших контактных напряжениях на поверхности катания головки рельса при воздействии подвижного состава по мере накопления циклов нагружения. Поперечные трещины в головке возникают также из-за различных механических повреждений: при проходе колес с большими ползунами; от удара по головке рельса путевым инструментом, рельса о рельс. На поверхности катания рельса появляются насечки, которые концентрируют напряжения и могут стать причиной образования трещин даже при нормальной нагрузке.

Вертикальные расслоения головки и шейки рельса являются опасными дефектами. Причинами их возникновения являются остатки усадочной раковины в рельсе, неметаллические включения (скопления примесей серы и фосфора). Развитие таких дефектов ускоряется при неудовлетворительном содержании пути. Горизонтальные расслоения головки рельса обычно возникают и развиваются в средней части по высоте головки в любом месте по длине рельса. Основная их причина – загрязнение металла неметаллическими включениями, газовые пузыри и флокены, вытянутые вдоль прокатки.

Большое распространение имеют дефекты смятия головки. Смятие наблюдается как в стыках, так и по всей их длине. Наиболее часто сминается головка стыковой части рельсов. В результате сжатия рельсов на наружных боковых гранях образуются наплывы с равными отслоениями и трещинами. Такие наплывы не являются опасными дефектами, но под ними часто скрываются опасные трещины в недеформированной части рельса. Смятие рельсов связано со сроком их службы.

В шейке рельса (вне стыковой части) косые или продольные трещины возникают и развиваются в местах маркировочных знаков или повреждений от ударов путевым инструментом.

Продольные трещины в местах перехода головки в шейку и шейки в подошву являются обычно очень тонкими и в начальной стадии развития представляют собой серию небольших надрывов. По мере развития надрывы соединяются и образуют под головкой длинную тонкую трещину, иногда заходящую в головку. Как правило, эти трещины развиваются после многих лет эксплуатации рельсов. Это трещины коррозионной усталости, возникающие при концентрации напряжений в месте сопряжения шейки с головкой. В подошве рельса дефекты могут зарождаться и развиваться из-за коррозионных процессов вследствие накопления влаги между подошвой и элементами скрепления, что приводит к образованию каверн, концентрирующих напряжение и впоследствии – поперечных трещин усталости. Опасность таких трещин велика, так как они находятся в растянутой зоне, что приводит к излому рельса даже при небольших по площади трещинах.

Поперечные трещины в подошве рельса возникают также из-за различных механических повреждений. При ударах инструментом, рельса о рельс и при других механических воздействиях на поверхности появляются насечки – концентраторы напряжений. Поперечные изломы рельсов могут происходить из-за наличия шлаковых или других посторонних включений при их изготовлении, вследствие прохода колес с большими ползунами и выбоинами. Трещины в сварных стыках возникают из-за наличия в сварных соединениях силикатных включений, пузырей, рыхлостей, непроваров, кратерных усадок и неудовлетворительной обработки сварного шва. При эксплуатации сварных стыков в пути под воздействием знакопеременных нагрузок наиболее часто развиваются поперечные трещины в подошве [3].

Оборудование, используемое в УЗД рельсов. На рисунке 1 представлен внешний вид ультразвукового дефектоскопа УДС-2-РДМ-22.

Дефектоскоп является переносной механизированной системой ультразвукового контроля с использованием эхо-метода и зеркально-теневого метода при контактном способе ввода ультразвуковых колебаний (УЗК). Количество каналов возбуждения и приема УЗК, предусмотренных для работы с блоками пьезоэлектрических резонаторов в режиме сплошного контроля, для каждой из нитей пути – 8. Количество каналов возбуждения и приема УЗК, предусмотренных для работы с ручными пьезоэлектрическими преобразователями – 3. Сигнализация о наличии дефектов – звуковая и цветовая по экрану матричного индикатора, предусмотрена индикация установленных значений условной чувствительности каналов контроля (dB), коэффициента выявляемости дефекта (dB), координат дефектов (mm), текущей путевой координаты контролируемого участка пути - цифровая на экране матричного индикатора.

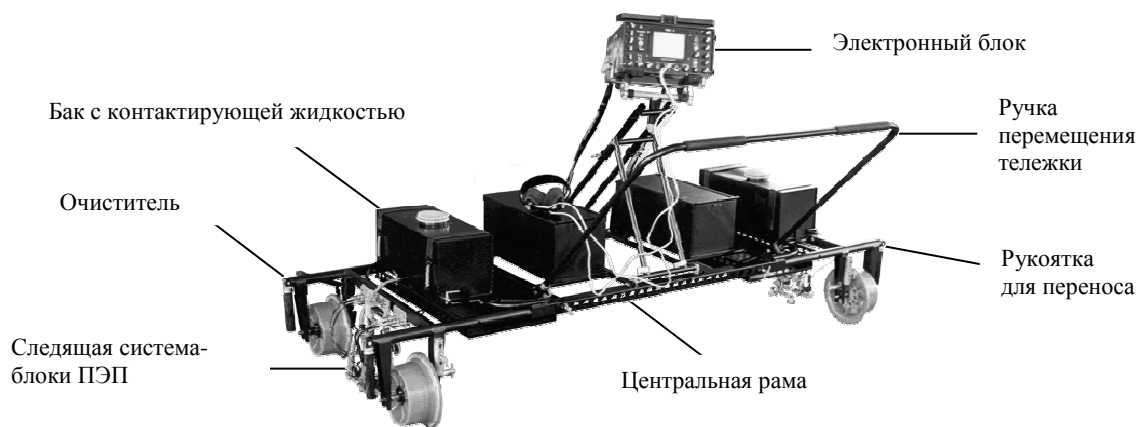


Рис. 1. Внешний вид ультразвукового дефектоскопа УДС-2-РДМ-22

Предусмотрено непрерывное документирование (регистрация) результатов контроля в виде дефектограмм в формате В-сканов проконтролированных сечений рельсов, регистрируемых через 1 мм пути, с сопроводительной информацией об амплитудно-временных характеристиках зарегистрированных эхо-сигналов, установленной в каждом из каналов условной чувствительности контроля и текущих значений путевой координаты [4].

На рисунке 2 показан ультразвуковой дефектоскоп УДС2-РДМ-33.



Рис. 2. Внешний вид дефектоскопа УДС-2-РДМ-33

Дефектоскоп УДС-2-РДМ-33 предназначен для ультразвукового контроля зон сварки стыковых, угловых, нахлесточных и тавровых соединений, выполненных электродуговой, электрошлаковой, газовой, газопрессовой, электронно-лучевой и стыковой сваркой, оплавлением в конструкциях из углеродистых и легированных сталей и сплавов, в том числе в железнодорожных рельсах для выявления трещин, непроваров, пор, неметаллических и инородных металлических включений.

Преимущества:

- хорошая воспроизводимость результатов контроля в широком диапазоне воздействия климатических факторов;
- устойчивость к воздействию дестабилизирующих и разрушающих факторов, возникающих в процессе эксплуатации;
- сочетание встроенных программ для решения типовых задач контроля с гибкостью регулировки прибора для оптимальной настройки на решение конкретной задачи;
- возможность документирования результатов контроля с последующим просмотром протоколов контроля с осциллограммами сигналов в развертках типа А, О и В на экране дефектоскопа или мониторе персонального компьютера;
- наличие миниатюрных датчиков для удобства контроля проблемных зон сварки;
- использование в автономном источнике питания герметичного кислотного аккумулятора, не требующего присмотра при его зарядке, т.к. отсутствует риск перезарядки при длительном подключении его к сетевому блоку питания [5].

ЛИТЕРАТУРА

1. Система государственных испытаний продукции. Испытания и контроль качества продукции. Основные термины и определения : ГОСТ 16504-81. – (Нац. система подтверждения соответствия Респ. Беларусь. Порядок декларирования соответствия продукции. Основные положения).
2. Методы проведения неразрушающего контроля [Электронный ресурс]. – 2012. – Режим доступа: http://www.devicesearch.ru/article/metody_nerazgushayuschego_kontrolya. – Дата доступа: 29.05.2015.
3. Катен-Ярцев, А.С. Методы рельсовой дефектоскопии : учеб. пособие / А.С. Катен-Ярцев. – Хабаровск : Изд-во ДВГУПС, 2011. – С. 4–8.
4. Дефектоскоп ультразвуковой ДС2-РДМ-22. Руководство по эксплуатации. – (Нац. система подтверждения соответствия Респ. Беларусь. Порядок декларирования соответствия продукции. Основные положения).
5. Дефектоскоп ультразвуковой УДС2-РДМ-33. Руководство по эксплуатации. – (Нац. система подтверждения соответствия Респ. Беларусь. Порядок декларирования соответствия продукции. Основные положения).