

Федеральное государственное автономное
образовательное учреждение
высшего образования
"СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ"

Институт космических и информационных технологий
институт

Кафедра вычислительной техники
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ВТ

_____ А.И.Легалов

"__" _____ 20__ г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Подсистема формирования графических образов
тема

для системы активного тестирования

09.04.01 Информатика и вычислительная техника
код и наименование направления

09.04.01.02 Информационное и программное обеспечение САПР
код и наименование магистерской программы

Руководитель	_____	проф., д-р. тех. наук	С.А.Бронов
	<i>подпись, дата</i>	<i>должность, ученая степень</i>	<i>инициалы, фамилия</i>
Выпускник	_____	_____	Д.С.Степанов
	<i>подпись, дата</i>		<i>инициалы, фамилия</i>
Рецензент	_____	доц., канд. тех. наук	Ю.В.Краснобаев
	<i>подпись, дата</i>	<i>должность, ученая степень</i>	<i>инициалы, фамилия</i>
Нормоконтролер	_____	доц., канд. тех. наук	В.И.Иванов
	<i>подпись, дата</i>	<i>должность, ученая степень</i>	<i>инициалы, фамилия</i>

Красноярск 2017

АННОТАЦИЯ

Магистерская диссертация "Подсистема формирования графических образов для системы активного тестирования" выполнена в НУЛ САПР СФУ.

Объект и предмет исследования — формат представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.

Цель работы — разработка и программная реализация система формирования графических образов для системы активного тестирования.

На языке программирования Java с использованием технологии JavaFX разработано приложение редактор графических образов для системы активного тестирования.

СОДЕРЖАНИЕ

Введение.....	5
1 Проблематика тестирования, цель и задачи работы.....	10
1.1 Обзор существующих электронных обучающих курсов и их систем тестирования.....	10
1.2 Проблематика тестирования.....	23
1.3 Требования к системе формирования графических образов для системы активного тестирования.....	25
1.4 Требования к формату представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.....	29
1.5 Цели и задачи работы.....	30
1.6 Средства разработки.....	31
1.6.1 Java.....	31
1.6.2 JavaFX.....	32
1.6.3 XML.....	34
1.6.4 XSD.....	35
Выводы по разделу 1.....	35
2 Разработка формата представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.....	37
2.1 Представление данных.....	37
2.2 Графические примитивы.....	37
2.3 Вспомогательные элементы.....	44
2.4 Правила описания элементов ("element.xsd").....	50
2.5 Правила описания объекта задания.....	52
Выводы по разделу 2.....	58
3 Разработка системы формирования графических образов для системы активного тестирования.....	59
3.1 Разработка структуры базы данных.....	59

3.2 Разработка взаимодействие с веб-сервером.....	60
3.3 Разработка пользовательского интерфейса редактора графических образов.....	63
Выводы по разделу 3.....	65
Заключение	67
Список сокращений	68
Список использованных источников	69
ПРИЛОЖЕНИЕ А (Обязательное) Исходный текст примера экспорта графического образа делителя напряжения в формат SVG в веб-приложении Draw.io.....	77
ПРИЛОЖЕНИЕ Б Руководство по интерфейсу веб-сервиса анализа корректности выполнения графических тестовых заданий.....	92
ПРИЛОЖЕНИЕ В (Обязательное) Листинг набора XSD-схем для формата представления графических объектов для системы анализа корректности выполнения графических тестовых заданий	80
ПРИЛОЖЕНИЕ Г (Обязательное) Исходный код метода loadDBfromServer() класса DBManager, реализующего синхронизацию локальной базы данных с базой данных веб-сервиса.	88

ВВЕДЕНИЕ

Актуальность. Современное развитие информационных технологий предоставляет большие возможности для автоматизации процессов, протекающих в вузах. Одним из примеров автоматизации процессов в вузе является автоматизация такой части учебного процесса, как контроль усвоенных знаний у студентов — тестирование.

Применение тестирования возможно, как для текущего, промежуточного, так и итогового контроля усвоенных знаний, что позволяет получить более объективную их оценку[27;47]. Актуализация проблемы автоматизации составления учебных планов(УП) в вузах и способы ее решения[1;49] позволяет говорить о том, что тестирование будет являться одним из основных способов контроля знаний студентов.

В настоящее время на коллективном НУЛ САПР СФУ ведутся работы над методом автоматизации синтеза учебных планов на основе массива дидактических единиц(ДЕ)[1] и основанной на нем активной системе тестирования. Разрабатываемая система активного тестирования, основанная на предложенном методе формирования тестовых вопросов при построении ДЕ, позволит получать более точные данные об уровне усвоения студентами ДЕ и расширить возможности студента в самоконтроле и работе над ошибками.

Учитывая формирование укрупненных ДЕ[0] из более мелких ДЕ, то при неверном ответе возможно выделять ту ДЕ, которую студент изучил недостаточно хорошо, что впоследствии позволит провести работу над ее дополнительным изучением, т. е. перейти от обычных тестов к интеллектуальным.

Разрабатываемая коллективом НУЛ САПР СФУ интеллектуальная система тестирования будет расширена возможностями предоставления вопроса как в виде схемы или графического образа, так и получения ответов в таком виде[58;59].

В рамках разработки метода создания тестовых заданий в виде схем и графических образов, и проверки такого типа информации поставлены следующие задачи:

- используя подготовленную элементную базу, сформировать графический образ схемы, установив параметры элементов и связи между ними;
- используя созданную схему, получать элементную базу, достаточную для решения задания;
- предоставлять информацию о схеме и ее элементной базе в формате, удобном для дальнейшего использования другими подсистемами.

То есть предполагается использование дополнительного программного обеспечения, реализующего требуемые функции.

Среди графических редакторов и приложений для создания электронных схем имеется множество приложений, решающих различные задачи: от редакторов печатных плат до сложных систем, поддерживающих функции моделирования процессов, протекающих в созданной схеме.

Такими образом, проблема заключается в том, что использование какого-либо готового продукта не является возможным, так как перед приложением стоят конкретные задачи, которые частично возможно решить существующими программными средствами, но применение их не является рациональным ввиду избыточного функционала или его недостаточности.

Объектом исследования является способ представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.

Предметом исследования формат представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.

Объектом разработки является программное обеспечение позволяющие конструировать графический образ задания и предоставлять информацию о нем в формате, удобном для дальнейшего использования.

Цель работы — разработка и реализация системы формирования графических образов для системы активного тестирования.

Задачи работы:

1 Определение функционала реализуемого приложения и требований к формату предоставляемых данных.

2 Выбор методов и инструментальных средств для разработки системы формирования графических образов.

3 Разработка формата представления графических данных для системы анализа корректности выполнения графических тестовых заданий.

4 Разработка системы формирования графических образов.

Основная идея работы. Разработка программы-редактора предоставляющего базовой функционал (редактирование элементов графического образа задания — добавление, удаление, установка или изменение параметров элементов, установка связей между элементами и редактирование этих связей, предоставление простого доступа к базовым элементам), и функционал, решающий поставленные задачи (преобразование графического образа задания и его элементной базы в формат удобный для дальнейшего использования).

Методы, инструментальные средства и технологии разработки.

Формат представления графических данных для системы анализа выполнения корректности графических тестовых заданий разработан с использованием расширяемого языка разметки XML.

Для разработки системы формирования графических образов использовался объектно-ориентированный язык программирования Java с использованием технологии JavaFX.

Значение для практики.

Разработанный формат представления графических данных, позволяет представить графический образ задания в текстовом варианте, для его более удобной обработки системой анализа выполнения корректности графических тестовых заданий.

Реализованная в виде приложения-редактора система формирования графических образов позволяет решить задачи поставленные в рамках разработки

метода создания тестовых заданий в виде схем и графических образов, которые не возможно решить существующими программными продуктами.

Особенности работы заключаются в том, что задача разработки метода создания тестовых заданий в виде схем и графических образов, и проверки такого типа информации, поставленная в рамках разрабатываемой коллективом НУЛ САПР СФУ системы активного тестирования, разбита над подсистемы:

- подсистема формирования графических элементов для системы активного тестирования;
- подсистема формирования графических образов для системы активного тестирования;
- подсистема анализа корректности выполнения графических тестовых заданий.

Разработка этих систем ведется параллельно.

В разделе 1 представлен обзор существующих систем дистанционного образования. Обозначена проблематика развития систем тестирования. Составлены требования, выбраны программные средства для разработки формата представления данных и системы формирования графических образов. Сформулированы цель и задачи работы.

В разделе 2 выполнена разработка формата представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.

В разделе 3 выполнена разработка системы формирования графических образов для системы активного тестирования.

Список использованных источников содержит 64 наименований.

В приложении А представлен исходный текст примера экспорта графического образа делителя напряжения в формат SVG в веб-приложении Draw.io.

В приложении Б представлен набор XSD-схем для формата представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.

В приложение В представлен исходный код метода, реализующего синхронизацию локальной базы данных с базой данных веб-сервиса.

В приложение Г приведено руководство по интерфейсу веб-сервиса разработанного в рамках системы анализа корректности выполнения графических тестовых заданий.

Использование результатов работы. Разработанное программное обеспечение может быть использовано при разработке материалов для активной системы тестирования графического материала

1 Проблематика тестирования, цель и задачи работы

1.1 Обзор существующих электронных обучающих курсов и их систем тестирования

ILIAS

ILIAS — свободная, с открытым исходным кодом, система управления обучением, базирующаяся на Apache, PHP, MySQL, XML. Отвечает стандарту SCORM. Система получила распространение в зарубежных ВУЗах, преимущественно немецких. Обусловлено это тем что, в русскоязычном сегменте интернета достаточно мало информации об ее установки и поддержки, переведенной на русский язык, хотя в системе имеется его поддержка.

Сама же документация о платформе достаточно объемно описывает все аспекты работы с системой и доступна на официальном сайте[25] на нескольких языках (немецкий, английский, испанский, французский, японский, польский). Помощь и ответы на вопросы также можно получить в тематических форумах на официальном сайте системы (требуется знание немецкого языка для более комфортного общения, и получение ответа в более короткие сроки).

На официальном сайте ILIAS есть ссылка на веб-ресурс с установленной системой в демонстрационном режиме[4] позволяющий познакомиться с системой в роли преподавателя или студента (Рисунок 1.1).

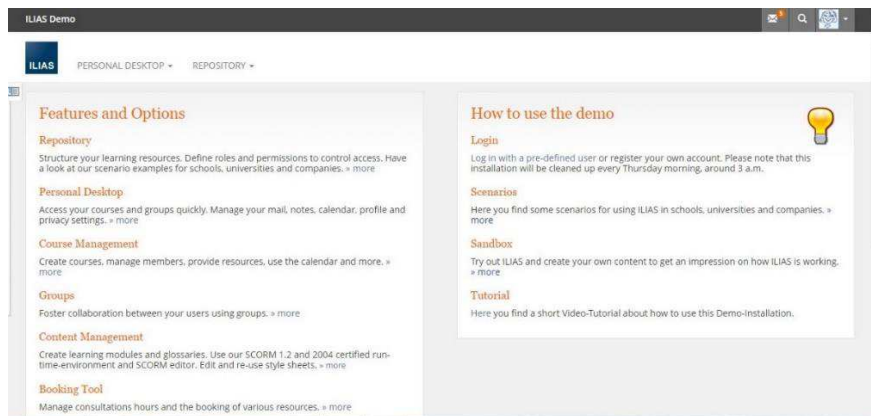


Рисунок 1.1 — Демонстрационный режим ILIAS

Система поддерживает форматы аудио и видео, которые можно использовать для предоставления лекционного материала наряду с обычным текстовым форматом.

Для взаимодействия между пользователями, ILIAS поддерживает возможность обмена сообщениями при помощи почтовых ящиков, форумов, комментариев. Каждому пользователю доступно свое рабочее пространство на котором отображаются текущие обновления курсов, календарь, системные сообщения, почта, новости (Рисунок 1.2).

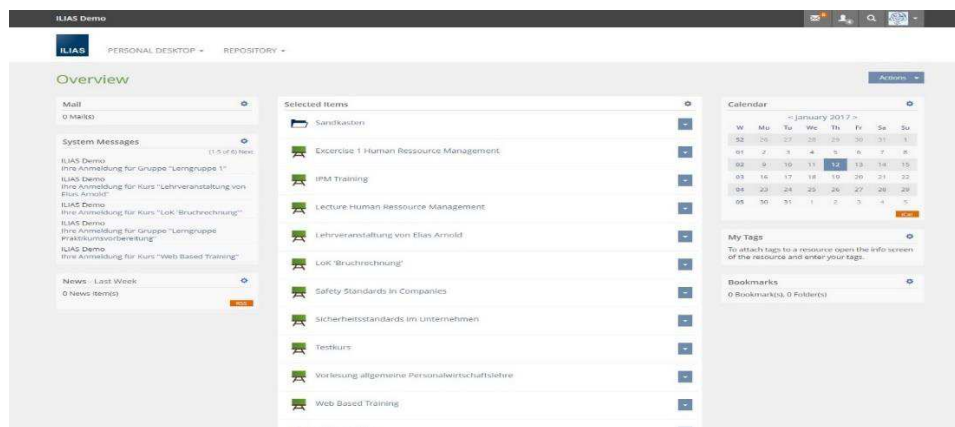


Рисунок 1.2 — Рабочее пространство пользователя

Кроме того, каждому пользователю доступен свой репозиторий через который можно обмениваться информацией (Рисунок 1.3).

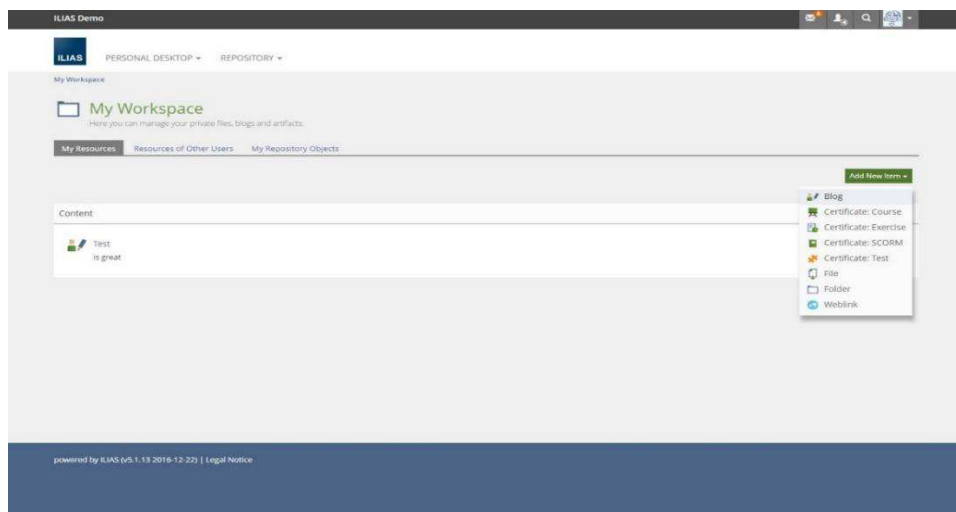


Рисунок 1.3 — Страница репозитория пользователя

Использование этих средств коммуникации позволяют обмениваться мнениями о том или ином учебном материале или курсе в целом, налаживать взаимодействие между учащимися и преподавателем.

ILIAS предоставляет несколько способов управления процессом обучения. Учебные объекты (лекции, задания, тесты и т.д.) могут быть доступны при заранее определенных условиях, например, студент должен выполнить одно или несколько условий, прежде чем получить доступ к новому учебному объекту. Кроме того, можно создавать учебные курсы на основе оценки знаний каждого ученика.

Для тестирования и работы с результатами ILIAS предлагает интегрированные средства. Данные о результатах тестирования могут быть использовано для мониторинга прогресса изучения материала, как студентами, так и преподавателями. Тестирование поддерживает следующие типы вопросов:

- множественный выбор;
- единичный выбор;
- упорядочивание ответов;
- свободный ввод;
- графический вопрос (поиск ответа на активном изображении).

Вопросы хранятся в банке вопросов и могут использоваться несколько раз для создания различных вариантов теста. Все виды и типы вопросов могут

быть сочетаемы при создании теста. Поддерживается установка количества баллов, получаемых за ответ на вопрос, определение схем для оценки и предел погрешности в полученных ответах (например, описки).

Наибольшей отличительной особенностью ILIAS является его направление развития, разработчики стараются наделить систему всеми необходимым средствами, для создания среды электронного обучения сохраняя гибкость системы.

К сожалению, ILIAS не отличается простотой в использовании, большое количество возможностей являются как положительными, так и отрицательными качествами системы, хоть и старания разработчиков направлены на унификацию интерфейсов подсистем, на практике интерфейс системы получился достаточно нагромождённым что в конечном итоге сказывается на простоте использования.

Moodle

Moodle — система управления курсами (электронное обучение), также известная как система управления обучением или виртуальная обучающая среда. Является аббревиатурой от англ. Modular Object-Oriented Dynamic Learning Environment (модульная объектно-ориентированная динамическая обучающая среда). Представляет собой свободное (распространяющееся по лицензии GNU GPL) веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения.

Создание системы началось еще в конце 90х годов. Ее создатель, австралиец Мартин Дугиамас специалист в области компьютерных наук и образования, защитивший докторскую диссертацию по проблематике использования свободного программного обеспечения в интернет-обучении.

Первая версия Moodle была выпущена в 2002 году. Версия Moodle 2.0 была представлена в конце 2010 года, и, помимо множества нововведений, привела уже несколько устаревшее ядро системы к современным стандартам интернета. Релиз версии Moodle 3.0 состоялся 16 ноября 2015 года.

На сегодняшний день система дистанционного обучения Moodle является самой распространенной системой дистанционного обучения с самым большим количеством пользователей и разработчиков. Интерфейс системы дистанционного обучения Moodle переведен на 82 языка и используется почти в 50 тысячах организаций из более чем 200 стран мира.

Учебный материал в системе структурирован в курсах. Курсы могут быть реализованы в различных форматах представления содержимого. Содержимое курсов представляют ресурсы (веб-страницы, файлы), элементы деятельности — activities (задания, тесты, форумы, анкеты и т. д.) и блоки (дополнительная навигация, информация или функционал на странице курса). Обучающий может планировать онлайн-обучение, гибко настраивать доступ к учебным элементам, делать массовую и выборочную рассылку, анализировать результаты деятельности обучающихся.

Возможна реализация обучения по группам в рамках одного курса, а также оперирование «глобальными группами» в рамках всего сайта или категории курсов.

В системе Moodle имеется мощная система разработки тестов и анализа результатов тестирования. Возможно воспроизведение SCORM 1.2. Имеется встроенная система обмена сообщениями (не мгновенными). Каждый пользователь имеет домашнюю страницу (персональное пространство).

Высокую степень погружения в интерактивное и проектное обучение обеспечивают такие элементы совместной работы пользователей как: глоссарий (составление базы данных определений-значений), база данных, вики (совместное создание наборов веб-страниц), семинар (работа по оценке сокурсников).

Для расширения функционала системы разработано множество плагинов, которые позволяют, в том числе, реализовать интеграцию с системами веб-конференций (Openmeeting, Big Blue Button), файловыми хранилищами (Dropbox, Google Drive и т.п.), системами анти плагиата и электронного портфолио. Реализована система присвоения наград (Badges), интегрированная с

Open Badges. Базовая тема оформления адаптирована для работы с мобильных устройств. Также имеется мобильное приложение для Android, iOS и Windows.

Управление тестовыми вопросами в Moodle осуществляется через «Банк вопросов». Тесты могут решать задачи входного контроля, текущего, итогового контроля или это могут быть тесты-тренажёры. При этом, тестовые вопросы могут быть общими для некоторых тестов, а также выбираться случайным образом из некоторого набора вопросов — оба эти условия могут быть реализованы благодаря «Банку вопросов». Кроме того, используя «Банк вопросов» легко организовать совместную работу над созданием тестовых вопросов и быстро найти нужный вопрос для теста.

Базовая сборка Moodle включает возможность создавать следующие типы тестовых вопросов:

- выбор одного/нескольких из нескольких;
- истина / ложь;
- краткий открытый ответ (или «вопрос с пропуском»);
- числовой ответ;
- вычисляемый (по формуле) ответ;
- установление соответствия;
- эссе (проверяется вручную);
- вложенные ответы (комбинированный);
- выбор пропущенных слов в тексте;
- перетащите в текст (Рисунок 1.4);
- перетащите маркеры.

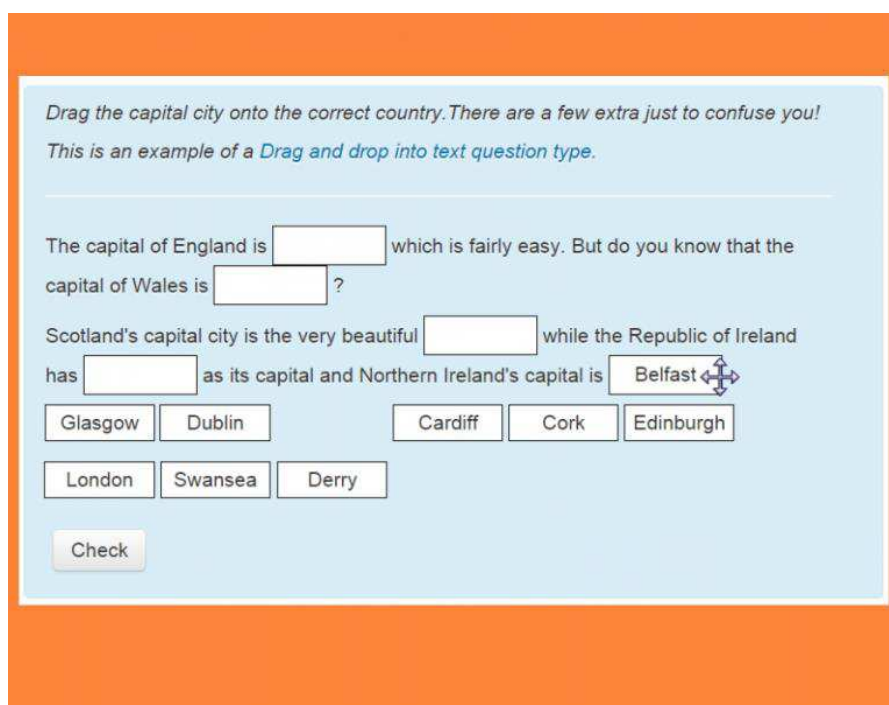


Рисунок 1.4 — Тип вопроса "перетащите в текст"

Moodle используется для организации электронного дистанционного обучения многими университетами мира. Moodle обладает рядом преимуществ, начиная со стоимости дистанционного курса и поддержки русского интерфейса, и заканчивая широкими функциональными возможностями. Однако в системе Moodle наряду со значительными преимуществами существуют и некоторые недостатки. Так у пользователей часто возникают проблемы с получением качественной технической поддержки, следовательно, вузы сталкиваются с необходимостью иметь в штате высококвалифицированного специалиста по Moodle для развертывания и поддержания системы. Сумма затрат учебного заведения на оплату труда такого специалиста в год может превысить стоимость внедрения коммерческой системы дистанционного образования.

Источником финансирования проекта Moodle является сеть партнеров, которые оказывают услуги по установке, технической поддержке, хостингу и т.д.

eFront

eFront — представляет собой платформу для электронного обучения, сочетающая в себе функции систем управления обучением и систем управления и создания учебных материалов.

Разработка системы началась в 2001 году, как исследовательский проект спонсируемый правительством Греции. Стабильная версия системы была опубликована в 2005 году с поддержкой стандарта SCORM и использование технологии AJAX. В течении последующих двух лет eFront была переписана с нуля, были внесены существенные изменения в основной структуре системы, и в сентябре 2007 года система была выпущена под лицензией с открытым исходным кодом. Последняя версия eFrontPro была представлена в 2014 года <https://www.efrontlearning.com/press>, по сути представляя собой абсолютно новый продукт переписанный с нуля с использованием современных веб-технологий и парадигм в веб дизайне. В настоящий момент разработка системы ведется, последняя стабильная версия eFrontPro с обновлением 4.5 от 8 сентября 2016 года.

eFrontPro включает в себя все необходимые инструменты для создания, как дистанционных обучающих курсов, так и организовать смешанный вариант образовательного процесса.

Система eFrontPro позволяет решать задачи организации учебного процесса в учебных заведениях разного формата, но все же получила большее признание среди коммерческих организациях различного масштаба как среда для организации обучения, аттестации и отбора сотрудников [30].

На рисунке 1.5 представлена панель администратора системы eFront, на которой отображаются доступные к настройке параметры курса, а также активность пользователей.

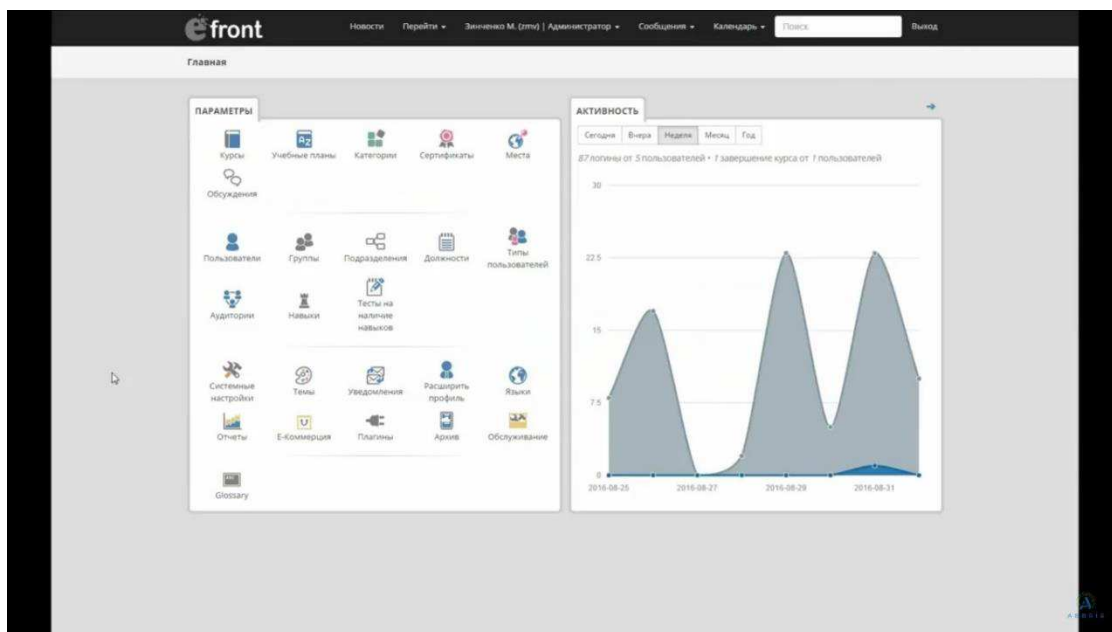


Рисунок 1.5 — Панель управления администратора

В системе четко выделены модули, навигация по которым представлена иконками на главной странице, и быстрым доступом к отдельным модулям в виде вкладок вверху страницы. Учебный материал представлен в уроках, которые могут быть объединены в курсы, а те, в свою очередь, могут стать составляющей учебного плана.

Учебный материал в уроке может быть представлен структурированными веб-страницами, файлами, тестами, проектами (заданиями), опросами, SCORM, форумами.

Система тестирования в eFrontPro предоставляет мощный инструмент для оценки навыков и знаний пользователя. При создании вопроса можно указать навыки на проверку которых направлен вопрос, что в последствии при неудовлетворительном порождении теста в целом дает возможность указать пользователю на тот материал который он освоил не достаточно хорошо.

Система предоставлять возможность использования следующих типов вопросов:

- заполнить пропуск: пользователь должен заполнить пропущенные слова во фразе;

- свободный текст или загрузка файлов: где пользователь загружает файл (например, эссе в текстовом формате) ответ проверяется в ручную куратором;
- свободный текст с ключевыми словами: похож на предыдущий тип, но для него поддерживается автоматическая проверка на основе содержания или отсутствия ключевых слов в тексте;
- множественный выбор: пользователь должен выбрать один или несколько ответов из предложенного списка;
- соответствия: где пользователь должен сопоставить связанные элементы (вопросы и ответы, слова на языке с их переводом в другой и т.д.);
- истина/ложь: где обучаемый должен определить, справедливо ли утверждение, или нет;
- перетаскивание: более "интерактивный" вариация на тему типа вопроса на установку соответствия;
- матрица: несколько вопросов представлены на сетке и правильные ответы зависят от горизонтальных и вертикальных заголовков;
- указать область: вид визуального типа вопроса, который просит пользователя, определить конкретную область на изображении;
- частичная корректность: похож на вопросы с несколькими вариантами, но позволяет, при создании, определить различные оценки корректности для каждого из вариантов.

Пользователь в системе имеет собственную домашнюю страницу с планом занятий. Роли пользователей имеют различные наборы прав, поэтому администратор системы не может параллельно выполнять функции обучающего или обучающегося (в отличие от Moodle). Однако имеется возможность прикрепить несколько профилей к одному логину.

Основной отличительной особенностью системы eFront, являлся ее интерфейс. На всех этапах развития системы, разработчики, старались сделать его простым и интуитивно понятным. В последней версии системы eFrontPro при разработке пользовательского интерфейса использовались современные веб-технологии (jQuery, Bootstrap-3, HTML5, CSS3) что позволило разработчикам,

создать современный пользовательский интерфейс великолепно выглядящий и работающий на всех современных устройствах и сохранить простату использования и надежность.

Система eFront пользуется не большой популярностью среди учебных заведений, так как больше ориентирована на коммерческое использование.

ATutor LMS

ATutor — Система обучения и управления контентом с открытым исходным кодом, разработанная Грегом Гай при Центре адаптивных технологий и ресурсов университета Торонто. Система поддерживает IMS/SCORM. Первая версия системы ATutor 1.0 была выпущена в 2002г. ATutor применяется в разных странах, имеет локализацию более чем на пятнадцати языках, насчитывает более сорока дополнительных языковых модулей и в настоящее время находятся в стадии разработки.

Концепция ATutor была представлена в рамках выпускного проекта, разрабатываемого Греггом Гай во время обучения в аспирантуре в 1994г. В рамках этого проекта Грег Гай разрабатывал онлайн курс под названием Learning to Learn (с англ. — учиться усваивать знания). Курс рассказывал студентам об обучении навыкам, индивидуальных различиях и был направлен на развитие самосознания и индивидуальных возможностям, чтобы помочь учащимся лучше адаптироваться к различным аспектам учебной деятельности. В последующем, после двух исследований в 1999 и 2000 годах, направленных на доступность онлайн-курсов людям с ограниченными возможностями, было задано основное направление развития системы, в которой все пользователи, смогли бы в полной мере участвовать в во всех аспектах образовательной деятельности как в роле учащегося, так и в роле преподавателя.

ATutor первая система обучения и управления контентом полностью поддерживаемая спецификации доступности W3WCAG (Web Content Accessibility Guidelines) в соответствие с этими рекомендациям ресурс должен

быть доступным, в том числе и для пользователей с различными нарушениями здоровья.

Основная особенность системы — это поддержка большого количества стандартов доступности и функциональной совместимости, такие как W3C WCAG 2.0, ISO/IEC 24751, OpenSocial 1.0, SCORM 1.2 LMS RTE3 и многие другие, со списком которых можно ознакомиться на официальном сайте ATutor.

Интерфейс системы прост в использовании, к том уже поддержка стандартов доступности позволяет использовать увеличенные шрифты или нейтральную цветовую схему при необходимости. Так же, в системе есть возможность использования собственных схем оформления.

Использование стандартов совместимости в ATutor, позволяет использовать как электронные курсы, разработанные для других систем (при условии, что система для которой разрабатывался курс поддерживает один из используемых стандартов функциональной совместимости) так и создавать собственные курсы используя любой способ предоставления информации.

На официальном сайте программы[24] размещен демонстрационный курс для знакомства с системой (Рисунок 1.6).

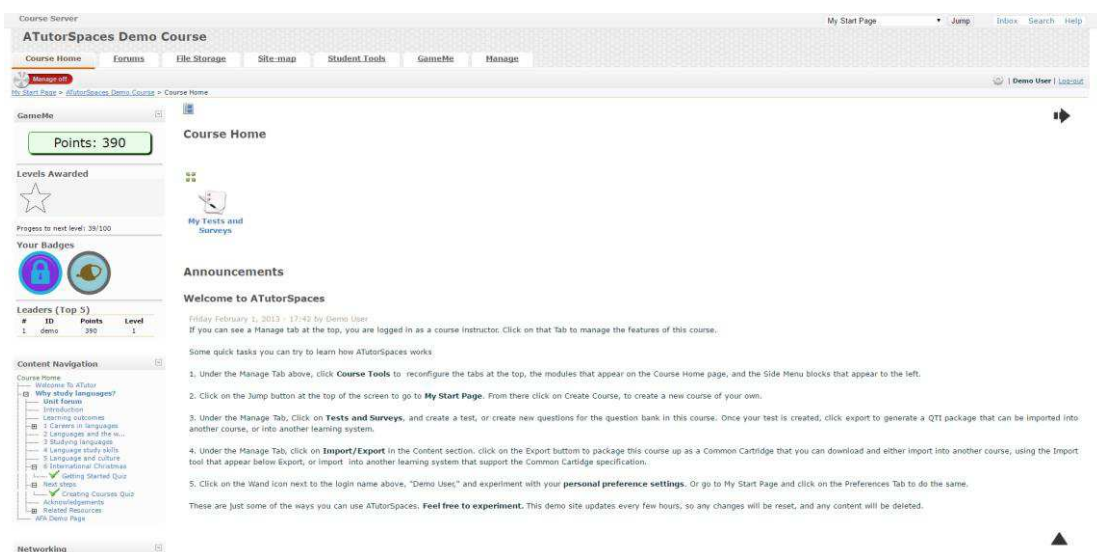


Рисунок 1.6 — Демонстрационный режим ATutor

В ATutore используется игрофикация учебного процесса таких как получения различных достижений, отличительных значков, уровней и использования таблицы лидеров, позволяет дополнительно мотивировать учащихся визуализируя результат обучения.

Оценка знаний и полученных навыков в ATutor проводится при помощи тестирования. Модуль тестирования ATutor поддерживает следующие типы вопросов:

- likert (Ликерт) — тип вопроса требующий от тестируемого указать свой выбор на основе шкалы согласия или несогласия с каким либо утверждением;
- истина/ложь — тип вопроса требующий от тестируемого определить, справедливо ли утверждение, или нет;
- множественный выбор — тип вопроса, ответом на который может являться одно или несколько утверждений;
- открытый вопрос — тип вопроса требующий от тестируемого ввода ответа в текстовое поле;
- сопоставление — тип вопроса на составление пар связей между утверждениями.

Тип вопроса на сопоставление, доступен в двух вариантах графическом и обычном. В графическом варианте между парами утверждений устанавливаются цветные линии, в обычном составление пар происходит путем нумерации.

Система ATutor пройдя путь от выпускного проекта до полноценной системы электронного обучения, являет собой доказательство концепции что электронная система обучения может быть доступной для всех.

Развивающаяся как некоммерческий проект с открытым исходным кодом и разрабатываемая сообществом, тем не менее поддерживает высокий уровень качества системы. К сожалению, как не коммерческий проект разработчики сталкивались с проблемами привлечения средств на проведение маркетинговой компании, для продвижения своего продукта на мировых рынках.

1.2 Проблематика тестирования

Рассмотренные в части 1.1 системы дистанционного образования(СДО) являются яркими примерами развития различных концепций и подходов к процессу организации электронного обучения. Развитие СДО все больше идет по пути социального взаимодействия между учащимися и преподавателями, а также организации доступной среды для всех людей вне зависимости от их возможностей.

В каждой системе так или иначе поддерживается возможность социального взаимодействия и совместной работы пользователей. Активно применяются принципы игрофикации процесса обучения как дополнительный стимул для обучающихся.

Тестирование используется как способ оценки начальных навыков, так и промежуточного и итогового контроля. Большую роль играет тестирование как способ самоконтроля, учащегося без участия преподавателя.

Большинство систем поддерживает стандартный набор вопросов таких как: множественный выбор, установка связей, свободный текст, и т.д.

Развитие современных технологий позволяет развивать и способы тестирования, начинают появляться новые типы вопросов, например, в которых от пользователя требуется найти область на изображении, или заполнить пропуски в тексте перетаскивая слова из предложенных вариантов, то есть появляются элементы интерактивности направленные оценки творческих и практических способностей построения ответа.

Возникает задача построения таких систем в которых задачей прохождения тестирования будет служить не просто выбор правильного ответа из предложенных вариантов, а построение ответа студентом в более сложной форме, например, ответом на задачу может служить построенная схема алгоритма, или собранная электрическая цепь.

Данный вид ответа может быть затронут обычным тестированием (к примеру, вариантами ответа могут быть предложенные варианты построения схем

или алгоритмов), но в таком случае пропадает возможность проверки тех самых теоретических и практических знаний, которые нужны для качественной оценки навыков студента.

В настоящее время коллективом НУЛ САПР СФУ ведутся работы над методом автоматизации синтеза учебных планов на основе массива дидактических единиц[0] и основанной на нем активной системе тестирования. Разрабатываемая система активного тестирования, основанная на предложенном методе формирования тестовых вопросов при построении ДЕ, позволит получать более точные данные об уровне усвоения студентами ДЕ и расширить возможности студента в самоконтроле и работе над ошибками. В частности, при получении оценки после прохождения тестирования, например, при самоконтроле изучения ДЕ, возможно получить данные о том, насколько студент изучил материал, охватываемый ДЕ.

Учитывая формирование укрупненных ДЕ[0] из более мелких ДЕ, то при неверном ответе возможно выделять ту ДЕ, которую студент изучил недостаточно хорошо, что впоследствии позволит провести работу над ее дополнительным изучением, т. е. перейти от обычных тестов к интеллектуальным.

Разрабатываемая коллективом НУЛ САПР СФУ система интеллектуального тестирования позволит увеличить объективность и качество оценки освоения студентами тех или иных дидактических единиц. Новизна разрабатываемой системы активного тестирования и методов ввода и проверки заданий в графическом виде, позволит охватить больший объем проверяемых посредством тестирования знаний, умений и навыков студентов, расширить разнообразие тестов за счет перехода от обычных типов тестов к их интеллектуальному варианту.

1.3 Требования к системе формирования графических образов для системы активного тестирования

Для разрабатываемой коллективом НУЛ САПР СФУ интеллектуальной системы тестирования поставлена задача разработки методов ввода различных типов информации:

- ввод схем;
- ввод графических образов;
- ввод формул;
- ввод текста.

В настоящее время выполняется разработка методов проверки данных таких типов, т. е. разрабатываемая интеллектуальная система тестирования будет расширена возможностями предоставления вопроса как в виде схемы или графического образа, так и получения ответов в таком виде.

В частности, в рамках разработки метода создания тестовых заданий в виде схем и графических образов, и проверки такого типа информации поставленная задача разбита над подсистемы: подсистема формирования графических элементов для системы активного тестирования, подсистема формирования графических образов для системы активного тестирования, подсистема анализа корректности выполнения графических тестовых заданий. Разработка этих систем ведется параллельно.

Разрабатываемая подсистема формирования графических образов, позволит конструировать тестовые задания, ответ на которые ожидается получить в форме схем (например, электронных схем) или графических образов (например, блок-схема алгоритма) и эталон для их проверки.

Следовательно, данная подсистема должна решать следующие задачи:

- используя подготовленную элементную базу, сформировать графический образ схемы, установив параметры элементов и связи между ними;
- используя созданную схему, получать элементную базу, достаточную для решения задания;

– предоставлять информацию о схеме и ее элементной базе в формате, удобном для дальнейшего использования другими подсистемами.

В предметной области, среди графических редакторов и редакторов для создания электронных схем, имеется множество приложений, решающих различные задачи: от редакторов печатных плат до сложных систем, поддерживающих функции моделирования процессов, протекающих в электронных схемах. Например, такие известные приложения как Proteus, NI Multisim, Micro-Cap, LTspice/SwitcherCAD, и другие приложения для симуляции процессов, протекающих в электронных схемах обладают обширным функционалом и огромными элементными базами. Но на использование таких приложений накладывается множество ограничений от стоимости лицензии на использование до закрытости исходного кода. Кроме того, в виду специфики систем тестирования, которые в основном представляют собой веб-приложения или веб-сервис, сложность разбора данных предоставляемых об объекте задания или элементов из которых состоит объект задание является критичным. То есть формат в котором предоставляется данные в объекте задания должен быть ориентированным на веб-технологии.

Некоторые приложения позволяют сохранять схемы в формате SVG или XML. Эти форматы достаточно распространены в среде веб-приложений. В основном такой функционал распространён в среде графических редакторов, и редакторов диаграмм и блок-схем. В некоторых веб-приложениях редакторах диаграмм и схем, таких как: Draw.io[19], Lucidchart[20] — существуют библиотеки компонентов для разработки электронных схем. Но зачастую данные сохраненные в таком формате несут себе либо неполные сведения о элементах, либо содержат излишние служебные данные. Например, на рисунке 1.7 представлена простейшая схема делителя напряжения, выполненная в веб-приложении Draw.io.

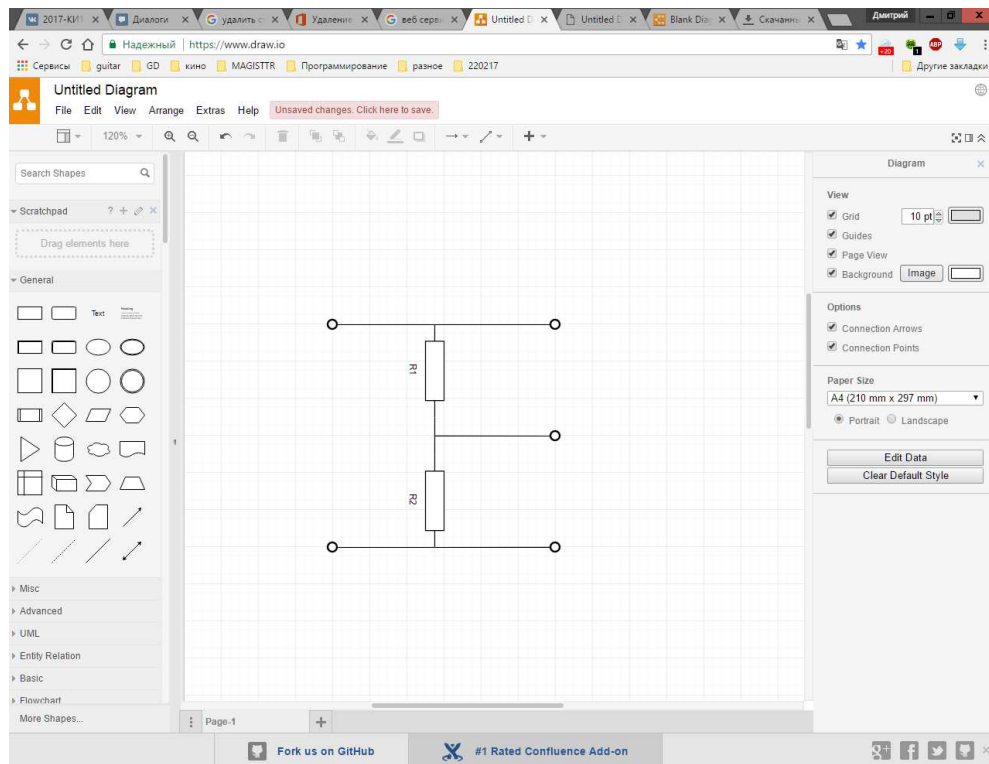


Рисунок 1.7 — Схема делителя напряжения в веб-приложении Draw.io

Веб-приложение Draw.io позволяет экспортировать проект в формат SVG[44]. SVG — это язык разметки масштабируемой векторной графики, входящий в подмножество расширяемого языка разметки XML[46]. Но SVG-файл в который сохраняется данные, несет информацию только о графическом составляющем схемы:

```
<g transform="translate(0.5,0.5)">
  <path d="M 241 6.17 L 11 6.17" fill="none"
stroke="#000000" stroke-miterlimit="10" pointer-events="none"/>
  <rect x="84" y="46" width="64" height="20" fill="#ffffff"
stroke="#000000" transform="rotate(90,116,56)" pointer-
events="none"/>
  <path d="M 66 56 L 84 56 M 148 56 L 166 56" fill="none"
stroke="#000000" stroke-miterlimit="10" trans-
form="rotate(90,116,56)" pointer-events="none"/>
<g transform="translate(91.5,55.5) rotate(90,6.5,0)">
  <switch>
    <foreignObject style="overflow:visible;" pointer-
events="all" width="13" height="10" requiredFea-
tures="http://www.w3.org/TR/SVG11/feature#Extensibility">
      <div xmlns="http://www.w3.org/1999/xhtml"
style="display: inline-block; font-size: 10px; font-family: Hel-
vetica; color: rgb(0, 0, 0); line-height: 1.2; vertical-align:
top; white-space: nowrap; text-align: center;">
```

```

        <div xmlns="http://www.w3.org/1999/xhtml"
style="display:inline-block;text-align:inherit;text-
decoration:inherit;background-color:#FFFFFF;">R1</div>
    </div>
    </foreignObject>
    <text x="7" y="10" fill="#000000" text-
anchor="middle" font-size="10px" font-family="Helvetica">R1</text>
</switch>
</g>
<path d="M 116 146 L 116 106.17" fill="none" stroke="#000000"
stroke-miterlimit="10" pointer-events="none"/>
<path d="M 116 246.17 L 11 246.17" fill="none"
stroke="#000000" stroke-miterlimit="10" pointer-events="none"/>
<rect x="84" y="186" width="64" height="20" fill="#ffffff"
stroke="#000000" transform="rotate(90,116,196)" pointer-
events="none"/>
<path d="M 66 196 L 84 196 M 148 196 L 166 196" fill="none"
stroke="#000000" stroke-miterlimit="10" trans-
form="rotate(90,116,196)" pointer-events="none"/>

```

В вышеуказанном примере нет никакой информации о том какой элемент использован. Присутствуют только XML-теги, для обозначения графических элементов, составляющих резистор, так как формат SVG разработан для описания векторной и смешанной векторно-растровой графики. Полный листинг файла примера представлен в приложении А.

Как говорилось выше, в рамках разработки методов проверки данных, представленных в виде схем или графических образов, разрабатывается веб-сервис для анализа корректности выполнения графических тестовых заданий. В данном сервисе не учитываются графические образы элементов, так как для работы алгоритма достаточны свойства элементов и информация о связях между ними.

Анализ существующих программных продуктов, позволяет выделить общую специфику и функционал реализуемого приложения:

- предоставление базового функционала редакторов электронных схем (редактирование элементов на схеме: добавление, удаление, установка или изменение параметров элементов, установка связей между элементами и редактирование этих связей);
- предоставление простого доступа к базовым элементам;

– функционал, решающий поставленные задачи (преобразование графического представление схемы в формат удобный для использования в подсистеме анализа корректности выполнения графических тестовых заданий).

Для реализации системы формирования графических образов для системы активного тестирования был выбран объектно-ориентированный язык программирования Java с использованием технологии JavaFX.

1.4 Требования к формату представления графических объектов для системы анализа корректности выполнения графических тестовых заданий

В разделе 1.2 было упомянуто, что система анализа корректности выполнения графических тестовых заданий должна быть выполнена в виде веб-сервиса. Для проверки тестовых заданий в веб-сервис достаточно отправить следующую структуру данных:

- сведения о типе и параметрах элемента;
- сведения о соединениях между элементами.

Так же для системы тестирования предполагается передавать данные о графическом образе объекта задания.

Единый формат данных должен быть ориентирован на веб-технологии. Среди распространённых форматов передачи данных стоит выделить форматы XML и JSON.

JSON [43; 53] — текстовый формат обмена данными, основанный на JavaScript[45]. Разработанный для более удобного обмена данными между сервером и браузером. JSON ориентирован на использование в разработке на языке JavaScript, формат считается независимым и может использоваться с практически любым языком программирования.

XML[46;54] — расширяемый язык разметки, разрабатывался как язык с простым формальным синтаксисом. XML — хранит данные в текстовом фор-

мате, что позволяет обеспечивать не зависящий, от программного и аппаратного обеспечения, способ хранения транспортировки и обмена данными.

Форматы данных JSON и XML позволяют записывать данные в структурированном виде. В следующем примере предоставлены данные адресной книги в форматах JSON и XML:

```
{
  "firstName": "Владимир",
  "lastName": "Иванов",
  "address": {
    "street": "Солнечная, 1, кв.1",
    "city": "Москва"
  },
  "phone": [
    "8888888888",
    "9999999999"
  ]
}
```

На языке разметки XML подобная структура выглядит следующим образом:

```
<person firstName="Владимир" lastName="Иванов">
  <address street="Солнечная, 1, кв.1" city="Москва"/>
  <phoneNumbers>
    <phone>8888888888</phone>
    <phone>9999999999</phone>
  </phoneNumbers>
</person>
```

Но реализация взаимодействия между системами удобней в формате XML, т.к. он является более мощным средством описания содержания документа и типов данных в нем, за счет использования DTD[32;51] или XSD[34;52] схем, что позволит избежать лишних ошибок при обмене данными между системами.

Для реализации формата передачи данных был выбран язык разметки XML, с использованием XSD-схем для описания внутренней структуры XML-документов.

1.5 Цели и задачи работы

В части 1.2 был определен базовый функционал системы:

– предоставление базового функционала редакторов электронных схем (редактирование элементов на схеме: добавление, удаление, установка или изменение параметров элементов, установка связей между элементами и редактирование этих связей);

– предоставление простого доступа к базовым элементам;

– функционал, решающий поставленные задачи (преобразование графического представление схемы в формат удобный для использования в подсистеме анализа корректности выполнения графических тестовых заданий).

Из приведенного функционала можно составить следующие задачи:

1 Разработать формат представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.

2 Разработать приложение-редактор графических образов.

Для решения поставленных задач следует выполнить следующие этапы проектирования:

1 Проектирование структуры базы данных.

2 Проектирование взаимодействие с веб-сервером.

3 Проектирование пользовательского интерфейса редактора графических образов.

1.6 Средства разработки

1.6.1 Java

Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Дата первого официального выпуска — 23 мая 1995 года. Последняя версия на текущий момент — Java SE 8(выпущена 18 марта 2014 года).

В руководстве по Java[18], разработчики указали цели и достоинство нового языка. В этом документе приведены следующие характерные особенности Java:

- простой;
- объектно-ориентированный;
- распределенный;
- надежный;
- безопасный;
- не зависящий от архитектуры компьютера;
- переносимый;
- интерпретируемый;
- высокопроизводительный;
- многопоточный;
- динамичный;

Также к особенностям платформы Java следует отнести: большой объем кода, созданный за время существования Java, пригодный для повторного использования, независимость от операционной системы, а также среду для выполнения программ обеспечивающую безопасность.

Исходя из преимуществ, объектно-ориентированный язык программирования Java подходит для работы системы формирования графического образа для системы активного тестирования.

1.6.2 JavaFX

JavaFX — платформа на основе Java для создания приложений с насыщенным графическим интерфейсом. Может использоваться как для создания настольных приложений, запускаемых непосредственно из-под операционных систем, так и для интернет-приложений (RIA), работающих в браузерах, и для приложений на мобильных устройствах. JavaFX призвана заменить использовавшуюся ранее библиотеку Swing. Платформа JavaFX конкурирует с Microsoft Silverlight, Adobe Flash и аналогичными системами.

Технология JavaFX обеспечивает создание мощного графического интерфейса пользователя для крупномасштабных приложений, ориентированных на

обработку данных, насыщенных медиа приложений, поставляющих разнообразный медиа контент пользователю, различного рода пользовательских программ с графикой, анимацией и интерактивными элементами. Технологии создания RIA-приложений платформы Java берут свое начало от Java-апплетов, GUI-интерфейсы которых использовали графические системы AWT и Swing для организации взаимодействия с пользователем и отображения ему данных, текста, графики и анимации. Графическая библиотека AWT была самой первой графической Java-системой набора JDK 1.0, дополненной затем библиотекой Java 2D двумерной графики и изображений. Библиотека AWT предоставляет разработчику возможность использования таких основных компонентов GUI-интерфейса, как кнопки, переключатели, списки, метки, окна выбора файла, меню, компоненты визуализации и редактирования текста, функции "drag&drop", возможность обработки событий UI-компонентов, компоновки компонентов в рабочей области, работы с цветом, шрифтом, графикой, рисования и печати.

Библиотека AWT является тяжеловесной, т. к. она содержит собственную (родную, native) библиотеку `java.awt.peer`, через которую взаимодействует с операционной системой компьютера, поэтому отображение интерфейса программы зависит от операционной системы, в которой приложение развернуто.

Ограниченность набора GUI-компонентов библиотеки AWT и ее тяжеловесность послужили причиной создания графической системы Swing, которая основывается на библиотеке AWT и поэтому является уже легковесной. Кроме того, библиотека Swing дополняет библиотеку AWT различными компонентами GUI-интерфейса. Библиотека Swing реализует архитектуру MVC (Model-View-Controller) и потоковую модель Event Dispatch Thread (EDT).

Несмотря на богатые возможности графических систем AWT и Swing, они не удовлетворяют современным требованиям работы с медиа контентом, что и послужило причиной создания платформы JavaFX, которая предоставляет современные GUI-компоненты, богатый набор библиотек графического и медиа

программного API-интерфейса, а также высокопроизводительную среду выполнения приложений.

Исходя из преимуществ технологии JavaFX над библиотеками Swing и AWT, для разработки графического пользовательского интерфейса системы формирования графических образов для системы активного тестирования, выбрана технология JavaFX.

1.6.3 XML

XML — расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому).

XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка.

Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих какие атрибуты и элементы могут входить в состав других элементов.

Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как собственно XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

1.6.4 XSD

XML Schema — язык описания структуры XML-документа. Спецификация XML Schema является рекомендацией W3C.

Как большинство языков описания XML, XML Schema была задумана для определения правил, которым должен подчиняться документ. Но, в отличие от других языков, XML Schema была разработана так, чтобы её можно было использовать в создании программного обеспечения для обработки документов XML.

После проверки документа на соответствие XML Schema читающая программа может создать модель данных документа, которая включает:

- словарь (названия элементов и атрибутов);
- модель содержания (отношения между элементами и атрибутами и их структура);
- типы данных.

Каждый элемент в этой модели ассоциируется с определённым типом данных, позволяя строить в памяти объект, соответствующий структуре XML-документа. Языкам объектно-ориентированного программирования гораздо легче иметь дело с таким объектом, чем с текстовым файлом.

Выводы по разделу 1

Обзор существующих ЭОК и их систем тестирования показал, что разработка интеллектуальных систем тестирования, является текущим направлением развития ЭОК. Тем самым обуславливая актуальность разработки системы активного тестирования.

Развитие современных технологий позволяет развивать и способы тестирования, начинают появляться новые типы вопросов, например, в которых от пользователя требуется найти область на изображении, или заполнить пропуски

в тексте перетаскивая слова из предложенных вариантов, то есть появляются элементы интерактивности направленные оценки творческих и практических способностей построения ответа.

Возникает задача построения таких систем в которых задачей прохождения тестирования будет служить не просто выбор правильного ответа из предложенных вариантов, а построение ответа студентом в более сложной форме, например, ответом на задачу может служить построенная схема алгоритма, или собранная электрическая цепь.

Разработка метода создания тестовых заданий в виде схем и графических образов, и проверки такого типа информации — осложнена отсутствием подходящего программного обеспечения способного решить поставленные задачи. Для решения этих задач, требуется разработать приложение-редактор графических образов, и формата обмена данными между подсистемами. В качестве средств разработки редактора графических образов выбран объектно-ориентированный язык программирования Java, с использованием технологии JavaFX, а для формата обмена данными язык разметки XML с использованием XML Schema.

2 Разработка формата представления графических объектов для системы анализа корректности выполнения графических тестовых заданий.

2.1 Представление данных

Передача и хранение объектов, таких как, тестовая схема или компонент схемы, осуществляется в формате XML[46]. Для решения задачи разработки формата представления графических данных для системы анализа корректности выполнения графических тестовых заданий разработан альбом документов XML Schema. Использование формата XML Schema, позволяет определить правила которым должны подчиняться документы, то есть определяется порядок и содержание XML-тэгов в документе.

Для использования в системе анализа корректности выполнения графических тестовых заданий, графический образ должен быть представлен в формате XML-документа.

Набор XML-схем имеет следующую структуру:

- Правила описания графических примитивов ("graphicsPrimitives.xsd");
- Вспомогательные элементы ("util.xsd");
- Правила описания элементов ("element.xsd");
- Правила описания объекта задания ("modelTaskObject.xsd").



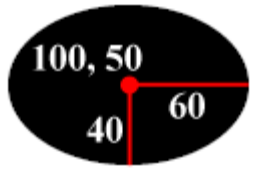

Данные файлы представлены в приложении Б.

2.2 Графические примитивы

Графические примитивы — составные части графического образа элемента. Правила описания графических примитивов находятся в файле "graphicsPrimitives.xsd" представленном в приложении Б.

Графические изображения примитивов и советующие им XML-тэг представлены в таблице 2.1.

Таблица 2.1 — Графические примитивы и их XML-тэги

Изображение	Название	XML-тэг
	Прямоугольник	<pre><gp:rect x = "50" y = "40" height = "30" width = "100" fill = "#000000"/></pre>
	Окружность	<pre><gp:line x1 = "" y1 = "" x2 = "" y2 = ""/></pre>
	Эллипс	
	Линия	

Для более удобного использования графического описания элемента, описания примитивов приведены к формату SVG, и расширены дополнительными элементами:

- ломанная линия (SVG-элемент — "polyline");
- многоугольник (SVG-элемент — "polygone");
- объект "path".

Описание параметров и атрибутов XML тэгов графических элементов.

Линия

Линия, соединяющая точки x_1, y_1 и x_2, y_2 .

Пример использования тэга:

```
<gp:line x1="x1" y1="y1" x2="x2" y2="y2"/>
```

Описание атрибутов:

– x1, y1 — x, y координаты точки начала прямой, принимают значения с плавающей точкой (float);

– x2, y2 — x, y координаты точки конца прямой, принимают значения с плавающей точкой (float).

Описание в "graphicsPrimitives.xsd":

```
<xs:complexType name="line">
  <xs:attribute name="x1" type="xs:float" use="required" />
  <xs:attribute name="y1" type="xs:float" use="required" />
  <xs:attribute name="x2" type="xs:float" use="required" />
  <xs:attribute name="y2" type="xs:float" use="required" />
  <xs:attributeGroup ref="gp:stroke-options" />
  <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
  <xs:attribute name="transform" type="gp:transform"
use="optional"/>
</xs:complexType>
```

Прямоугольник

Прямоугольник задается координатами левого верхнего угла — точка (x, y), и значениями длины(width) и высоты(height).

Пример использования тэга:

```
<gp:rect x="" y="" height="" width="" />
```

Описание атрибутов:

– x, y — координаты левого верхнего угла прямоугольника, принимают значения с плавающей точкой (float);

– height — высота прямоугольника, принимает значение с плавающей точкой (float);

– width — ширина прямоугольника, принимает значение с плавающей точкой (float).

– rx, ry — задают значение скругление углов. Максимальное значение атрибутов равны половине ширины и высоты прямоугольника.

Описание в "graphicsPrimitives.xsd":

```
<xs:complexType name="rect">
  <xs:attribute name="x" type="xs:float" use="required" />
  <xs:attribute name="y" type="xs:float" use="required" />
  <xs:attribute name="width" type="xs:float" use="required"
/>
```

```

        <xs:attribute name="height" type="xs:float"
use="required" />
        <xs:attribute name="rx" type="xs:double" use="optional"
/>
        <xs:attribute name="ry" type="xs:double" use="optional"
/>
        <xs:attributeGroup ref="gp:stroke-options" />
        <xs:attributeGroup ref="gp:fill-options" />
        <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
        <xs:attribute name="transform" type="gp:transform"
use="optional"/>
    </xs:complexType>

```

Окружность

Окружность задается координатами центра (x, y), и радиусом (r).

Пример использования тэга:

```
<gp:circle x="" y="" ""/>
```

Описание атрибутов:

– x,y — координаты левого верхнего угла прямоугольника описывающего эллипс, принимают значения с плавающей точкой (float);

– r — радиус окружности, принимает значение с плавающей точкой (float).

Описание в "graphicsPrimitives.xsd":

```

<xs:complexType name="circle">
    <xs:attribute name="cx" type="xs:float" use="required" />
    <xs:attribute name="cy" type="xs:float" use="required" />
    <xs:attribute name="r" type="xs:float" use="required" />
    <xs:attributeGroup ref="gp:stroke-options" />
    <xs:attributeGroup ref="gp:fill-options" />
    <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
    <xs:attribute name="transform" type="gp:transform"
use="optional"/>
</xs:complexType>

```

Ломанная линия

Ломанная линия используется для рисования фигуры состоящий из нескольких отрезков.

Пример использования тэга:

```
<gp:polyline points="x1,y1 x2,y2, ... xn, yn/>
```


Описание атрибутов:

– `points` — массив координат отрезков составляющих фигуру(float);

Описание в "graphicsPrimitives.xsd":

```
<xs:complexType name="polyline">
  <xs:attribute name="points" type="gp:points"
use="required"/>
  <xs:attributeGroup ref="gp:stroke-options" />
  <xs:attributeGroup ref="gp:fill-options" />
  <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
  <xs:attribute name="transform" type="gp:transform"
use="optional"/>
</xs:complexType>
```

Многоугольник

В отличие от ломанной линии используется для рисования фигуры с замкнутым контуром. Начальная и конечные точки будут соединяться автоматически.

Пример использования тэга:

```
<gp:polygone points="x1,y1 x2,y2, ... xn, yn"/>
```

Описание атрибутов:

– `points` — массив координат отрезков составляющих фигуру(float);

Описание в "graphicsPrimitives.xsd":

```
<xs:complexType name="polygon">
  <xs:attribute name="points" type="gp:points"
use="required"/>
  <xs:attributeGroup ref="gp:stroke-options" />
  <xs:attributeGroup ref="gp:fill-options" />
  <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
  <xs:attribute name="transform" type="gp:transform"
use="optional"/>
</xs:complexType>
```

Объект Path

Объект Path позволяет создавать фигуры, к которым могут быть применены атрибуты оформления — заливка, штриховка.

Фигура, образуемая объектом Path, состоит из отрезков прямых или дуг, которые строятся последовательно, от точки к точке. Для этого используются

следующие команды: установить опорную точку (M), провести линию (L), провести горизонтальную линию (H), провести вертикальную линию (V), завершить построение объекта Path (Z), провести кубическую кривую Безье (C,S), провести квадратичную кривую Безье (Q,T), провести эллиптическую кривую (A).

Пример использования тэга:

```
...
<el:view>
  <gp:g>
    <gp:line x1="0" y1="10" x2="10" y2="10"/>
    <gp:path d="M 10,10 C10, 10 17.5, 0 25, 10
C25, 10 32.5, 0 40, 10
C40, 10 47.5, 0 55, 10
C55, 10 62.5, 0 70, 10"/>
    <gp:line x1="70" y1="10" x2="80" y2="10"/>
  </gp:g>
</el:view>
...
```

Данный пример описывает графическое представление катушки индуктивности, изображенной на рисунке 2.1.



Рисунок 2.1 — Графический образ катушки индуктивности

В примере, представленном выше, объект "path" описывает последовательность кубических кривых Безье, обозначающих на рисунке 2.1 тело катушки индуктивности.

Здесь d – атрибут, определяющий объект "Path". Далее устанавливается опорная точка – M x,y. От этой опорной точки строится последовательность из кубических кривых Безье.

Описание в "graphicsPrimitives.xsd":

```
<xs:complexType name="path">
  <xs:attribute name="d">
```

```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="(M(\s*|\s*-
)?(\d+(\.(\d+))*),(\s*|\s*-)?(\d+(\.(\d+))*))\s((C(\s*|\s*-
)?(\d+(\.(\d+))*),(\s*|\s*-)?(\d+(\.(\d+))*))\s((\s*|\s*-
)?(\d+(\.(\d+))*),(\s*|\s*-)?(\d+(\.(\d+))*))\s{2,2})|(Q(\s*|\s*-
)?(\d+(\.(\d+))*),(\s*|\s*-)?(\d+(\.(\d+))*))\s((\s*|\s*-
)?(\d+(\.(\d+))*),(\s*|\s*-)?(\d+(\.(\d+))*))|(\.){1,})?" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attributeGroup ref="gp:stroke-options" />
    <xs:attributeGroup ref="gp:fill-options" />
    <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
    <xs:attribute name="transform" type="gp:transform"
use="optional"/>
</xs:complexType>

```

Эллипс

Эллипс задается координатами левого верхнего угла (x, y), прямоугольника, описывающего эллипс и значениями длины(width) и высоты(height).

Пример использования тэга:

```
<gp:ellipse x="" y="" rx="" ry="" />
```

Описание атрибутов:

– x,y — координаты левого верхнего угла прямоугольника описывающего эллипс, принимают значения с плавающей точкой (float);

– rx, ry — высота и ширина эллипса, принимают значение с плавающей точкой (float).

Описание в "graphicsPrimitives.xsd":

```

<xs:complexType name="ellipse">
    <xs:attribute name="cx" type="xs:float" use="required" />
    <xs:attribute name="cy" type="xs:float" use="required" />
    <xs:attribute name="rx" type="xs:float" use="required" />
    <xs:attribute name="ry" type="xs:float" use="required" />
    <xs:attributeGroup ref="gp:stroke-options" />
    <xs:attributeGroup ref="gp:fill-options" />
    <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
    <xs:attribute name="transform" type="gp:transform"
use="optional"/>
</xs:complexType>

```

Пример использования графических примитивов для описания графического образа резистора:

```
...
<el:view>
  <gp:g>
    <gp:line x1="20" y1="37" x2="30" y2="37" />
    <gp:line x1="80" y1="37" x2="30" y2="37"/>
    <gp:rect x="30" y="27" width="40" height="20"/>
  </gp:g>
</el:view>
...
```

Графическое отображение резистора описание в примере выше представлено на рисунке 2.2.

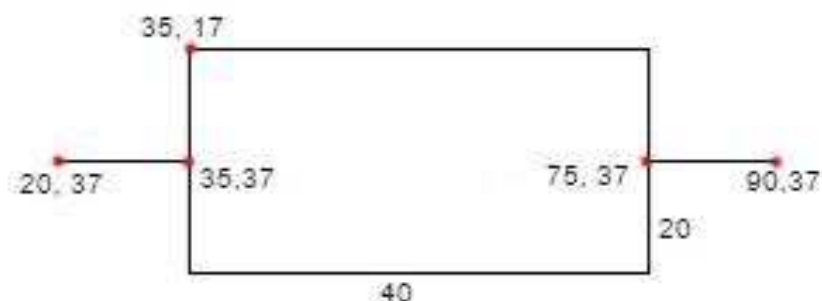


Рисунок 2.2 — Графический образ резистора

2.3 Вспомогательные элементы

Для описания графического образа элемента или объекта задания, используются особые типы объектов — вспомогательные элементы. Список вспомогательных элементов представлен в таблице 2.2.

Таблица 2.2 — Описание вспомогательных элементов

Название	Краткое описание
elementref	Служит для указания подключенного элемента
pin	Описывает точку подключения
pinType	Описывает тип точки подключения
point	Описывает координаты точки
node	Обобщённая точка соединения между тремя и более элементами

Правила описания вспомогательных элементов содержится в файле "util.xsd" представленном в приложение Б.

Ссылка на элемент (elementref)

В контексте XML вспомогательный элемент — "elementref" это комплексный тип данных. Для описание ссылки на элемент, к которому осуществлено подключение используются два атрибута: "elementID" — в котором хранится значение атрибута "id" подключенного элемента, и "pinID" значение которого, является значением атрибута "id" XML-элемента "pin" подключенного элемента. Для атрибутов используется тип данных "IDREF", что означает что допустимыми значениями атрибутов "elementID" и "pinID" будут значения XML-элементов или атрибутов, типом данных которых определён как "ID".

Элемент "elementref" используется для определения типа вложенного элемента "connect" элемента "pin", и также вложенного элемента "element" элемента "node".

Пример использования:

```
...
<el:pin id="r1.p2" type="NOT" visible="true" connect="true">
  <ut:center x="119" y="146"/>
  <ut:connections>
    <ut:connect elementID="r2" pinID="r2.p1"/>
    <ut:connect elementID="pt3" pinID="pt3.p1"/>
  </ut:connections>
</el:pin>
...
```

В данном примере у элемента которому принадлежит пин, есть два соединения, описанных XML-элементами "connect". Первое соединение уставлено с элементом, идентификатор которого равен "r2" и подключение осуществлено через его пин с идентификатором "r2.p1". Второе соединение установлено с элементом, идентификатор которого равен "pt3" через его пин с идентификатором "pt3.p1".

Описание атрибутов:

– elementID — Идентификатор элемента, принимает значения XML-элементов или атрибутов, тип данных которых указан как "ID";

– pinID — Идентификатор пина элемента, принимает значения XML-элементов или атрибутов, тип данных которых указан как "ID".

Описание в "util.xsd":

```
<xs:complexType name="elementref">
  <xs:attribute name="pinID" type="xs:IDREF" use="required"/>
  <xs:attribute name="elementID" type="xs:IDREF" use="required"
/>
</xs:complexType>
```

Точка подключения — пин (pin)

Точка подключения или пин, описывает конкретную точку подключения одного элемента к другому, а также содержит информацию о имеющихся подключениях.

В базовом варианте точка подключения не имеет описания ее графического представления, предполагается что функция отображения точки подключения будет реализована в конечном приложении.

Пример использования:

```
<?xml version="1.0" encoding="UTF-8"?>
<el:element name = "resistor" dataBaseID="2"
xmlns:el="http://www.example.org/element"
xmlns:gp="http://www.example.org/graphicsPrimitives"
xmlns:ut="http://www.example.org/util"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.example.org/element
../xsd/element.xsd ">
  <el:subClass subClassName="RESISTORS" subClassID="7"/>
  <el:type typeName="BASE_RESISTOR" typeID="2"/>
  <el:center x="40" y="20"/>
  <el:view>
    <gp:g>
      <gp:line x1="0" y1="10" x2="20" y2="10" />
      <gp:line x1="60" y1="10" x2="80" y2="10"/>
      <gp:rect x="20" y="0" width="40" height="20"/>
    </gp:g>
  </el:view>
  <el:pins>
    <el:pin id="r1.p1" type="IN" visible="true">
```

```

    <ut:center x="0" y="10"/>
  </el:pin>
  <el:pin id="r1.p2" type="OUT" visible="true">
    <ut:center x="80" y="10"/>
  </el:pin>
</el:pins>
</el:element>

```

В данном примере представлено описание элемента, имеющего две точки подключения с идентификаторами: "r1.p1" и "r1.p2". У элемента имеется два подключения к другим элементам: к элементу с идентификатором "pt1" через его точку подключения с идентификатором "pt1.p1", и к элементу "pt2" через его точку подключения "pt2.p1". Оба подключения осуществлены через точку подключения "r1.p1". Как видно из примера точка подключения "r1.p1" не имеет соединений, и XML-элемент "connections" отсутствует.

Описание атрибутов и элементов:

- атрибут "id" — идентификатор, уникален в пределах документа, принимает значения типа "string";

- атрибут "type" — тип точки подключения, принимает одно из значений, указанных в элементе pinType;

- атрибут "visible" — определяет отображается ли точка подключения по умолчанию, принимает значения типа "boolean";

- элемент "center" — хранит данные о координатах центра объекта, данные передаются через атрибуты x, y, принимающие значения типа "float";

- элемент "connections" — служит для структуризации описаний подключений, служит корневым элементом для элементов "connect".

Элемент "connect" — содержит данные о подключенном элементе и его точке подключения. Данные о элементе передаются через параметры "elementID" и "pinID" типа "IDREF" в которых указываются идентификаторы элемента и пина.

Описание в "util.xsd":

```

<xs:complexType name="pin">
  <xs:sequence>

```

```

        <xs:element name="center" type="ut:point" minOccurs="1"
maxOccurs="1"/>
        <xs:element name="connections" minOccurs="0" max-
Occurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="connect"
type="ut:elementref" minOccurs="1" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="type" type="ut:pinType" use="required"/>
    <xs:attribute name="visible" type="xs:boolean"
use="required"/>
</xs:complexType>

```

Тип точки подключения (pinType)

XML-элемент "pinType" используется для обозначения типа точки подключения. Значение XML-элемента принимает одно из предопределенных значений: "IN" — вход, "OUT" — выход, "NOT" — без определения. XML-элемент "pinType" используется для определения типа атрибута "type" XML-элемента "pin".

Пример использования:

```

<el:pin id="r1.p1" type="IN" visible="true" connect="false">
    <ut:center x="20" y="37"/>
</el:pin>

```

Описание в "util.xsd":

```

<xs:simpleType name="pinType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="NOT"/>
        <xs:enumeration value="IN"/>
        <xs:enumeration value="OUT"/>
    </xs:restriction>
</xs:simpleType>

```

Точка (point)

XML-элемент "point" служит для описания точки с координатами x и y.

Пример использования:


```

<ts:element id = "r1" name = "resistor">
  <el:center x="118" y="96"/>
  <el:view>...</el:view>
  <el:pins>
    <el:pin id="r1.p1" type="NOT" visible="true" connect="true">
      <ut:center x="119" y="46"/>
      <ut:connections>...</ut:connections>
    </el:pin>
    ...
  </el:pins>
</ts:element>

```

В данном примере XML-элемент "point" используется для определения типов XML-элементов "center", которые обозначают координаты середины объектов "элемент" и "точка подключения".

Атрибуты x, y — принимают значения типа "float".

Описание в "util.xsd":

```

<xs:complexType name="point">
  <xs:attribute name="x" type="xs:float" use="required"/>
  <xs:attribute name="y" type="xs:float" use="required"/>
</xs:complexType>

```

Точка соединения (node)

Точка соединения — условное обозначение точки, в которой к одной соединительной линии подключена другая (Рисунок 2.3).

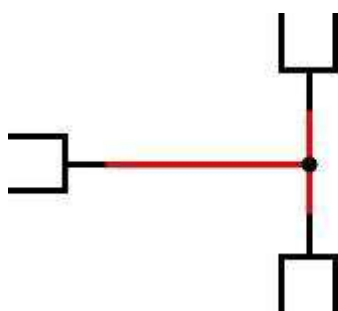


Рисунок 2.3 — Точка соединения

Пример использования:

```

<ts:node id="n1">
  <ut:center x="119" y="29"/>

```

```

<ut:element elementID="r1" pinID="r1.p1"/>
<ut:element elementID="pt1" pinID="pt1.p1"/>
<ut:element elementID="pt2" pinID="pt2.p1"/>
</ts:node>

```

Внутри XML-элемента "node" указаны ссылки на элементы с идентификаторами "r1", "pt1", "pt2". У каждого из этих элементов описано соединение с другими. Соединение с элементом "node" не указывается в списке соединений элемента схемы.

Описание атрибутов и элементов:

- "id" — атрибут, идентификатор узла, принимает строковое значение;
- "center" — XML-элемент типа "point", содержит значения координат x, y середины графического представления узла. Его атрибуты x и y принимают значения типа float;
- "element" — XML-элемент типа "elementref". Описывает ссылку на элемент. Атрибуты "elementID" и "pinID" содержат значения идентификатора элемента и его пина к которому осуществлено подключение, описанное в ссылке.

2.4 Правила описания элементов ("element.xsd")

Как и в случае с графическим образом объекта задания, единственный элемент имеет свой графический образ. Графический образ элемента (Рисунок 2.4а) строится из примитивов (квадратов, линий, кругов и др.) (Рисунок 2.4б).

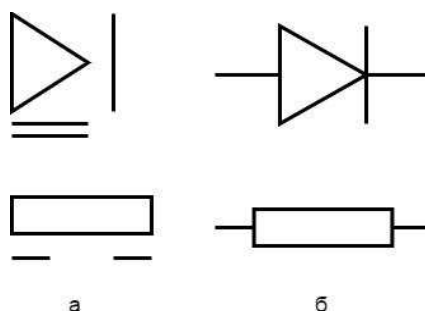


Рисунок 2.4 — Элементы и образующие их графические примитивы

На рисунке 2.4 представлены графические образы элементов, и их составляющие части. Разбиение графического образа элемента на примитивы, а

также разбиение графического образа объекта задания на элементы, позволяет передавать эти объекты в формате XML.

XML-документ содержащий данные о элементе, должен быть построен с использованием XML-схемы (файл "element.xsd" представленный в приложение В). В XML-схеме описана структура документа и все необходимые XML-элементы.

Графический образ элемента, это лишь часть его описания. При представлении элемента в формате XML-документа, описываются не только его графический образ, но и его возможные внутренние параметры такие как: название, идентификатор, координаты центра, список пинов и подключений к ним, параметр, текстовое содержание, подкласс и тип элемента.

Пример использования:

```
<el:subClass subClassName="RESISTORS" subClassID="7"/>
<el:type typeName="BASE_RESISTOR" typeID="2"/>
<el:center x="40" y="20"/>
<el:view>
  <gp:g>
    <gp:line x1="20" y1="37" x2="30" y2="37" />
    <gp:line x1="80" y1="37" x2="90" y2="37"/>
    <gp:rect x="30" y="27" width="40" height="20"/>
  </gp:g>
</el:view>
<el:pins>
  <el:pin id="r1.p1" type="IN" visible="true">
    <ut:center x="20" y="37"/>
  </el:pin>
  <el:pin id="r1.p2" type="OUT" visible="true">
    <ut:center x="90" y="37"/>
  </el:pin>
</el:pins>
</el:element>
```

Приведенный выше пример соответствует графическому отображению элемента, представленному на рисунке 2.5.

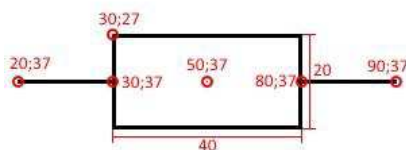


Рисунок 2.5 — Графическое отображение элемента

На рисунке 2.5 обозначены точки каждого примитива из которого строится графическое отображение объекта. Точка 50;37 соответствует середине элемента и указана в примере в атрибутах элемента "center".

Описание элементов:

– element — корневой XML-элемент, внутри которого содержатся XML-элементы, описывающие объект. Содержит атрибут "id" — обозначающий идентификатор объекта, принимающий строковое значение и имеющий тип "ID", и атрибут "name" — обозначающий название описываемого элемента, и принимающий значения типа "string";

– subClass — информация о принадлежности элемента подклассу, в атрибутах "subClassName" и "subClassID" указывается название и идентификатор подкласса элемента, атрибуты принимают значения типа "string";

– type — тип элемента, в атрибутах "typeName" и "typeID" указывается название и идентификатор типа элемента;

– center — описывает координаты центра элемента, атрибуты "x" и "y" принимают значения типа "float";

– view — корневой XML-элемент для XML-элементов описания графического отображения объекта;

– pins — корневой XML-элемент для XML-элементов "pin" описывающих точки подключения элемента;

– param — элемент содержит текстовое значение параметра элемента, принимает значения типа "string";

– text — текстовое содержание элемента, принимает значение типа "string".

2.5 Правила описания объекта задания

Система активного тестирования предполагает использование графического типа заданий, ответ, на который будет представлен схемой, советующей

требованиям задания. Например, для решения задания требуется составить блок-схему алгоритма сортировки вставками из предложенных элементов (2.6).

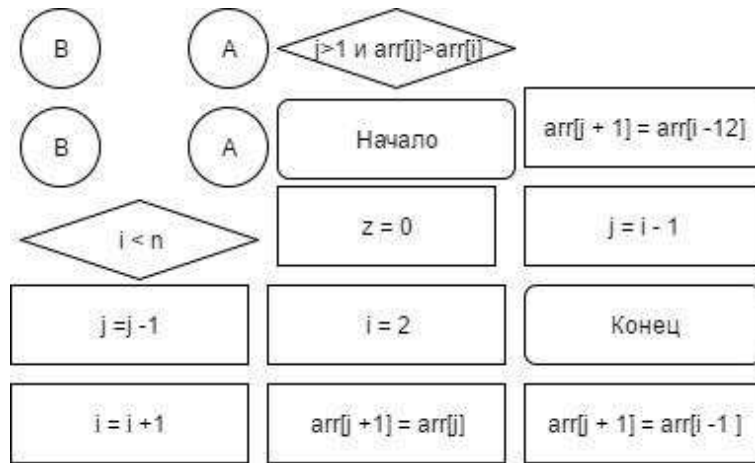


Рисунок 2.6 — Элементная база задания

На рисунке 2.6 представлена элементная база задания. Где каждый элемент представлен своим графическим образом.

Для решения задания нужно установить соединения между элементами, чтобы получить результат представленный на рисунке 2.7.

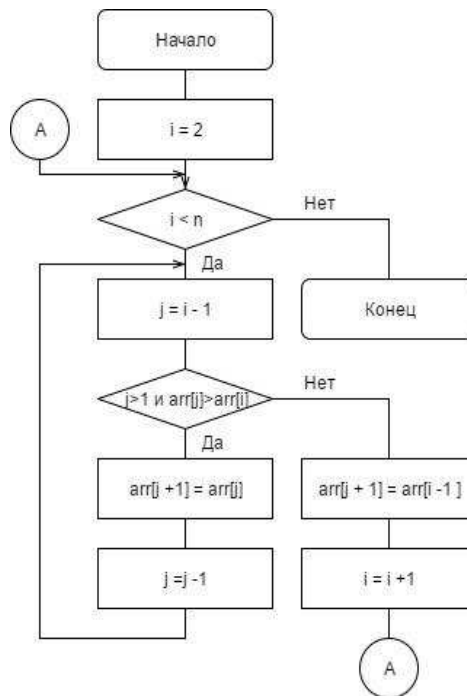


Рисунок 2.7 — Графический образ объекта задания

Как уже говорилось передача данных об объекте задания и элементах, выполняется в формате XML-документов. Эти документы строятся с использованием XML-схем. Для элементов — это файл "element.xsd", для объекта задания — "modelTaskObject.xsd". Листинг этих файлов представлен в приложении В.

XML-схема объекта задания, использует XML-схему "element.xsd", так как при описании в формате XML, элементы из которых строится графический образ задания, записываются в его документ.

Документ с описанием объекта задания, строится из описания каждого его элемента с указаниями их свойств, и параметров объекта задания. Для примера рассмотрим графический образ задания представленный на рисунке 2.8.

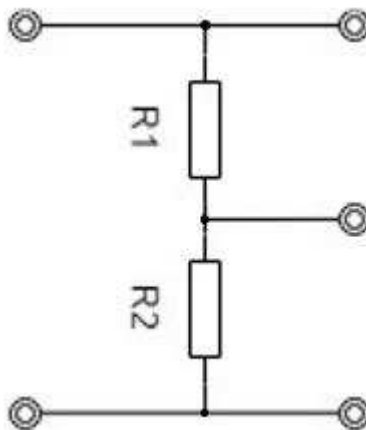


Рисунок 2.8 — Графический образ задания

Для составления XML - документ описывающий графический образ задания, каждый элемент представленный на рисунке 1, описывается в теле документа, в соответствие с XML-схемой элементов (файл "element.xsd") с указанием их внутренних параметров, и подключений. XML-документ описывающий графический образ задания представленный на рисунке 2.8, будет иметь следующее содержание:

```
<?xml version="1.0" encoding="UTF-8"?>
<mto:schema name="example" id="NULL" ...>

  <mto:element id = "r1" name = "resistor">
    <el:subClass subClassName="RESISTORS" subClassID="3"/>
```

```

    <el:type typeName="BASE_RESISTOR" typeID="1"/>
    <el:center x="118" y="96"/>
    <el:view>
        <gp:line x1="119" y1="62" x2="119" y2="46" />
        <gp:line x1="119" y1="130" x2="119" y2="146"/>
        <gp:rect x="99" y="62" rectwidth="40" rec-
theight="67"/>
    </el:view>
    <el:pins>
        <el:pin id="r1.p1" type="NOT" visible="true">
            <ut:center x="119" y="46"/>
            <ut:connections>
                <ut:connect elementID="pt1" pinID="pt1.p1"/>
                <ut:connect elementID="pt2" pinID="pt2.p1"/>
            </ut:connections>
        </el:pin>
        <el:pin id="r1.p2" type="NOT" visible="true">
            <ut:center x="119" y="146"/>
            <ut:connections>
                <ut:connect elementID="r2" pinID="r2.p1"/>
                <ut:connect elementID="pt3" pinID="pt3.p1"/>
            </ut:connections>
        </el:pin>
    </el:pins>
</mto:element>
<mto:element id = "r2" name = "resistor">
    <el:subClass subClassName="RESISTORS" subClassID="3"/>
    <el:type typeName="BASE_RESISTOR" typeID="1"/>
    <el:center x="119" y="214"/>
    <el:view>
        <gp:line x1="119" y1="180" x2="119" y2="164" />
        <gp:line x1="119" y1="248" x2="119" y2="264"/>
        <gp:rect x="99" y="180" rectwidth="40" rec-
theight="67"/>
    </el:view>
    <el:pins>
        <el:pin id="r2.p1" type="NOT" visible="true">
            <ut:center x="119" y="180"/>
            <ut:connections>
                <ut:connect elementID="r1" pinID="r1.p2"/>
                <ut:connect elementID="pt3" pinID="pt3.p1"/>
            </ut:connections>
        </el:pin>
        <el:pin id="r2.p2" type="NOT" visible="true">
            <ut:center x="119" y="264"/>
            <ut:connections>
                <ut:connect elementID="pt4" pinID="pt4.p1"/>
                <ut:connect elementID="pt5" pinID="pt5.p1"/>
            </ut:connections>
        </el:pin>
    </el:pins>
</mto:element>

```

```

<mto:element id = "pt1" name = "port">
  <el:subClass subClassName="CONNECTIONS" subClassID="99"/>
  <el:type typeName="PORT" typeID="99"/>
  <el:center x="58" y="29"/>
  <el:view/>
  <el:pins>
    <el:pin id="pt1.p1" type="NOT" visible="true">
      <ut:center x="58" y="29"/>
      <ut:connections>
        <ut:connect elementID="r1" pinID="r1.p1"/>
        <ut:connect elementID="pt2" pinID="pt2.p1"/>
      </ut:connections>
    </el:pin>
  </el:pins>
</mto:element>
<mto:element id = "pt2" name="port">
  <el:subClass subClassName="CONNECTIONS" subClassID="99"/>
  <el:type typeName="PORT" typeID="99"/>
  <el:center x="180" y="29"/>
  <el:view/>
  <el:pins>
    <el:pin id="pt2.p1" type="NOT" visible="true">
      <ut:center x="180" y="29"/>
      <ut:connections>
        <ut:connect elementID="r1" pinID="r1.p1"/>
        <ut:connect elementID="pt1" pinID="pt1.p1"/>
      </ut:connections>
    </el:pin>
  </el:pins>
</mto:element>
<mto:element id = "pt3" name="port">
  <el:subClass subClassName="CONNECTIONS" subClassID="99"/>
  <el:type typeName="PORT" typeID="99"/>
  <el:center x="180" y="154"/>
  <el:view/>
  <el:pins>
    <el:pin id="pt3.p1" type="NOT" visible="true">
      <ut:center x="180" y="154"/>
      <ut:connections>
        <ut:connect elementID="r1" pinID="r1.p2"/>
        <ut:connect elementID="r1" pinID="r2.p1"/>
      </ut:connections>
    </el:pin>
  </el:pins>
</mto:element>
<mto:element id = "pt4" name="port">
  <el:subClass subClassName="CONNECTIONS" subClassID="99"/>
  <el:type typeName="PORT" typeID="99"/>
  <el:center x="119" y="214"/>
  <el:view/>
  <el:pins>
    <el:pin id="pt4.p1" type="NOT" visible="true">

```



```

        <ut:center x="179" y="280"/>
        <ut:connections>
            <ut:connect elementID="r2" pinID="r2.p2"/>
            <ut:connect elementID="pt5" pinID="pt5.p1"/>
        </ut:connections>
    </el:pin>
</el:pins>
</mto:element>
<mto:element id = "pt5" name="port">
    <el:subClass subClassName="CONNECTIONS" subClassID="99"/>
    <el:type typeName="PORT" typeID="99"/>
    <el:center x="59" y="280"/>
    <el:view>
</el:view>
    <el:pins>
        <el:pin id="pt5.p1" type="NOT" visible="true">
            <ut:center x="59" y="280"/>
            <ut:connections>
                <ut:connect elementID="r2" pinID="r2.p2"/>
                <ut:connect elementID="pt4" pinID="pt4.p1"/>
            </ut:connections>
        </el:pin>
    </el:pins>
</mto:element>
<mto:node id="n1">
    <ut:center x="119" y="29"/>
    <ut:element elementID="r1" pinID="r1.p1"/>
    <ut:element elementID="pt1" pinID="pt1.p1"/>
    <ut:element elementID="pt2" pinID="pt2.p1"/>
</mto:node>
<mto:node id="n2">
    <ut:center x="119" y="154"/>
    <ut:element elementID="r1" pinID="r1.p2"/>
    <ut:element elementID="r2" pinID="r2.p1"/>
    <ut:element elementID="pt3" pinID="pt3.p1"/>
</mto:node>
<mto:node id="n3">
    <ut:center x="119" y="280"/>
    <ut:element elementID="r2" pinID="r2.p2"/>
</mto:node>
</mto:schema>

```

Как видно из представленного примера, все составляющие графического образа задания (Рисунок 2.8), представлены в документе со всеми своими свойствами. Описание элементов (резисторы, и порты) соответствует XML-схеме, используемой для описание элементов (файл "element.xsd"). К внутренним параметрам схемы стоит отнести элементы "node" расширяющие описание подключений между элементами.

Описание XML-элементов:

– "schema" — корневой элемент схемы, в нем декларируются импорты других XML-схем (используются описания, вспомогательных элементов, графических примитивов, и элементов). В атрибутах "name" и "id" указывается название и идентификатор объекта задания, атрибуты принимают значения типа "string";

– "node" — условное обозначение точки, в которой к одной соединительной линии подключена другая.

Содержание XSD-файла "modelTaskObject.xsd" представлено в приложении В.

Выводы по разделу 2

Предложенный подход к разбиению графического образа объекта задания, на его элементы и последующее разбиение элемента на его графические примитивы, позволит дополнить данные о внутренних состояниях элементов и их связей, данными о их графическом представлении, тем самым сохранив графический образ элемента в полном объеме и сделав его не зависимым от программных и аппаратных средств разработки.

Описание графического образа элемента с использованием SVG-элементов, позволит в дальнейшем упростить интеграцию с ЭОК.

3 Разработка системы формирования графических образов для системы активного тестирования

3.1 Разработка структуры базы данных

Процесс обмена данными, между подсистемами формирования графических элементов и формирования графических образов, происходит через, разработанный в рамках подсистемы анализа корректности выполнения графических тестовых заданий, веб-сервис.

Веб-сервис получает данные от приложений и хранит их в базе данных, схема которой изображена на рисунке 3.1.

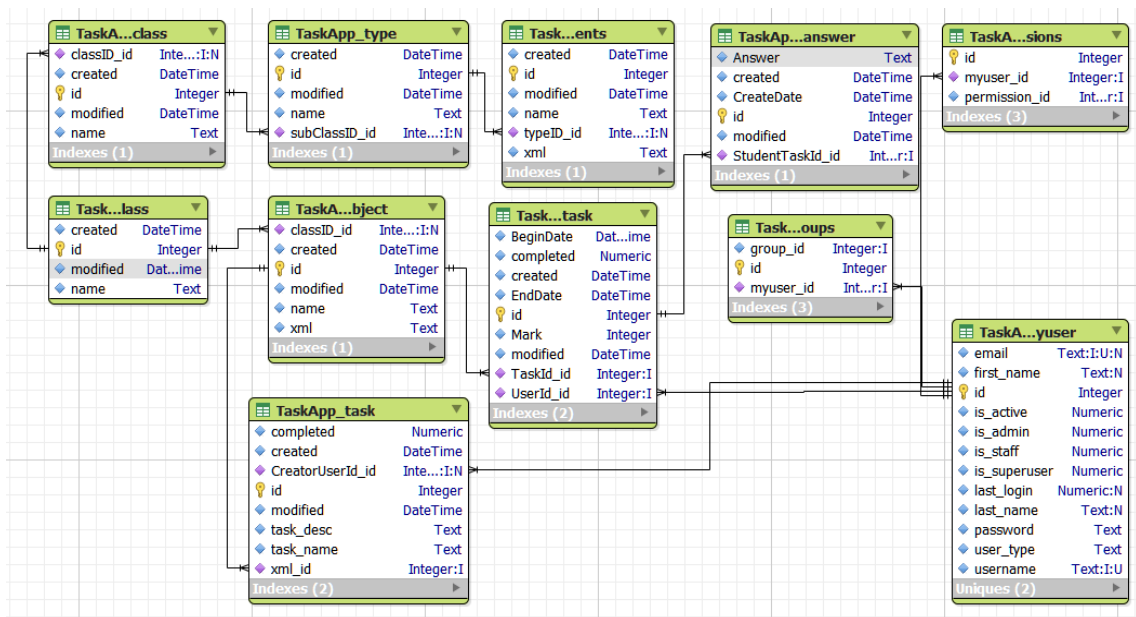


Рисунок 3.1 — Схема базы данных веб-сервиса

Исходя из структуры базы данных серверного приложения, можно получить следующую структуру базы данных системы формирования графических образов изображенную на рисунке 3.2.

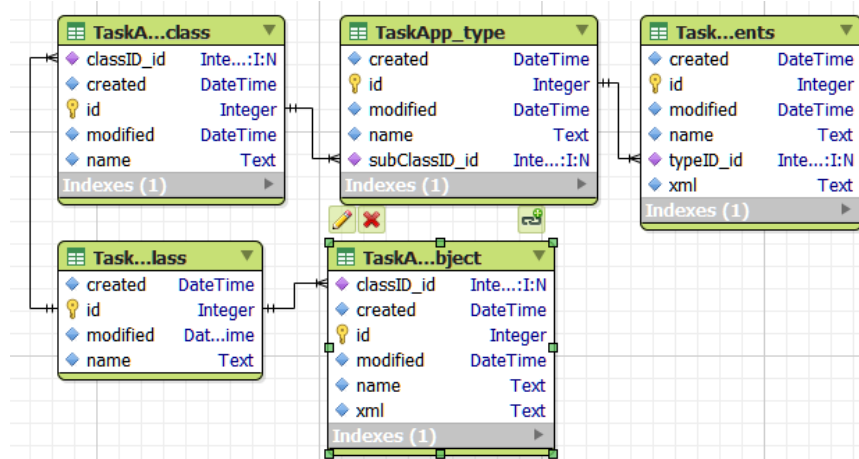


Рисунок 3.2 — Схема локальной базы данных системы формирования графических образов

Для реализации базы данных приложения выбрана — встраиваемая реляционная база данных SQLite3[33]. В отличие от многих других баз данных, таких как — MySQL, MS SQL Server, Oracle Database, Sqlite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором выполняется программа.

3.2 Разработка взаимодействие с веб-сервером

Работа по реализации метода создания тестовых заданий в виде схем и графических образов, и проверки такого типа разделена на подсистемы:

- подсистема формирования графических элементов для системы активного тестирования;
- подсистема формирования графических образов для системы активного тестирования;

– подсистема анализа корректности выполнения графических тестовых заданий.

Разработка этих систем ведется параллельно.

В процессе работы подсистемы формирования графического образа, требуется получать элементную базу, подготовленную в подсистеме формирования графических элементов, из которой и будет разрабатываться графический образ объекта задания.

Реализовать процесс обмена данными между приложениями удобней и безопасней, через веб-сервис реализованный в рамках работы по разработке подсистемы анализа корректности выполнения графических тестовых заданий.

У веб-сервиса разработан программный интерфейс и задокументирован, позволяющий получать все необходимые данные для работы систем.

Взаимодействие между подсистемами реализовано по схеме, представленной на рисунке 3.3.

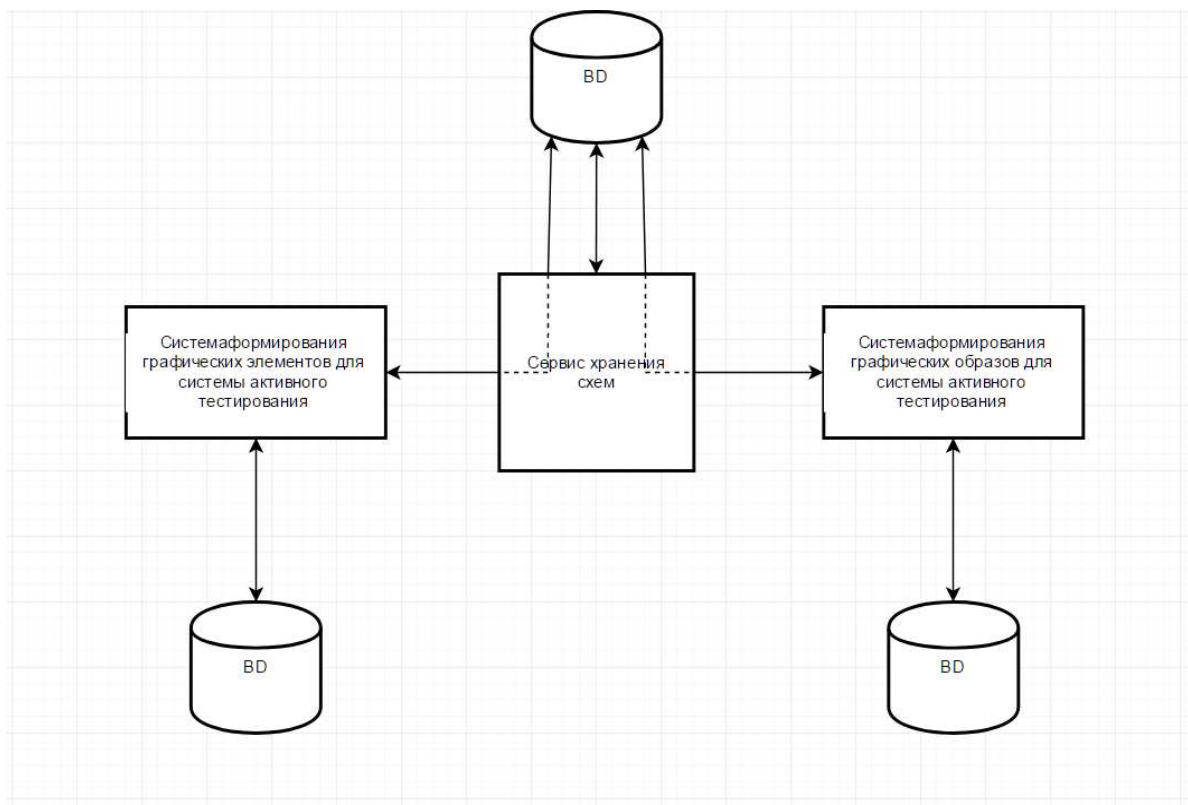


Рисунок 3.3 — Схема обмена информацией между системами

В схеме на рисунке 3.3 процесс синхронизации элементной базы в подсистеме формирования графических образов будет выглядеть следующим образом:

1 Подсистема формирования графических образов обращается к интерфейсу веб-сервису, для получения актуальных изменений.

2 Веб-сервис возвращает подсистеме ответ, в котором указаны текущие изменения в элементной базе.

3 Подсистема формирования графических образов сравнивает полученные данные о изменениях в элементной базе, с данными хранящимися в локальной базе данных подсистемы.

4 В случае нахождения несоответствий, данные о недостающих или измененных элементах получают от сервиса.

Исходный код метода, реализующего синхронизацию с базой данных веб-сервиса представлен в приложение В. На рисунке 3.4 изображено окно программы на стадии пред загрузки элементной базы, с выводом данных о синхронизации с базой данных веб-сервиса.

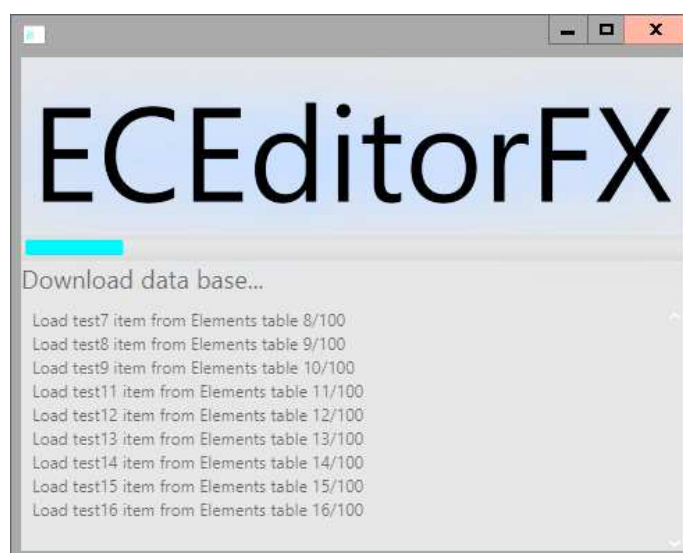


Рисунок 3.4 — Процесс загрузки данных из базы данных веб-сервиса

Загрузка данных о созданном или измененном графическом образе объекта задания происходит по следующему алгоритму:

1 Данные о изменённом или созданном графическом образе объекта задания записываются в локальную базу данных.

2 Изменения, внесенные в локальную базу данных подсистемы отправляются на веб-сервис.

3 Веб-сервис вносит полученные изменения в общую базу данных

3.3 Разработка пользовательского интерфейса редактора графических образов

Основное требование при проектировании графического пользовательского интерфейса это — предоставление простого и удобного доступа к элементной базе системы. Структуризацию и общую структуру графического пользовательского интерфейса можно построить опираясь на известные системы моделирования электронных схем и графических редакторов.

Учитывая поставленные задачи по разработке пользовательского интерфейса и опыт уже существующих приложений, внешний разрабатываемого приложения представлен на рисунке 3.5.

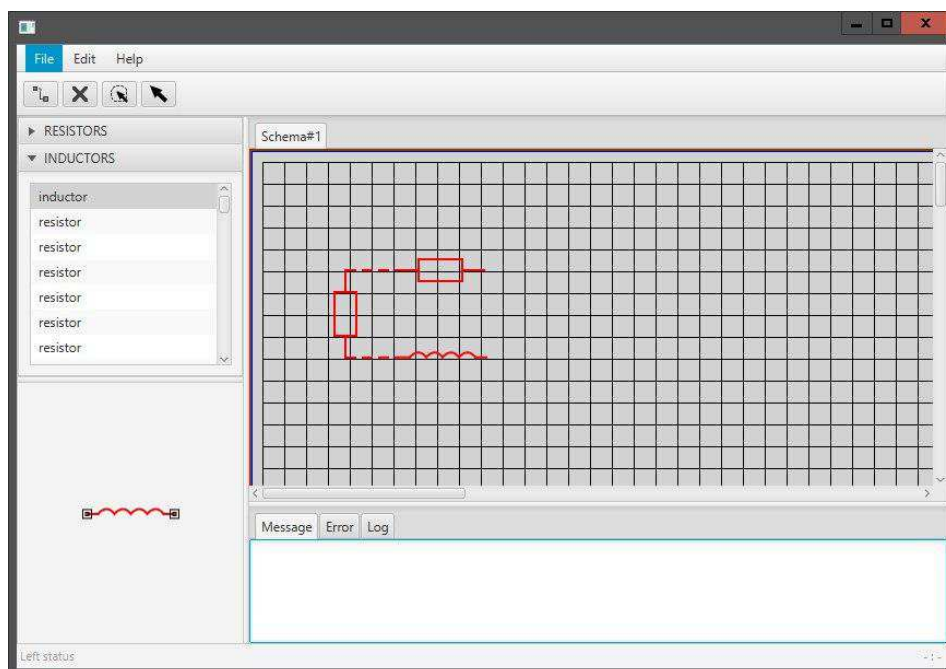


Рисунок 3.5 — Общий вид графического пользовательского интерфейса системы

Графический пользовательский интерфейс системы представленный на рисунке 3.4, состоит из нескольких частей:

- меню программы;
- панель инструментов;
- панель доступа к элементной базе;
- панель отображения выбранного элемента;
- рабочая область редактора;
- панель вывода сообщений пользователю.

Меню программы и панель инструментов представлена на рисунке 3.6.

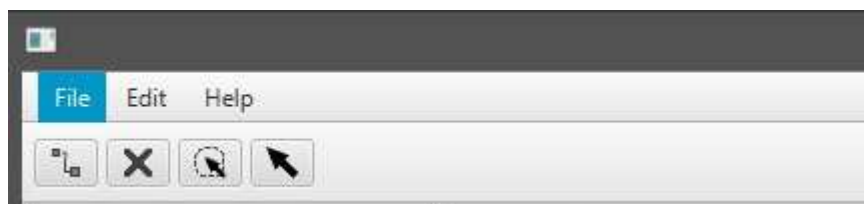


Рисунок 3.6 — Главное меню программы и панель инструментов

Меню программы содержит функции импорта/экспорта графического элемента в предложенном формате XML-файла, а также в графическом формате PNG. Также в главном меню программы размещены функции по синхронизации локальной базы данных с сервером, и записи готовых или измененных объектов в базу данных.

Панель инструментов содержит необходимые инструменты, такие как: перетаскивание объекта, групповое выделение, соединение объектов, удаление, поворот.

На рисунке 3.7 изображена панель доступа к элементной базе системы.

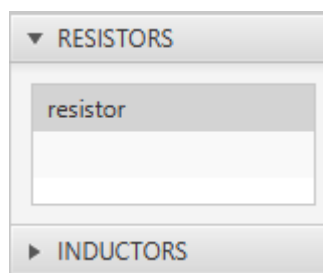


Рисунок 3.7 — Панель доступа к элементной базе системы

Панель доступа к элементной базе системы, представленная на рисунке 3.6, группирует элементы в по их подклассами. Группы получают названия в соответствии с подклассами элементов, представленных в них.

Основным объектом графического пользовательского интерфейса является поле редактора, изображенное на рисунке 3.8.

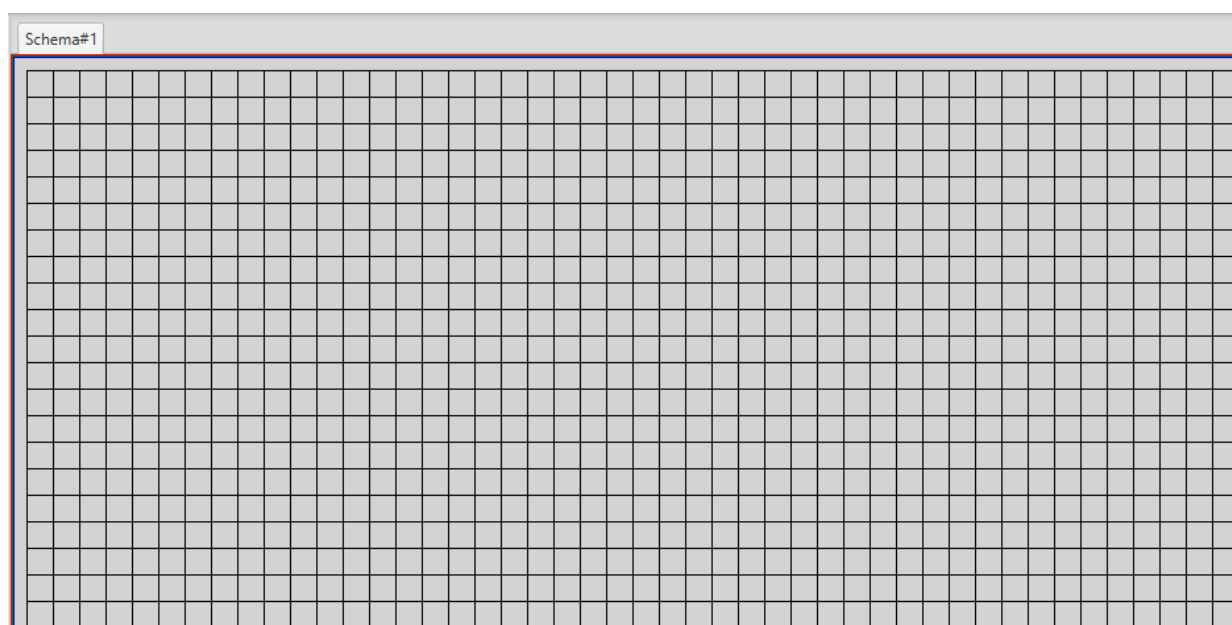


Рисунок 3.8 — Рабочая область редактора

Рабочая область редактора имеет разметку в виде сетки, для более удобного позиционирования элементов. Вверху на текущей рабочей области располагается заголовок с текущим редактируемым графическим образом задания (Рисунок 3.8).

Выводы по разделу 3

Организация хранения данных с использованием базы данных SQLite3, и проектирование схемы базы данных на основе общей схемы базы данных веб-сервиса, обеспечивает простое и надежное хранение данных.

Реализация предложенной схемы взаимодействия с общей базой данных для всех подсистем, позволит избежать возможных ошибок недопустимых дей-

ствий со стороны клиентских приложений, путем реализации доступа к общей базе данных только через интерфейсы веб-сервиса.

Графический пользовательский интерфейс разработанный с учетом требований, поставленных на этапе проектирования, предоставляет простой и удобный доступ как элементной базе системы, так и к основным функциям работы с ней.

ЗАКЛЮЧЕНИЕ

В процессе выполнения работы разработана система формирования графических образов для системы активного тестирования. Разработанная система реализована в виде приложения-редактора с использованием объектно-ориентированного языка программирования Java и технологии JavaFX.

Разработанный с использованием языка разметки XML формат представления данных для системы анализа корректности выполнения тестовых графических заданий, позволяет представить графический образ задания в его текстовом виде, что упрощает его дальнейшую обработку и использование.

СПИСОК СОКРАЩЕНИЙ

ДЕ — дидактическая единица

УП — учебный план

ИКИТ — Институт космических и информационных технологий

НУЛ — Научно-учебная лаборатория

САПР — система автоматизированного проектирования

СФУ — Сибирский федеральный университет

СДО — Система дистанционного образования

ЭОК — Электронные обучающие курсы

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Публикации автора:

1 Кацунова, А. С. Активная система тестирования графического материала: подсистема формирования образов / С. А. Бронов, А. С. Кацунова, И. К. Камиллов, Д. С. Степанов // Информатизация образования и методика электронного обучения : материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы развития» (Красноярск, 27–30 сентября 2016 г.) // под общ. ред. М. В. Носкова. – Красноярск : Сиб. федер. ун-т, 2016. – С. 363–367. – ISBN 978-5-7638-3559-5.

2 Кацунова, А. С. Тесты как элемент контура автоматического управления информационным процессом обучения / С. А. Бронов, А. С. Кацунова, И. К. Камиллов, Б. Р. Хождаев, Д. С. Рогов, Д. С. Степанов, Д. С. Севостьянов, Е. О. Мартыненко // Информатизация образования и методика электронного обучения : материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы развития» (Красноярск, 27–30 сентября 2016 г.) // под общ. ред. М. В. Носкова. – Красноярск : Сиб. федер. ун-т, 2016. – С. 56–60. – ISBN 978-5-7638-3559-5.

Использованные источники:

1 Автоматизированный анализ и синтез учебных планов вуза на основе массива дидактических единиц / С. А. Бронов, Е. А. Степанова, К. В. Калиновский, И. В. Соколов, Н. С. Храброва // Вестник КрасГАУ. 2014. Вып. 3. С. 216—221.

2 Системы дистанционного образования. [Электронный ресурс] / В. Богданов, А. Прохоров // КомпьютерПресс. — Режим доступа: <http://compress.ru/article.aspx?id=11518>.

3 Демаков, В. И. Математическое моделирование процесса формирования учебных планов для высших учебных заведений: автореф. дис. ... канд. техн. наук: 05.13.18 / Демаков Владимир Иванович. — Иркутск, 2006. — 16 с.

4 Демонстрационный сайт системы ILIAS [Электронный ресурс]. — Режим доступа: <https://demo.ilias.de/>

5 Доброзракова Г. А. Использование тестирования при изучении дисциплины «Теория и практика массовой информации» // Поволжский педагогический вестник. — 2015. — № 1(6). — С. 12—15.

6 Документация по системе ATutor [Электронный ресурс] . — Режим доступа: <http://help.atutor.ca/instructor/index.php?>.

7 Документация по системе eFrontPro [Электронный ресурс] . — Режим доступа: <http://www.abbris.ru/blog/docs/>.

8 Документация по системе eFrontPro. Обзор панели управления администратора [Электронный ресурс]. — Режим доступа: <http://www.abbris.ru/blog/панель-управления-администратора/>.

9 Документация по системе eFrontPro [Электронный ресурс] : Тестирование. Тесты и типы вопросов. — Режим доступа: <http://www.abbris.ru/blog/знакомство-с-efrontpro-часть-8-тесты-и-типы-вопросов/>.

10 Документация по системе Moodle. О системе Moodle [Электронный ресурс] . — Режим доступа: https://docs.moodle.org/32/en/About_Moodle#Built_for_learning.2C_globally.

11 Лавлинская, О. Ю. Модели, методы и алгоритмы управления процессом оптимального формирования учебного плана с учётом внешних требований: автореф. дис. ... канд. техн. наук: 05.13.10 / Лавлинская Оксана Юрьевна — Воронеж, 2008. — 17 с.

12 Машнин, Т. С. JavaFX 2.0: разработка RIA-приложений / Т. С. Машнин. — Санкт-Петербург : БХВ-Петербург, 2012. — 320 с.

13 Наумова, С. В. Модели и методы автоматизированного синтеза учебных планов высшего образования: автореф. дис. ... канд. техн. наук: 05.13.18 / Наумова Светлана Викторовна. — Саратов, 2005. — 20 с.

14 Обзор основных возможностей системы Moodle [Электронный ресурс]. — Режим доступа: <http://moodlelearn.ru/mod/page/view.php?id=174>.

15 Обзор платных и бесплатных LMS [Электронный ресурс] . — Режим доступа: <http://it-digger.net/index.php/2012-09-27-13-14-09/95--lms-1>.

16 Обзор релиза Moodle 3.0 Highlights [Электронный ресурс] . — Режим доступа: <http://www.umass.edu/it/support/moodle/moodle-30-highlights>.

17 Обзор системы eFrontPro [Электронный ресурс] . — Режим доступа: <http://blog.uchu.pro/lms-efront/>.

18 Официальное руководство по языку программирования Java [Электронный ресурс] . — Режим доступа: <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>.

19 Официальный сайт приложения Draw.io [Электронный ресурс] . — Режим доступа: <https://www.draw.io/>.

20 Официальный сайт приложения Lucidchart [Электронный ресурс] . — Режим доступа: <https://www.lucidchart.com>.

21 Официальный сайт системы ATutor. Преимущества системы [Электронный ресурс] . — Режим доступа: <http://www.atutor.ca/atutor/features.php>.

22 Официальный сайт системы Moodle. Moodle HQ [Электронный ресурс] . — Режим доступа: <https://moodle.com/hq/>.

23 Официальный сайт системы Moodle. Преимущества системы [Электронный ресурс] . — Режим доступа: https://docs.moodle.org/30/en/New_features.

24 Официальный сайт ЭОК ATutoe [Электронный ресурс] . — Режим доступа: <http://www.atutor.ca/>.

25 Официальный сайт ЭОК ILIAS [Электронный ресурс] . — Режим доступа: <http://www.ilias.de/>.

26 Перова, Ю. П. Технологии тестирования в дистанционном обучении / Ю. П. Перова // Доклады ТУСУР. — 2015. — № 1(6). — С. 12—15. — 2015. — № 1 (35). — С. 138 — 141.

27 Попов, А. В. Тестирование как метод контроля качества знаний студентов / А. В. Попов // Труды СПбГУКИ. — 2013. — № 8. — С. 283 — 284.

28 Румянцева, Л. Н. Преимущества и недостатки системы дистанционного образования Moodle [Электронный ресурс] / Л. Н. Румянцева. — Вятский социально-экономический институт . — Режим доступа: <http://eduros.ru/konf/infor/e9.html>.

29 Сайт демонстрационного режима работы системы ATutor [Электронный ресурс] . — Режим доступа: <https://demo.atutorspaces.com/index.php>.

30 Список партнеров системы eFrontPro [Электронный ресурс] . — Режим доступа: <https://www.efrontlearning.com/customer-stories>

31 Статья о термине Игрофикация [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/Игрофикация>.

32 Статья о DTD [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/DTD>.

33 Статья о SQLite [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/SQLite>.

34 Статья о XML Schema [Электронный ресурс] . — Режим доступа: https://ru.wikipedia.org/wiki/XML_Schema.

35 Статья о подсистеме тестирования в системе Moodle [Электронный ресурс] . — Режим доступа: <http://blog.uchu.pro/testirovanie-sredstvami-moodle/>.

36 Статья о подсистеме тестирования используемой в системе eFrontPro [Электронный ресурс] . — Режим доступа: <https://elearningindustry.com/put-talent-test-acing-tests-in-efrontpro>.

37 Статья о рынке систем дистанционного образования [Электронный ресурс] . — Режим доступа: <http://ra-kurs.spb.ru/2/0/3/1/?id=3>.

38 Статья о системах управления обучением [Электронный ресурс] . — Режим доступа: <http://blog.uchu.pro/cistemy-upravleniya-obucheniem-lms-sdo/>.

39 Статья о системе ATutor [Электронный ресурс] . — Режим доступа: <http://oss-watch.ac.uk/resources/cs-atutor#project-history>.

40 Статья о системе ATutor [Электронный ресурс] . — Режим доступа: <http://wiki.linuxformat.ru/wiki/LXF112:Atutor>.

41 Статья о системе eFrontPro [Электронный ресурс] . — Режим доступа: <http://www.capterra.com/learning-management-system-software/spotlight/18528-/eFrontPro/Epignosis>.

42 Статья о системе Moodle [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/Moodle>.

43 Статья о формате JSON [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/JSON>.

44 Статья о формате SVG [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/SVG>.

45 Статья о языке программирования JavaScript [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/JavaScript>.

46 Статья о языке разметки XML [Электронный ресурс] . — Режим доступа: <https://ru.wikipedia.org/wiki/XML>.

47 Тельнов, Г. В. Тестовый модуль для автоматизированного мониторинга уровня освоения учебной дисциплины «Электроника и схемотехника» // Вестник Адыгейского государственного университета. Сер. 4. Естественно-математические и технические науки. — 2013. — №3 (122). — С. 90 — 97.

48 Темралиева, А. Я. Автоматизированная система формирования учебных планов с процедурой вычисления кредитов: автореф. дис. ... канд. техн. наук: 05.13.10 / А. Я. Темралиева. — Астрахань, 2004. — 28 с.

49 Трофимова, О. К. Автоматизация процесса составления учебных планов вузов: автореф. дис. ... канд. техн. наук: 05.13.10 / Трофимова Ольга Константиновна. — Москва, 1999. — 28 с.

50 Хорстман, Кей С. Java. Библиотека профессионала, том 1. Основы 9-е изд. : пер. с англ. / Кей С. Хорстман, Гари Корнелл. — ООО "И.Д. Вильямс", 2014. — 864 с.

51 Цикл статей о DTD с сайта w3schools [Электронный ресурс] . — Режим доступа: https://www.w3schools.com/xml/xml_dtd_intro.asp.

52 Цикл статей о XML Schema с сайта w3schools [Электронный ресурс] . — Режим доступа: https://www.w3schools.com/xml/schema_intro.asp.

53 Цикл статей о формате JSON с сайта w3schools [Электронный ресурс] . — Режим доступа: https://www.w3schools.com/js/js_json_intro.asp.

54 Цикл статей о языке разметки XML с сайта w3schools [Электронный ресурс] . — Режим доступа: <https://www.w3schools.com/xml/>.

55 Шеметев, А. А. Тесты как эффективный инструмент проверки знаний студентов высшей школы [Электронный ресурс] / А. А. Шеметев // Современные научные исследования и инновации. — 2014. — № 2. — Режим доступа: <http://web.snauka.ru/issues/2014/02/31055>.

56 Ягьяева, Л. Т. Применение информационных технологий в оценке качества образования / Л. Т. Ягьяева, Э. Ю. Замалетдинова // Вестник Казанского технологического университета. — 2011. — № 23. — С. 346 — 349.

57 Ярцев, М. И. Опыт использования центра дистанционного обучения МАДИ для проведения аудиторных занятий на примере дисциплины «Основы теории управления» / М. И. Ярцев, Л. Т. Милов // Молодой ученый. — 2011. — №4. Т.3. — С. 29-32.

58 Кацунова, А. С. Активная система тестирования графического материала: подсистема анализа выполнения тестовых заданий / С. А. Бронов, А. С. Кацунова, И. К. Камилов, Д. С. Рогов // Информатизация образования и методика электронного обучения : материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы развития» (Красноярск, 27–30 сентября 2016 г.) / под общ. ред. М. В. Носкова. – Красноярск : Сиб. федер. ун-т, 2016. – С. 49–52. – ISBN 978-5-7638-3559-5.

59 Кацунова, А. С. Активная система тестирования графического материала: подсистема формирования элементов / С. А. Бронов, А. С. Кацунова, И. К. Камилов, Д. С. Севостьянов // Информатизация образования и методика элек-

тронного обучения : материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы развития» (Красноярск, 27–30 сентября 2016 г.) / под общ. ред. М. В. Носкова. – Красноярск : Сиб. федер. ун-т, 2016. – С. 52–56. – ISBN 978-5-7638-3559-5.

60 Кацунова, А. С. Моделирование информационных процессов при реализации образовательной программы в вузе / С. А. Бронов, А. С. Кацунова, К. В. Калиновский, Т. Г. Боргоякова, Д. С. Тесленко, А. В. Мартынов, Е. Д. Потёмкина, М. Ю. Ушакова // Информатизация образования и методика электронного обучения : материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы развития» (Красноярск, 27–30 сентября 2016 г.) / под общ. ред. М. В. Носкова. – Красноярск : Сиб. федер. ун-т, 2016. – С. 358–363. – ISBN 978-5-7638-3559-5.

61 Кацунова, А. С. Системный анализ информационных процессов в учебном институте вуза / С. А. Бронов, А. С. Кацунова, И. В. Миндалёв, Т. Г. Боргоякова, Е. В. Лозицкая, Д. С. Тесленко, Д. Е. Алфимов, Д. И. Вашлаев, Д. А. Веремеенко // Информатизация образования и методика электронного обучения : материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы развития» (Красноярск, 27–30 сентября 2016 г.) / под общ. ред. М. В. Носкова. – Красноярск : Сиб. федер. ун-т, 2016. – С. 40–44. – ISBN 978-5-7638-3559-5.

62 Кацунова, А. С. Управление информационным процессом разработки учебного плана в вузе / С. А. Бронов, А. С. Кацунова, И. В. Миндалёв, К. В. Калиновский, Е. В. Лозицкая, С. Ю. Пичковская, М. В. Волков, Д. А. Веремеенко // Информатизация образования и методика электронного обучения : материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы

развития» (Красноярск, 27–30 сентября 2016 г.) / под общ. ред. М. В. Носкова. – Красноярск : Сиб. федер. ун-т, 2016. – С. 44–49. – ISBN 978-5-7638-3559-5.

ПРИЛОЖЕНИЕ А

(Обязательное)

Исходный текст примера экспорта графического образа делителя напряжения в формат SVG в веб-приложении Draw.io.

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  width="253px" height="253px"
  version="1.1"
  content="&lt;mxfile userAgent=&quot;Mozilla/5.0 (Windows NT
6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36&quot; version=&quot;6.7.8&quot;
editor=&quot;www.draw.io&quot; &gt;&lt;diagram id=&quot;9c7a19ec-
ac32-9ade-3ab9-10193e439994&quot; name=&quot;Page-
1&quot; &gt; 7VjJbtswEP0aXw1R8npM3KQ9pIDRHJqeCsZiJDaUxqDorV/foTnaQiU
I6kZNUetgk4+cITlv3ojQIFpk+4+ar9PPEAs1CIN4P4g+DMJwMh7jrwUODhhPRw5It
IwdxGrgVv4UBAaEbmQsitZEA6CMXLfBFfES5WJkWxrWGXXvaA6j2qmueCA+4XXHlo19
lbFKHzsJJjX8SMknLldlk7kbu+eox0bdJab1BGD0cHzec8dIXHbRIeQy7BhRdDaKFB
jCule0XQtnQlmFzdtfPjFb71iI3rzIgogpzKM8uYgwfDXPI8e/yeCBhTQLspSZT2GT
YFHttp7gi27W+2PRzbXm704a6cZjvNsfjCk1QvgMi1VIo8/RDGHCGp+MYAQqBNCgnkX
N0ArMltYTQ8VgRZ5AFyQ4bMenKHsyd6Nj4EFbDRK5o1pYzjOhE0a1JxgykvIBN4IJy
iheJGbtve-
OSVfUs2rCcAGcdDNR+RcbLnakNMvzGNoK7SRmKw3/F6oJRTSSMhx6B6MgcwGpswqG4
OYF2nFHVcysVNXeH6hbcQw7AtQoI+uy2RtkVwud0G2xhLQEXxcdW13mO0TWxSGQqE0
tbUcalHIwoAu-
qtZ3a6HsAS4ryTT2cX18OhjVYDgdd14zbLco9i9z7LNHBpUYqViFI+rvaumzgLC0If
vS7hTCT9TfiyrD7lJoiXuyVDuRvnPpjXzpRT1Jj/1C+51SWLYbRGDP5+HfLY8dHLGg
J5JGfn0Mz/Wx3/oYzXqsjxOPcF+1SuG90Opk10ojbtf8mKk7DHKbqFeE07HWGAmOj8
dn+EJqdKvsJApYeaEmCth83EGBzd7AwxMzwwYpJ//PQbY5M0u6e/7TTM78a1CpkuQ
uEhdz6ZP6lkYDFnzaTt07zny8YS0a1Ov4nF2VlKHkiLWo5I6Iv4/X7fn/1Vu1NNNbn
4WQ4cYRlGPYgjOFHTcrd6QAuzWn/bcC6T+fbpd/QI=&lt;/diagram&gt;&lt;/mxfile
&gt;"
  style="background-color: rgb(255, 255, 255);">
  <defs/>
  <g transform="translate(0.5,0.5)">
    <path d="M 241 6.17 L 11 6.17" fill="none"
stroke="#000000" stroke-miterlimit="10" pointer-events="none"/>
    <rect x="84" y="46" width="64" height="20" fill="#ffffff"
stroke="#000000" transform="rotate(90,116,56)" pointer-
events="none"/>
    <path d="M 66 56 L 84 56 M 148 56 L 166 56" fill="none"
stroke="#000000" stroke-miterlimit="10" trans-
form="rotate(90,116,56)" pointer-events="none"/><g trans-
form="translate(91.5,55.5) rotate(90,6.5,0)">
    <switch>
```

```

        <foreignObject style="overflow:visible;" pointer-
events="all" width="13" height="10" requiredFea-
tures="http://www.w3.org/TR/SVG11/feature#Extensibility">
            <div xmlns="http://www.w3.org/1999/xhtml"
style="display: inline-block; font-size: 10px; font-family: Hel-
vetica; color: rgb(0, 0, 0); line-height: 1.2; vertical-align:
top; white-space: nowrap; text-align: center;">
                <div xmlns="http://www.w3.org/1999/xhtml"
style="display:inline-block;text-align:inherit;text-
decoration:inherit;background-color:#FFFFFF;">R1</div>
            </div>
        </foreignObject>
        <text x="7" y="10" fill="#000000" text-
anchor="middle" font-size="10px" font-family="Helvetica">R1</text>
    </switch>
</g>
    <path d="M 116 146 L 116 106.17" fill="none" stroke="#000000"
stroke-miterlimit="10" pointer-events="none"/>
    <path d="M 116 246.17 L 11 246.17" fill="none"
stroke="#000000" stroke-miterlimit="10" pointer-events="none"/>
    <rect x="84" y="186" width="64" height="20" fill="#ffffff"
stroke="#000000" transform="rotate(90,116,196)" pointer-
events="none"/>
    <path d="M 66 196 L 84 196 M 148 196 L 166 196" fill="none"
stroke="#000000" stroke-miterlimit="10" trans-
form="rotate(90,116,196)" pointer-events="none"/>
    <g transform="translate(91.5,195.5)rotate(90,6.5,0)">
        <switch>
            <foreignObject style="overflow:visible;" pointer-
events="all" width="13" height="10" requiredFea-
tures="http://www.w3.org/TR/SVG11/feature#Extensibility">
                <div xmlns="http://www.w3.org/1999/xhtml"
style="display: inline-block; font-size: 10px; font-family: Hel-
vetica; color: rgb(0, 0, 0); line-height: 1.2; vertical-align:
top; white-space: nowrap; text-align: center;">
                    <div xmlns="http://www.w3.org/1999/xhtml"
style="display:inline-block;text-align:inherit;text-
decoration:inherit;background-color:#FFFFFF;">R2</div>
                </div>
            </foreignObject>
            <text x="7" y="10" fill="#000000" text-
anchor="middle" font-size="10px" font-family="Helvetica">R2</text>
        </switch>
    </g>
    <ellipse cx="6" cy="6" rx="5" ry="5" fill="#ffffff"
stroke="#000000" stroke-width="2" pointer-events="none"/>
    <ellipse cx="246" cy="6" rx="5" ry="5" fill="#ffffff"
stroke="#000000" stroke-width="2" pointer-events="none"/>
    <path d="M 241 126.17 L 116 126.11" fill="none"
stroke="#000000" stroke-miterlimit="10" pointer-events="none"/>
    <ellipse cx="246" cy="126" rx="5" ry="5" fill="#ffffff"
stroke="#000000" stroke-width="2" pointer-events="none"/>

```

```
    <path d="M 241 246 L 116 246.17" fill="none"
stroke="#000000" stroke-miterlimit="10" pointer-events="none"/>
    <ellipse cx="246" cy="246" rx="5" ry="5" fill="#ffffff"
stroke="#000000" stroke-width="2" pointer-events="none"/>
    <ellipse cx="6" cy="246" rx="5" ry="5" fill="#ffffff"
stroke="#000000" stroke-width="2" pointer-events="none"/>
  </g>
</svg>
```

ПРИЛОЖЕНИЕ Б

(Обязательное)

Листинг набора XSD-схем для формата представления графических объектов для системы анализа корректности выполнения графических тестовых заданий

Файл "elements.xsd", описывающий содержания файлов графического образа элемента:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/element"
  xmlns:el="http://www.example.org/element"
  elementFormDefault="qualified"
  xmlns:gp="http://www.example.org/graphicsPrimitives"
  xmlns:ut="http://www.example.org/util">

  <xs:import schemaLocation="graphicsPrimitives.xsd"
    namespace="http://www.example.org/graphicsPrimitives"/>
  <xs:import schemaLocation="util.xsd"
    namespace="http://www.example.org/util"/>

  <xs:element name="element" type="el:element"/>

  <xs:complexType name="element">
    <xs:sequence>
      <xs:element name="subClass" type="el:subClass" minOccurs="1" maxOccurs="1"/>
      <xs:element name="type" type="el:type" minOccurs="1" maxOccurs="1"/>
      <xs:element name="center" type="ut:point" minOccurs="1" maxOccurs="1" />
      <xs:element name="view" type="gp:graphicPrimitive" minOccurs="1" maxOccurs="1" />
      <xs:element name="pins" type="el:pins" minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="text" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="param" type="xs:string" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```



```

        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="dataBaseID" type="xs:integer"
use="optional"/>
    </xs:complexType>

    <xs:complexType name="type">
        <xs:attribute name="typeID" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength val-
ue="1"></xs:minLength>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="typeName" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength val-
ue="1"></xs:minLength>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="subClass">
        <xs:attribute name="subClassID"
use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength val-
ue="1"></xs:minLength>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="subClassName"
use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength val-
ue="1"></xs:minLength>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="pins">
        <xs:sequence>

            <xs:element name="pin" type="ut:pin" minOccurs="1"
maxOccurs="unbounded" >
                </xs:element>
        </xs:sequence>

```

```
</xs:complexType>

</xs:schema>
```

Файл "graphicsPrimitives.xsd", описывающий основные примитивы в формате SVG:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" target-
Namespace="http://www.example.org/graphicsPrimitives"
xmlns:gp="http://www.example.org/graphicsPrimitives" element-
FormDefault="qualified">

  <xs:complexType name="graphicPrimetive">
    <xs:sequence>
      <xs:element name="g" type="gp:g" minOccurs="0" max-
Occurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="g">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="line" type="gp:line"/>
      <xs:element name="rect" type="gp:rect"/>
      <xs:element name="circle" type="gp:circle"/>
      <xs:element name="ellipse" type="gp:ellipse"/>
      <xs:element name="polyline" type="gp:polyline" />
      <xs:element name="path" type="gp:path"/>
    </xs:choice>
    <xs:attributeGroup ref="gp:fill-options"/>
    <xs:attributeGroup ref="gp:stroke-options"/>
    <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
    <xs:attribute name="transform" type="gp:transform"
use="optional"/>
  </xs:complexType>

  <xs:complexType name="rect">
    <xs:attribute name="x" type="xs:float" use="required" />
    <xs:attribute name="y" type="xs:float" use="required" />
    <xs:attribute name="width" type="xs:float" use="required"
/>
    <xs:attribute name="height" type="xs:float"
use="required" />
    <xs:attribute name="rx" type="xs:double" use="optional"
/>
    <xs:attribute name="ry" type="xs:double" use="optional"
/>
    <xs:attributeGroup ref="gp:stroke-options" />
    <xs:attributeGroup ref="gp:fill-options" />
    <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
```

```

        <xs:attribute name="transform" type="gp:transform"
use="optional"/>
    </xs:complexType>

    <xs:complexType name="circle">
        <xs:attribute name="cx" type="xs:float" use="required" />
        <xs:attribute name="cy" type="xs:float" use="required" />
        <xs:attribute name="r" type="xs:float" use="required" />
        <xs:attributeGroup ref="gp:stroke-options" />
        <xs:attributeGroup ref="gp:fill-options" />
        <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
        <xs:attribute name="transform" type="gp:transform"
use="optional"/>
    </xs:complexType>

    <xs:complexType name="ellipse">
        <xs:attribute name="cx" type="xs:float" use="required" />
        <xs:attribute name="cy" type="xs:float" use="required" />
        <xs:attribute name="rx" type="xs:float" use="required" />
        <xs:attribute name="ry" type="xs:float" use="required" />
        <xs:attributeGroup ref="gp:stroke-options" />
        <xs:attributeGroup ref="gp:fill-options" />
        <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
        <xs:attribute name="transform" type="gp:transform"
use="optional"/>
    </xs:complexType>

    <xs:complexType name="line">
        <xs:attribute name="x1" type="xs:float" use="required" />
        <xs:attribute name="y1" type="xs:float" use="required" />
        <xs:attribute name="x2" type="xs:float" use="required" />
        <xs:attribute name="y2" type="xs:float" use="required" />
        <xs:attributeGroup ref="gp:stroke-options" />
        <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
        <xs:attribute name="transform" type="gp:transform"
use="optional"/>
    </xs:complexType>

    <xs:complexType name="polyline">
        <xs:attribute name="points" type="gp:points"
use="required"/>
        <xs:attributeGroup ref="gp:stroke-options" />
        <xs:attributeGroup ref="gp:fill-options" />
        <xs:attribute name="opacity" type="gp:opacity"
use="optional"/>
        <xs:attribute name="transform" type="gp:transform"
use="optional"/>
    </xs:complexType>

```



```

                <xs:pattern val-
ue="(\s*\d+){1,}"></xs:pattern>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:attributeGroup>

    <xs:simpleType name="transform">
        <xs:restriction base="xs:string">
            <xs:pattern value="((\s*(translate\(((\s*|\s*-
)?(\d+(\.(\d+))*),(\s*|\s*-
)?(\d+(\.(\d+))*))\))|(\s*((scale|rotate)\((,*\s*((\s*|\s*-
)?(\d+(\.(\d+))*)))*,*\s*\)))+"></xs:pattern>
            </xs:restriction>
        </xs:simpleType>

    <xs:simpleType name="opacity">
        <xs:restriction base="xs:double">
            <xs:pattern
vaue="((0|1)\.\d*)|1|0){1}"></xs:pattern>
            </xs:restriction>
        </xs:simpleType>

    <xs:simpleType name="points">
        <xs:restriction base="xs:string">
            <xs:pattern value="((-*(\d+)(.\d+)*),(-
*(\d+)(.\d+)*)){2,}" />
            </xs:restriction>
        </xs:simpleType>

</xs:schema>

```

Файл "modelTaskObject.xsd", используется при описание графического образа объекта задания:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/modelTaskObject"
  xmlns:mto="http://www.example.org/modelTaskObject"
  elementFormDefault="qualified"
  xmlns:el="http://www.example.org/element"
  xmlns:ut="http://www.example.org/util">
  <xs:import schemaLocation="element.xsd"
  namespace="http://www.example.org/element">
  </xs:import>

  <xs:import schemaLocation="util.xsd"
  namespace="http://www.example.org/util" />
  <xs:element name="schema" type="mto:schema" />

<xs:complexType name="schema">

```

```

        <xs:sequence>
            <xs:sequence>
                <xs:element name="element" type="el:element"
minOccurs="1" maxOccurs="unbounded">
                    </xs:element>
                </xs:sequence>
            <xs:element name="node" type="ut:node" minOccurs="0"
maxOccurs="unbounded">
                </xs:element>
            <xs:element name="rules" type="xs:string" minOc-
curs="0" maxOccurs="1">
                </xs:element>
        </xs:sequence>
        <xs:attribute name="id" type="xs:ID" use="optional"/>
        <xs:attribute name="name" type="xs:string"
use="required"/>
    </xs:complexType>
</xs:schema>

```

Файл "util.xsd", содержащий вспомогательные элементы для описания графического образа объекта задания, и графического образа элемента:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" target-
Namespace="http://www.example.org/util"
xmlns:ut="http://www.example.org/util" element-
FormDefault="qualified">
    <xs:complexType name="point">
        <xs:attribute name="x" type="xs:float" use="required"/>
        <xs:attribute name="y" type="xs:float" use="required"/>
    </xs:complexType>

    <xs:complexType name="pin">
        <xs:sequence>
            <xs:element name="center" type="ut:point" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="connections" minOccurs="0" max-
Occurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="connect"
type="ut:elementref" minOccurs="1" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>

        </xs:sequence>
        <xs:attribute name="id" type="xs:ID" use="required"/>
        <xs:attribute name="type" type="ut:pinType" use="required"/>
        <xs:attribute name="visible" type="xs:boolean"
use="required"/>
    </xs:complexType>

```

```

<xs:complexType name="node">
  <xs:sequence>
    <xs:element name="center" type="ut:point" minOccurs="1"
maxOccurs="1"/>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="element"
type="ut:elementref"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

<xs:simpleType name="pinType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NOT"/>
    <xs:enumeration value="IN"/>
    <xs:enumeration value="OUT"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="elementref">

  <xs:attribute name="pinID" type="xs:IDREF" use="required"/>
  <xs:attribute name="elementID" type="xs:IDREF" use="required"
/>
</xs:complexType>

</xs:schema>

```

ПРИЛОЖЕНИЕ В

(Обязательное)

**Исходный код метода loadDBfromServer() класса DBManager,
реализующего синхронизацию локальной базы данных с базой данных
веб-сервиса.**

```
public static void loadDBfromServer() {
    ApiClient.init();
    System.out.println(ApiClient.getApiUrl());

    for(String apiUrl : ApiClient.getApiUrl()) {
        JSONArray jsonArray =
new JSONArray(ApiClient.sendGet(apiUrl));
        int jsonArrayCount;
        int jsonArraySize;

        if(jsonArray.length() != 0) {
            jsonArrayCount = 0;
            jsonArraySize = jsonArray.length();
            for(Object o : jsonArray){
                jsonArrayCount++;
                JSONObject arr = new JSONObject(o.toString());
                PreparedStatement pstmt;
                String sqlInsert = "";
                Pattern pattern = null;
                Matcher matcher = null;
                try {
                    switch (apiUrl) {
                        case
"http://rodmis.pythonanywhere.com/class_event/":
                            if(printLog) {
                                if(output != null){
                                    output.set("Load item
from Class table "+jsonArrayCount + "/" + jsonArraySize);
                                }
                            }
                            sqlInsert = "INSERT OR IGNORE
INTO Class (id, name, modified, created) VALUES (?, ?, ?, ?)";
                            pstmt =
conn.prepareStatement(sqlInsert);
                            pstmt.setInt(1, (int)
arr.get("id"));
                            pstmt.setString(2,
arr.get("name").toString());
                            pstmt.setString(3,
arr.get("modified").toString());
```



```

                                pstmt.setString(4,
arr.get("created").toString());
                                pstmt.executeUpdate();
                                break;
                                case
"http://rodmis.pythonanywhere.com/subclass_event/":
                                if(printLog) {
                                    if(output!=null){
                                        output.set("Load item
from SubClass table "+jsonArrayCount + "/" + jsonArraySize);
                                    }
                                }
                                sqlInsert = "INSERT OR IGNORE
INTO SubClass (id, name, classID_id, modified, created) VALUES
(?,?,?,?,?)";
                                pstmt =
conn.prepareStatement(sqlInsert);
                                pstmt.setInt(1, (int)
arr.get("id"));
                                pstmt.setString(2,
arr.get("name").toString());
                                pattern = Pat-
tern.compile("(?<=\\/)\\w*(?=\\/$)");
                                matcher = pat-
tern.matcher(arr.get("classID").toString());
                                if(matcher.find()) {
                                    pstmt.setInt(3, Inte-
ger.parseInt(matcher.group()));
                                }
                                pstmt.setString(4,
arr.get("modified").toString());
                                pstmt.setString(5,
arr.get("created").toString());
                                pstmt.executeUpdate();
                                break;
                                case
"http://rodmis.pythonanywhere.com/type_event/":
                                if(printLog) {
                                    if(output!=null){
                                        output.set("Load item
from Type table "+jsonArrayCount + "/" + jsonArraySize);
                                    }
                                }
                                sqlInsert = "INSERT OR IGNORE
INTO Type (id, name, subclassID_id, modified, created) VALUES
(?,?,?,?,?)";
                                pstmt =
conn.prepareStatement(sqlInsert);
                                pstmt.setInt(1, (int)
arr.get("id"));
                                pstmt.setString(2,
arr.get("name").toString());

```

```

        pattern = Pattern.compile("(?<=\\/)\\w*(?=\\/$)");
        matcher = pattern.matcher(arr.get("subClassID").toString());
        if(matcher.find()) {
            pstmt.setInt(3, Integer.parseInt(matcher.group()));
        }
        pstmt.setString(4, arr.get("modified").toString());
        pstmt.setString(5, arr.get("created").toString());
        pstmt.executeUpdate();
        break;
    case
"http://rodmis.pythonanywhere.com/elements_event/":
        if(printLog) {
            if(output!=null){
                output.set("Load
"+arr.get("name").toString()+" item from Elements table
"+jsonArrayCount + "/" + jsonArraySize);
            }
        }
        pattern = Pattern.compile("(?<=\\/)\\w*\\.\\w*$");
        matcher = pattern.matcher(arr.get("xml").toString());
        String fileName = "";
        if(matcher.find()) {
            System.out.println(matcher.group());
            fileName = matcher.group();
        }
        pstmt = getStatment("SELECT *\n"
            +"FROM Elements\n"
            +"WHERE ('Elements'.id' =
?)"");
        pstmt.setInt(1, (int) arr.get("id"));
        ResultSet rs = pstmt.executeQuery();
        if(rs.next()) {
            String dbloc =
rs.getString("modified");
            dbloc = dbloc.replaceAll("[Z]",
            "");
            String dbs =
arr.getString("modified");
            dbs = dbs.replaceAll("[Z]", "");
            try {
                System.out.println(LocalDate.parse(dbs));
                System.out.println(LocalDate.now());
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

if( LocalDateTime.parse(dbs).compareTo(LocalDateTime.parse(dbloc))
>= 0){

File(rs.getString("xml"));

File f = new
if(f.exists()) {
break;
}
}
pattern =
Pattern.compile("(?<=\\/)\\w*(?=\\/$)");
matcher =
pattern.matcher(arr.get("typeID").toString());
matcher.find();
pstmt = getStatment("SELECT
'Type'.'name', 'SubClass'.'name', 'Class'.'name'\n"
+"FROM Type\n"
+"INNER JOIN SubClass ON
'Type'.'subclassID_id' = 'SubClass'.'id'\n"
+"INNER JOIN Class ON 'SubClass'.'classID_id' =
'Class'.'id'\n"
+"WHERE ( 'Type'.'id' = ?
)");
pstmt.setInt(1, Integer.parseInt(matcher.group()));
rs = pstmt.executeQuery();
String file = "re-
sources/components/"+rs.getString(3)+"/"+rs.getString(2)+"/"+rs.ge
tString(1)+"/"+fileName;
String urlStr =
arr.get("xml").toString();
ApiClient.downloadFile(urlStr, file);
insertInElements((int) arr.get("id"),
arr.get("name").toString(),
Integer.parseInt(matcher.group()),
file, arr.get("modified").toString(),
arr.get("created").toString());
break;
default:
break;
}
} catch (SQLException e1) {
e1.printStackTrace();
}
}}}}

```

ПРИЛОЖЕНИЕ Г

**Руководство по интерфейсу веб-сервиса анализа корректности
выполнения графических тестовых заданий.**

Федеральное государственное автономное
образовательное учреждение
высшего образования
"СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ"

Институт космических и информационных технологий


институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ВТ

 А.И.Легалов

"20" июня 20 17 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Подсистема формирования графических образов для системы ак-
тивного тестирования

тема

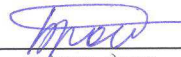
09.04.01 Информатика и вычислительная техника

код и наименование направления

09.04.01.02 Информационное и программное обеспечение САПР

код и наименование магистерской программы

Руководитель


подпись, дата

проф., д-р техн. наук

должность, ученая степень

С.А.Бронов

инициалы, фамилия

Выпускник


подпись, дата

Д.С.Степанов

инициалы, фамилия

Рецензент


подпись, дата

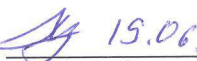
проф., д-р техн. наук

должность, ученая степень

Ю.В.Краснобаев

инициалы, фамилия

Нормоконтролер

 15.06.17
подпись, дата

доц., канд. техн. наук

должность, ученая степень

В.И.Иванов

инициалы, фамилия

Красноярск 2017