**АВТОМАТИЗАЦИЯ GUI ТЕСТИРОВАНИЯ В ПРОЕКТЕ WMP**

Безгузиков А.В., 4 курс. кафедра системного программирования СПбГУ,

artemii.bezguzikov@gmail.com

Брыксин Т.А., к.т.н., доц. кафедры системного программирования СПбГУ,

t.bryksin@spbu.ru

**Аннотация**

В работе рассматриваются основные инструменты тестирования пользовательского интерфейса, в частности, Selenium WebDriver, изучается вопрос их применимости к тестированию веб-редакторов диаграмм, предлагается API (Application Programming Interface) для GUI (Graphical User Interface) тестирования редактора диаграмм проекта WMP[[1]](#footnote-0).

# **Введение**

Во многих проектах помимо функционального и интеграционного тестирования используется и автоматизированное GUI тестирование. Само по себе GUI тестирование – это имитация действий пользователя. Оно предназначено для поиска ошибок, возникающих при взаимодействии с интерфейсом, и проверки выполнения всех основных сценариев работы с приложением.

На кафедре системного программирования СПбГУ разрабатывается проект WMP. Он служит для создания средств визуального программирования, в основе которых лежит построение диаграмм. На данном этапе развития WMP включает в себя редактор диаграмм для роботов и BPMN (Business Process Model and Notation) редактор. Как и любой другой программный продукт, WMP должен тестироваться, но, в отличие от большинства web проектов, главная функциональность клиентской части WMP – составление диаграмм. Существуют инструменты GUI тестирования, позволяющие имитировать нажатие на кнопку, ввод текста и других элементарных действий, но они не подходят для конструктора диаграмм ввиду того, что они слишком низкоуровневы. Необходимы более высокоуровневые операции, которые бы позволили производить элементарные манипуляции с диаграммой в рамках одной команды.

# Обзор

В начале обзора приводится общее описание проекта WMP. На основании его особенностей, формируются критерии для выбора средства графического тестирования. Осуществляется обзор средств тестирования и, учитывая приведенные критерии, происходит выбор подходящего.

## Описание проекта WMP

WMP – это web проект, и как любой другой сайт пользователю он представляется как набор страниц. Существуют страницы аутентификации, панели роботов, редактора диаграмм для роботов и BPMN редактора. Чтобы зайти на любую страницу, кроме первой, необходимо авторизоваться. Редакторы диаграмм однотипны, они состоят из компонент, перечисленных ниже.

* Сцена – область экрана, на которой размещаются диаграммы. Диаграммы состоят из блоков и стрелок, соединяющих некоторые из них.
* Палитра – область экрана, где расположен контейнер с различными типами блоков. Их можно перемещать на сцену.
* Редактор свойств – специальная область экрана, на которой отображаются имена свойств и их значения для выделенного в данный момент времени блока диаграммы.
* Верхнее меню – горизонтальное меню в верхней области экрана, используется для сохранения/открытия диаграммы и других операций с проектом.

## Критерии выбора инструмента GUI тестирования

1. Инструмент должен распространяться бесплатно.
2. Наличие функций поиска, захвата элемента и его перемещения. Они необходимы для манипуляций с диаграммами.
3. Доступность визуализации тестирующих скриптов. Каждый редактор WMP – это среда визуального программирования, и к ним необходимо создавать инструменты для обучения. Одним из таких инструментов является анимация.

## Автоматизация GUI тестирования

Существуют несколько флагманов среди инструментов автоматизации GUI тестирования, которые обычно используются в проектах. В частности, TestCompile, Ranorex Studio и Selenium WebDriver. К сожалению, только последний из них распространяется бесплатно. Кроме того, на рынке бесплатных инструментов следует выделить Sikuli и SoapUI. Sikuli построен на поиске изображений. Для того, чтобы найти какой-либо элемент, необходимо сначала загрузить его картинку. В диаграммах WMP могут содержаться разные элементы с одинаковыми изображениями, и из-за этого могут возникнуть сложности с обработкой. SoapUI, в свою очередь, чаще применяют как инструмент интеграционного тестирования. Он больше ориентирован на тестирование серверной части приложения.

Одним из самых популярных средств тестирования GUI является Selenium WebDriver. Он предоставляет набор библиотек, с помощью которых можно обращаться к браузеру, как будто это делает пользователь. На данный момент Selenium практически стал стандартом драйвера к браузеру, предоставляя весь необходимый API для поиска и манипуляций с элементами. Из перечисленных компонент именно WebDriver удовлетворяет всем поставленным критериям.

# Разработанное решение

Чтобы тестировать редакторы, сначала нужно зайти на страницу аутентификации, ввести логин и пароль, и уже потом переходить на нужную страницу браузера.

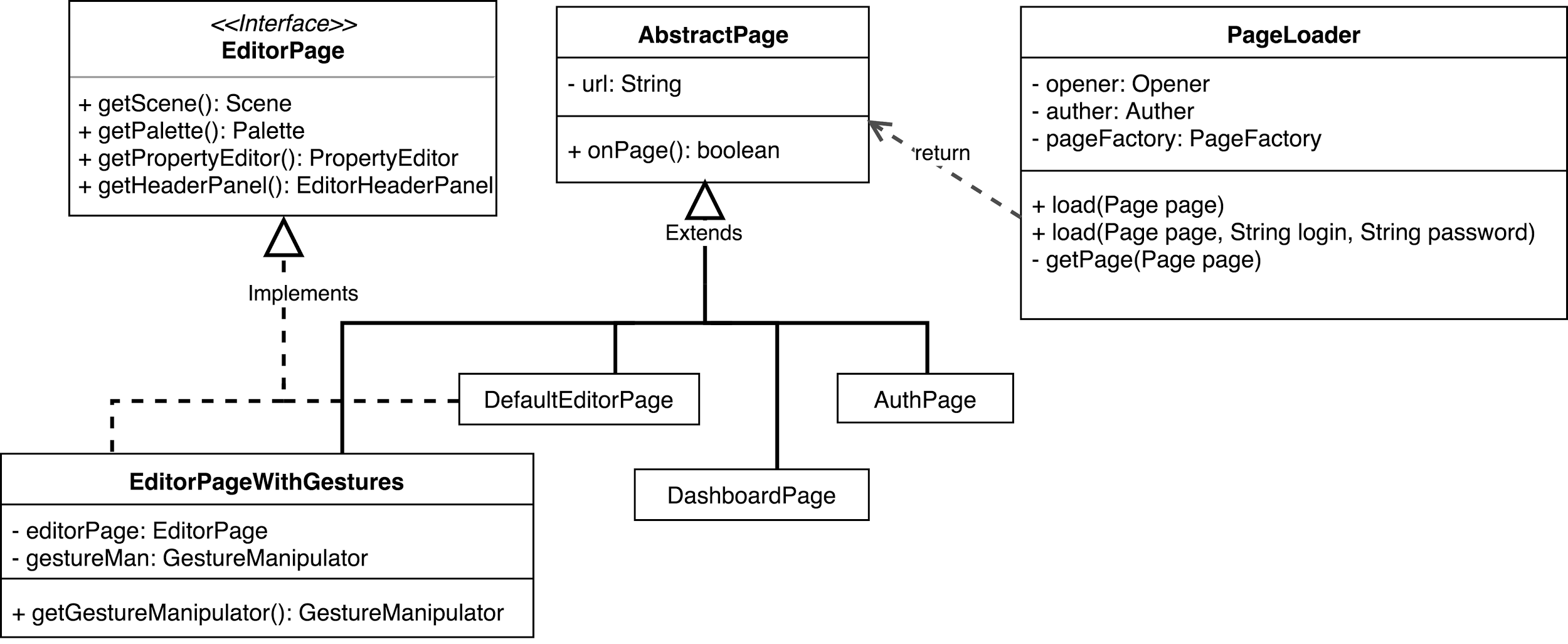


Рисунок 1: Архитектура системы навигации разработанного решения для GUI тестирования WMP

В предлагаемом решении за загрузку страниц отвечает класс PageLoader. Вызов метода load с одной стороны открывает запрошенную страницу в браузере, с другой – возвращает наследника класса AbstractPage. Страницы редакторов реализуют интерфейс EditorPage и предоставляют сцену, палитру, редактор свойств и верхнее меню в качестве сервисов.

## Сервисы тестирования конструктора диаграмм

Предоставляемые сервисы разработанного решения позволяют имитировать следующие действия:

1. перетаскивать блоки из палитры на сцену;
2. перемещать блоки внутри сцены;
3. удалять блоки со сцены;
4. добавлять/удалять стрелки между элементами;
5. задавать свойства блокам.

Также реализован весь необходимый API для проверки корректности результата перечисленных действий.

Сцена, палитра и редактор свойств работают с объектами класса SceneElement. SceneElement является классом родителем для всех элементов, которые находятся на сцене. Он агрегирует в себе WebElement, соответствующий отображаемому элементу на экране.

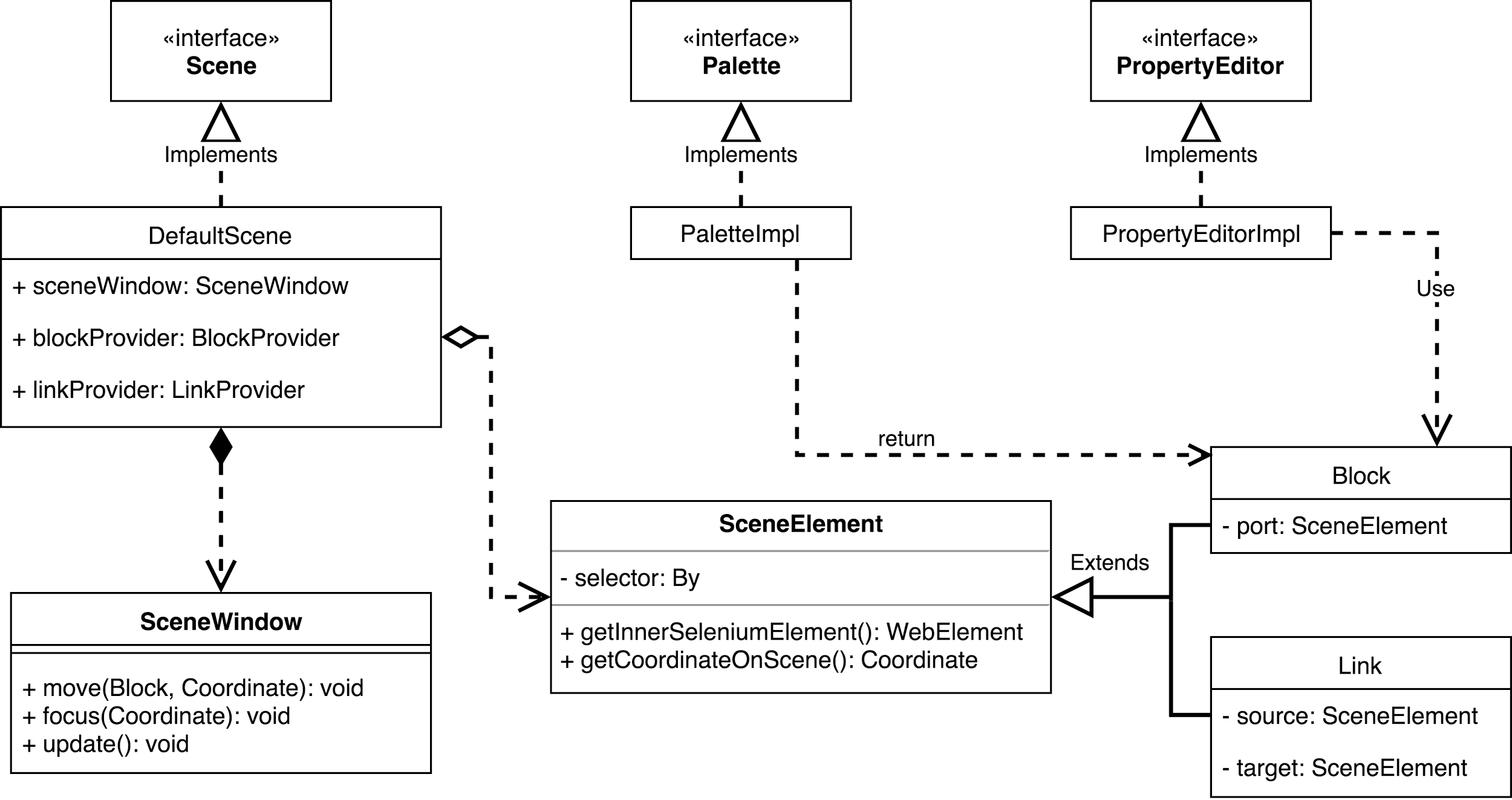


Рисунок 2: Архитектура разработанного решения для конструктора диаграмм

В реальности сцена больше, чем та часть, которая отображается пользователю. И, чтобы переместить элемент из одного края сцены в другой, требуется одновременно и двигать элемент в пределах экрана, и смещать сам экран. Эту функциональность реализует класс SceneWindow.

## Сервисы тестирования верхнего меню

В настоящее время верхнее меню редактора содержит кнопку Dashboard, осуществляющую переход на страницу панели роботов и пункт File, по клику на который открывается дополнительное меню. Если в нем выбрать пункты open или save, появится окно дерева папок. Внутри него можно создавать и удалять папки и подпапки, переходить вверх и вниз по каталогу. В рамках работы реализован API для имитации и тестирования всех сценариев работы с ним.

Также был реализован сервис для имитации и тестирования сохранения/открытия диаграмм. Он позволяет сохранять/открывать диаграммы по указанному пути в каталоге папок. Помимо этого он позволяет сравнивать диаграммы на эквивалентность.

## Сервис тестирования жестов мыши

Редактор роботов поддерживает жесты мышью. Если зажать правую кнопку мыши и начать перемещать мышь, она будет оставлять след. Если траектория движения описывает тот или иной элемент палитры, он появится на сцене. Также, если навести курсор на блок на сцене и зажать правую кнопку мыши, а после этого переместить мышь к другому блоку на сцене, и отпустить правую кнопку, эти блоки соединятся стрелкой. В рамках данной работы реализован API для имитации и тестирования этой функциональности. Помимо функций Selenium, данный сервис использует библиотеку java.awt.Robot. Она используется для захвата мыши.

# Файлы конфигураций

Selenium WebDriver осуществляет поиск элемента по его CSS-селектору. Если изменить селектор элемента на стороне клиента, возникает необходимость менять значение этого селектора и на стороне GUI тестирования. Отсюда возникает потребность в файлах конфигураций, где будут указываться и элемент, участвующий в тестировании, и его css селектор. В силу того, что элементов, участвующих в тестировании довольно много, предлагается перенести отношение вложенности элементов друг в друга в HTML представлении и на файлы конфигурации.

# Апробация

Предложенного API достаточно для составления достаточно сложных пользовательских сценариев. Ниже приведен пример такого сценария.  
// загрузка страницы редактора роботов  
EditorPage ed = pageLoader.load(Page.EditorRobots);

// перемещение блока “Initial Node” из палитры на сцену  
Block begin = ed.getScene().dragAndDrop(  
 e.getPalette().getElement("Initial Node"), 4, 4);   
// добавление блока “Final Node” с помощью жестов мыши

Block end = ed.getGestureManipulator().draw(“Final Node”);

// проверка, что создался правильный блок   
assert (“Final Node”).equals(end.getName())

// соединение блок стрелкой с помощью жестов мыши  
ed.getGestureManipulator().drawLine(begin, end);

// сохранение диаграммы в папке test под именем test  
ed.getHeaderPanel().saveDiagram(“test/test”);

После запуска приведенного кода откроется браузер, и шаг за шагом все описанные действия будут исполнены. Если включить запись видео с экрана, можно записать обучающую анимацию.

# **Заключение**

В документе были представлены основные средства GUI тестирования, приведено описание проекта WMP, сформулированы критерии выбора подходящего средства GUI тестирования, предложен API для тестирования редактора диаграмм, приведены некоторые подробности решения.

**Литература**

1. Yogesh Kumar. Comparative Study of Automated Testing Tools: Selenium, SoapUI, HP Unified Functional Testing and Test Complete. //JETIR, September 2015.   
   –– URL: <http://www.jetir.org/papers/JETIR1509007.pdf> [дата просмотра 01.11.2016].
2. Ranorex Vs. Selenium   
   –– URL: <http://www.ranorex.com/ranorex-vs-selenium.html>   
   [дата просмотра 01.11.2016].
3. Selenium WebDriver Reference.   
   –– URL: http://www.seleniumhq.org/docs/03\_webdriver.jsp [дата просмотра 01.11.2016].

1. Web Modeling Project – URL: https://github.com/qreal/wmp [↑](#footnote-ref-0)