Оглавление

Перечень сокращений 3
Терминология 3
1. Введение 4
2. Предпроектное исследование 6
2.1. Основные положения языка РДО[2] 6
2.2. Система имитационного моделирования RAO-studio[2] 6
2.3. Графики в системе имитационного моделирования RAO-studio[2] 8
2.4. Система имитационного моделирования RAO-XT 8
3. Формирование ТЗ 10
3.1. Введение 10
3.2. Общие сведения 10
3.3. Назначение разработки 10
3.4. Требования к программе или программному изделию 10
3.4.1. Требования к функциональным характеристикам 10
3.4.2. Требования к надежности 11
3.4.3. Условия эксплуатации 11
3.4.4. Требования к составу и параметрам технических средств 11
3.4.5. Требования к информационной и программной совместимости 12
3.4.6. Требования к маркировке и упаковке 12
3.4.7. Требования к транспортированию и хранению 12
3.5. Требования к программной документации 12
3.6. Стадии и этапы разработки 12
3.7. Порядок контроля и приемки 12
4. Концептуальный этап проектирования подсистемы 13
4.1. Настройка слайдеров 13
4.2. Выбор удобного способа масштабирования 13
4.3. Разработка алгоритма непосредственного вывода координат точек на графике 13

4.4. Разработка алгоритма построения графиков с одной или совпадающими точками. 14
5. Технический этап проектирования подсистемы 15
5.1. Настройка слайдеров 15
5.2. Создание удобного способа масштабирования 15
5.3. Вывод координат точек на графике 15
5.4. Построение графиков с одной или совпадающими точками 17
6. Рабочий этап проектирования подсистемы 19
6.1. Реализация правильной работы слайдеров 19
6.2. Реализация удобного способа масштабирования 19
6.3. Реализация перевода координат точки графика в координаты виджета. 21
6.4. Реализация вывода окошка с координатами точек на графике 21
Список используемых источников 23
Список использованного программного обеспечения 23

Перечень сокращений

ИМ – Имитационное Моделирование

СДС – Сложная Дискретная Система

IDE - Integrated Development Environment (ИнтегрированнаяСредаРазработки)

Терминология

Плагин – независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей

График – диаграмма, изображающая при помощи кривых количественные показатели развития, состояния модели

Подсистема визуализации – подсистема среды RAO-XT, отвечающая за построение графиков изменения состояния модели

Парсинг – процесс анализа последовательности входных данных и преобразование их в необходимых формат

Парсер – компонент, выполняющий парсинг данных

Сериализация – процесс перевода какой-либо структуры данных в последовательность битов

Слайдер – элемент графического интерфейса, позволяющий отображать область окна, соответствующую его положению

1. Введение

Имитационное моделирование (ИМ)[1] на ЭВМ находит широкое применение при исследовании и управлении сложными дискретными системами (СДС) и процессами, в них протекающими. К таким системам можно отнести экономические и производственные объекты, морские порты, аэропорты, комплексы перекачки нефти и газа, ирригационные системы, программное обеспечение сложных систем управления, вычислительные сети и многие другие. Широкое использование ИМ объясняется тем, что размерность решаемых задач и неформализуемость сложных систем не позволяют использовать строгие методы оптимизации. Эти классы задач определяются тем, что при их решении необходимо одновременно учитывать факторы неопределенности, динамическую взаимную обусловленность текущих решений и последующих событий, комплексную взаимозависимость между управляемыми переменными исследуемой системы, а часто и строго дискретную и четко определенную последовательность интервалов времени. Указанные особенности свойственны всем сложным системам.

Проведение имитационного эксперимента позволяет:

1. Сделать выводы о поведении СДС и ее особенностях:
    - без ее построения, если это проектируемая система
    - без вмешательства в ее функционирование, если это действующая система, проведение экспериментов над которой или слишком дорого, или небезопасно
    - без ее разрушения, если цель эксперимента состоит в определении пределов воздействия на систему
2. Синтезировать и исследовать стратегии управления.
3. Прогнозировать и планировать функционирование системы в будущем.
4. Обучать и тренировать управленческий персонал и т.д.

ИМ является эффективным, но и не лишенным недостатков, методом. Трудности использования ИМ, связаны с обеспечением адекватности описания системы, интерпретацией результатов, обеспечением стохастической сходимости процесса моделирования, решением проблемы размерности и т.п. К проблемам применения ИМ следует отнести также и большую трудоемкость данного метода.

Интеллектуальное ИМ, характеризующиеся возможностью использования методов искусственного интеллекта и прежде всего знаний, при принятии решений в процессе имитации, при управлении имитационным экспериментом, при реализации интерфейса пользователя, создании информационных банков ИМ, использовании нечетких данных, снимает часть проблем использования ИМ.

2. Предпроектное исследование

2.1. Основные положения языка РДО[2]

Основные положения системы РДО могут быть сформулированы следующим образом[1]:

- Все элементы СДС представлены как ресурсы, описываемые некоторыми параметрами. Ресурсы могут быть разбиты на несколько типов; каждый ресурс определенного типа описывается одними и теми же параметрами.

- Состояние ресурса определяется вектором значений всех его параметров; состояние СДС - значением всех параметров всех ресурсов.

- Процесс, протекающий в СДС, описывается как последовательность целенаправленных действий и нерегулярных событий, изменяющих определенным образом состояние ресурсов; действия ограничены во времени двумя событиями: событиями начала и событиями конца.

- Нерегулярные события описывают изменения состояния СДС, непредсказуемые в рамках продукционной модели системы (влияние внешних по отношению к СДС факторов либо факторов, внутренних по отношению к ресурсам СДС). Моменты наступления нерегулярных событий случайны.

- Действия описываются операциями, которые представляют собой модифицированные продукционные правила, учитывающие временные связи. Операция описывает предусловия, которым должно удовлетворять состояние участвующих в операции ресурсов, и правила изменения состояния ресурсов в начале и в конце соответствующего действия.

- Множество ресурсов R и множество операций O образуют модель СДС.

2.2. Система имитационного моделирования RAO-studio[2]

Программный комплекс RAO-studio предназначен для разработки и отладки имитационных моделей на языке РДО. Основные цели данного комплекса - обеспечение пользователя легким в обращении, но достаточно мощным

средством разработки текстов моделей на языке РДО, обладающим большинством функций по работе с текстами программ, характерных для сред программирования, а также средствами проведения и обработки результатов имитационных экспериментов.

В соответствии с основной целью программный комплекс решает следующие задачи:

- синтаксический разбор текста модели и настраиваемая подсветка синтаксических конструкций языка РДО
- открытие и сохранение моделей
- расширенные возможности для редактирования текстов моделей
- автоматическое завершение ключевых слов языка
- поиск и замена фрагментов текста внутри одного модуля модели
- поиск интересующего фрагмента текста по всей модели
- навигация по тексту моделей с помощью закладок
- наличие нескольких буферов обмена для хранения фрагментов текста
- вставка синтаксических конструкций языка и заготовок (шаблонов) для написания элементов модели
- настройка отображения текста моделей, в т.ч. скрытие фрагментов текста и масштабирование
- запуск и остановка процесса моделирования
- изменение режима моделирования
- изменение скорости работающей модели
- переключение между кадрами анимации в процессе моделирования
- отображение хода работы модели в режиме реального времени
- построение графиков изменения интересующих разработчика характеристик в режиме реального времени
- обработка синтаксических ошибок при запуске процесса моделирования

- обработка ошибок во время выполнения модели
- обеспечение пользователя справочной информацией

2.3.Графики в системе имитационного моделирования RAO-studio[2]

Имеется возможность визуально отображать процессы, происходящие в модели, в виде графиков. Добавление графиков состояния ресурса становится возможным только после запуска модели. Графики состояния ресурса можно добавлять как в процессе работы модели в режиме анимации, так и после завершения моделирования. График состояния ресурса может быть отображен только в том случае, если конкретный ресурс трассируется. Для добавления графика во вкладке Графики окна объектов в дереве модели следует найти необходимый вам параметр ресурса. Далее, чтобы создать новое окно графика, нужно щелкнуть два раза по выбранному параметру ресурса левой кнопкой мыши или один раз правой и в выпадающем меню выбрать команду «Plot». По оси абсцисс графика откладывается время наступления событий, при которых изменяется значение параметра выбранного ресурса.

При первом построении графика, масштаб по оси ординат устанавливается автоматически. Для изменения масштаба по оси абсцисс можно воспользоваться панелью инструментов Масштаб или использовать выпадающее меню при правом щелчке мышки по области графика. Основными функциями масштабирования являются:

- Увеличить масштаб – увеличение масштаба
- Уменьшить масштаб – уменьшение масштаба
- Автоматический масштаб – автоматический подбор масштаба по ширине области графика
- Восстановить масштаб – восстановление начального масштаба

2.4.Система имитационного моделирования RAO-XT

Система имитационного моделирования RAO-XT представляет собой плагин для интегрированной среды разработки Eclipse, позволяющий вести разработку имитационных моделей на языке РДО. Система написана на языке Java[3] и состоит из трех основных компонентов:

- rdo – компонент, производящий преобразование кода на языке РДО в код на языке Java

- rdo.lib – библиотека системы. Этот компонент реализует ядро системы имитационного моделирования
- rdo.ui – компонент, реализующий графический интерфейс системы с помощью библиотеки SWT[4]

На момент начала выполнения курсового проекта, система имела недостаточный набор инструментов для удобного пользования графиками. Также были обнаружены следующие ошибки в подсистеме:

- не строятся графики с одной точкой
- не работают слайдеры при масштабировании через меню

3. Формирование ТЗ

3.1.Введение

Программный комплекс RAO-XT предназначен для разработки и отладки имитационных моделей на языке РДО. Основные цели данного комплекса - обеспечение пользователя легким в обращении, но достаточно мощным средством разработки текстов моделей на языке РДО, обладающим большинством функций по работе с текстами программ, характерных для сред программирования, а также средствами проведения и обработки результатов имитационных экспериментов.

Текущая подсистема визуализации обладает рядом существенных недостатков, затрудняющих получение информации с графиков и замедляющие работу пользователя. Основные операции масштабирования реализованы через контекстное меню, что замедляет работу пользователя. Большой шаг масштабирования и нерабочие слайдеры осложняют восприятие информации с модели.Другими важными недоработками являются отсутствие возможности получения точного значения точки прямо на графике и неспособность подсистемы строить графики из одной или из нескольких совпадающих точек. Что может привести к неправильному пониманию результатов построения.

3.2.Общие сведения

Основание для разработки: задание на курсовой проект.

Заказчик: Кафедра «Компьютерные системы автоматизации производства» МГТУ им. Н.Э. Баумана

Разработчик: студент кафедры «Компьютерные системы автоматизации производства» Рахматулин В.В.

Наименование темы разработки: «Доработка подсистемы вывода графиков для системы имитационного моделирования RAO-XT»

3.3.Назначениеразработки

Доработать подсистему вывода графиковмодели для системы имитационного моделирования RAO-XT, устранить ошибки.

3.4.Требования к программе или программному изделию

3.4.1. Требования к функциональным характеристикам

Подсистема визуализации должна удовлетворять следующим требованиям:

- Построение по различным параметрам модели:
    - по параметрам ресурсов
    - выполняемым образцам
    - результатам
- Построение графиков в процессе прогона
- Построение ступенчатых графиков
- Построение по различным типам данных
    - *integer*
    - *real*
    - *enum*
- Удобная система масштабирования с мелким шагом
- Независимость от модуля трассировки
- Возможность получать значения на графике через всплывающее окошко

3.4.2. Требования к надежности

Основное требование к надежности направлено на поддержание в исправном и работоспособном состоянии ЭВМ, на которой происходит использование программного комплекса RAO-XT.

3.4.3. Условия эксплуатации

- Эксплуатация должна производиться на оборудовании, отвечающем требованиями к составу и параметрам технических средств, и с применением программных средств, отвечающим требованиям к программной совместимости
- Аппаратные средства должны эксплуатироваться в помещениях с выделенной розеточной электросетью 220В ±10%, 50 Гц с защитным заземлением

3.4.4. Требования к составу и параметрам технических средств

Программный продукт должен работать на компьютерах со следующими характеристиками:

- объем ОЗУ не менее 2 Гб
- объем жесткого диска не менее 50 Гб
- микропроцессор с тактовой частотой не менее 2ГГц
- монитор с разрешением от 1366*768 и выше

3.4.5. Требования к информационной и программной совместимости

Система должна работать под управлением следующих ОС: Windows 7, Ubuntu 14.10.

3.4.6. Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

3.4.7. Требования к транспортированию и хранению

Требования к транспортированию и хранению не предъявляются.

3.5. Требования к программной документации

Требования к программной документации не предъявляются.

3.6. Стадии и этапы разработки

Плановый срок начала разработки – 1октября 2016 г.

Плановый срок окончания разработки – 20 декабря 2016г.

Этапы разработки:

- Концептуальный этап проектирования системы
- Технический этап проектирования системы
- Рабочий этап проектирования системы

3.7. Порядок контроля и приемки

Контроль и приемка работоспособности системы осуществляются с помощью ручного тестирования подсистемы визуализации.

4. Концептуальный этап проектирования подсистемы

На концептуальном этапе проектирования требовалось:

- Найти и выявить ошибки, связанные с неправильной работой слайдеров при масштабировании через контекстное меню
- Разработать удобный способ масштабирования графиков
- Разработать алгоритм для непосредственного вывода данных на графике
- Найти способ строить графики, состоящие из одной точки

4.1. Настройка слайдеров

Проблема с появлением слайдеров проявляется, когда вызываются команды масштабирования через меню. Поэтому будем обновлять слайдеры после проведения каждой операции масштаба.

4.2. Выбор удобного способа масштабирования

На данный момент основные операции масштабирования вызываются через меню. В связке с неработающими слайдерами такой подход неудобен. Разработана концепция масштабирования через колесо мышки. Шаг следует уменьшить, как минимум, вдвое. Выбор оси масштабирования будет реализован через зажатие клавиш на клавиатуре. По оси абсцисс- с помощью Ctrl, по оси ординат - через Shift. При одновременном зажатии клавиш - сразу по обеим осям. ПО таким же принципам выделяется прямоугольная область методом `zoom(Rectangleselection)`. Клавиши в отпущенном состоянии запрещают увеличение-уменьшение графика.

4.3. Разработка алгоритма непосредственного вывода координат точек на графике

Была разработана схема отображения всплывающей подсказки. При наведении курсора на график будет вычисляться ближайшая к нему точка. Если курсор достаточно близок хотя бы к одной точке графика, то будет выводиться сообщение в виде окошка с координатами ближайшей. Точка будет выделяться выколотым кружком. Если мышка покидает поле графика, то гаснет окно и исчезает выделение фигурой.

4.4. Разработка алгоритма построения графиков с одной или совпадающими точками.

Был разработан алгоритм построения графиков с одной точкой. Так как системой построения графиков используется StepRendering (ступенчатая отрисовка), то для устранения проблемы необходимо будет в конце симуляции добавить еще одну точку. Это позволит механизму провести прямую. Чтобы отследить необходимость добавления, мы создадим подписчик, который будет срабатывать при получении сообщения о конце симуляции. В классе, который отвечает за постоянное обновление данных графика, будет проверяться флаг подписчика и количество данных на графике.

5. Технический этап проектирования подсистемы

5.1. Настройка слайдеров

Управление слайдерами осуществляется классом PlotFrame. Вметоде`setSliders(final Sliderhorizontal Slider, final Slidervertical Slider)` задаются горизонтальныйивертикальныйслайдерыиихпараметры. Внутри метода заданы два анонимных класса, добавляющие слушатели событий `SelectionListener()`к слайдерам. Они позволяют реализовывать прокрутку графиков. В методе `updateSliders()` происходит обновление размеров слайдеров в соответствии с масштабом графика. Следует отметить, что на текущий момент задано соответствие размера и масштаба только для horizontalslider, поэтому добавим код для verticalslider. Чтобы гарантировать появление слайдеров при любом вызове масштабирования, необходимо добавить метод `updateSliders()`после каждого вызова методов увеличения-уменьшения.

5.2. Создание удобного способа масштабирования

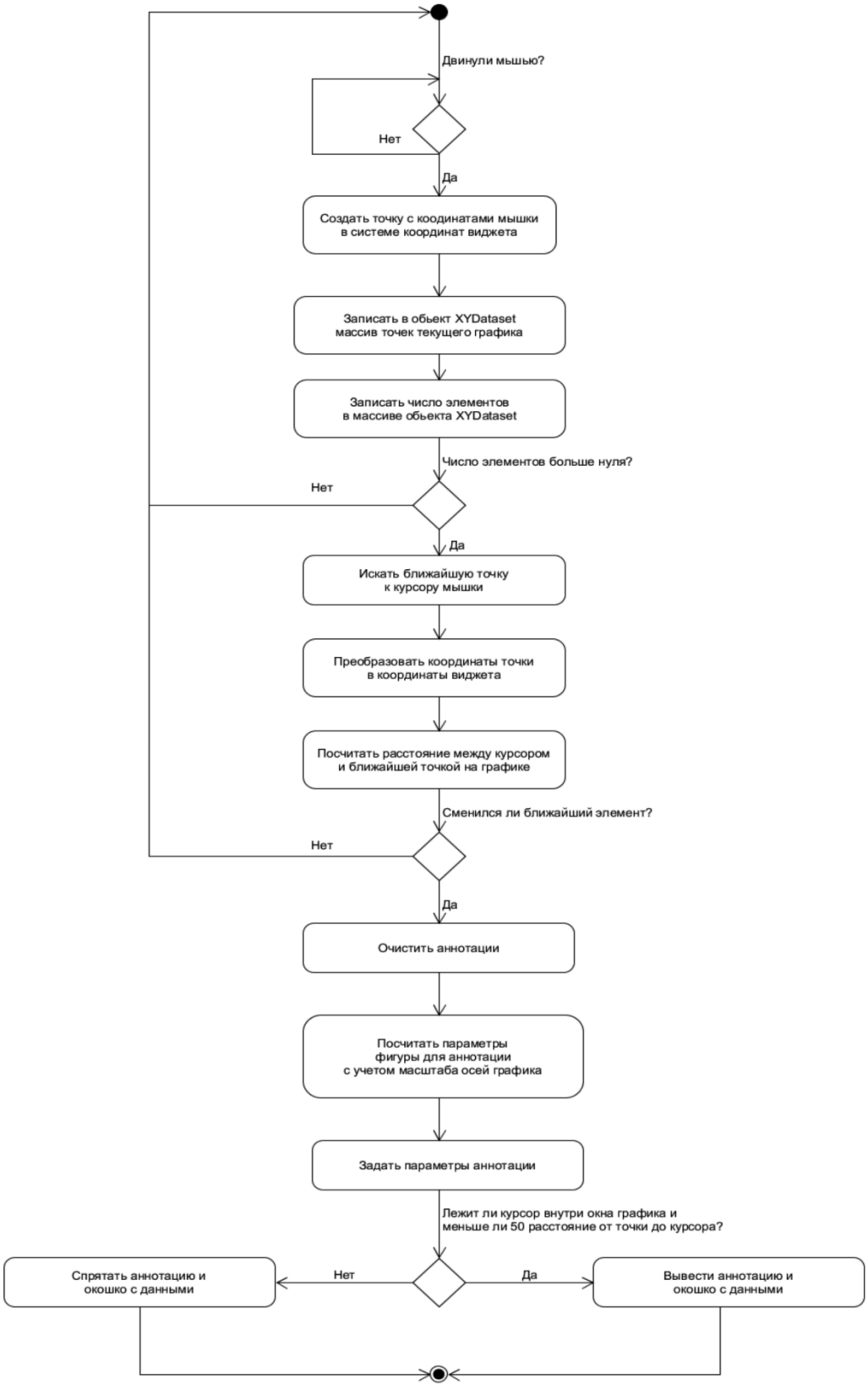
Методами внутреннего классаPlotKeyListener, реализующего интерфейс KeyListener, в Plotframe, будем отслеживать нажатия на кнопки и задавать значения trueдля флагов, отвечающих за масштаб осей. Нажатие на пробел будет вызывать метод `restoreAutoBounds()`, который возвращает исходный масштаб у графика. Методами внутреннего класса PlotMouseWheelListener,реализующем интерфейс MouseWheelListener, при прокрутке колесика будет проверяться значение флага, и запускаться необходимая команда масштаба.

5.3. Вывод координат точек на графике

Методом внутреннего класса PlotMouseMoveListener, реализующего интерфейс MouseMoveListener. Схема приведена ниже.

с помощью `mouseMove(MouseEvent e)`мы перехватываем движение мышью. Внутриметодапроводитсярасчет расстояния от ближайшей точки до курсора

в переменную distanceToMouse. Если индекс точки сменился и distanceToMouse<50,то высвечивается окошко со значениями координат ближайшей точки в системе графика. Подсвечивание точки выполняется с помощью метода XYShapeAnnotation(shape: shape,stroke: stroke,Color: color, Color: color). В качестве Shapeпередается обьектEllipse2D, который предварительно рассчитывается исходя из масштаба графика.



5.4.Построение графиков с одной или совпадающими точками

Создаем подписчик endSubscriber, подписанный на состояние симулятора «execution_completed». Он меняет значение флага isLastEntryна true.

Расширяем метод run() во внутреннем классеPlotViewRealTimeUpdateRunnable. При значении isLastEntry=truevыполняется проверка на количество точек. Если одна или две совпадающие, то добавляется точка в конец графика. Схему можно привести ниже:

Навигация по графику происходит с помощью вертикального и горизонтального слайдеров.

Размеры и положение слайдеров зависят от масштабирования, поэтому при каждом изменении масштаба графика необходимо переопределять данные характеристики слайдеров. Это выполняется в методах *setSliders()* и *setSlidersMaximum()* в графической части подсистемы визуализации.

6. Рабочий этап проектирования подсистемы

На рабочем этапе проектирования системы были реализованы разработанные на предыдущих этапах схемы и концепции.

6.1. Реализация правильной работы слайдеров

Для корректного обновления слайдеров, необходимо дополнить метод `updateSliders()`условием для пересчетаverticalslider.Условие аналогичное horizontalslider:

```
if (verticalMaximum - rangeAxis.getRange().getUpperBound() >sliderConst) {
                    verticalSlider.setVisible(true);
                    verticalSlider.setEnabled(true);

verticalRatio = verticalSlider.getMaximum() / verticalMaximum;
verticalSlider.setThumb((int) Math.round((rangeAxis.getUpperBound() -
rangeAxis.getLowerBound()) * verticalRatio));
                    verticalSlider
                                    .setSelection((int) Math.round((verticalMaximum -
rangeAxis.getUpperBound()) * verticalRatio));
             } else {
                    verticalSlider.setVisible(false);
                    verticalSlider.setEnabled(false);
             }
```

Чтобы обновление размеров слайдеров выполнялось при исполнении любого метода масштабирования, следует в каждый из них добавить метод`updateSliders()`.

```
@Override
      publicvoidzoomInDomain(double x, double y) {
            super.zoomInDomain(x, y);
            updateSliders();
      }
```

С помощью ключевого слова super вызывается стандартный метод из класса ChartComposite. В след за ним в переписанном методе масштаба вызывается`updateSliders()`.

6.2. Реализацияудобного способа масштабирования

В конструкторе PlotFramerегистрируем слушатели клавиатуры иколесамыши:

```
addSWTListener(newPlotKeyListener());
addMouseWheelListener(newPlotMouseWheelListener());
```

Задаем шаг масштабирования:`setZoomInFactor(0.75);  setZoomOutFactor(1.25);`

В методах класса PlotKeyListenerсоздается переключатель, который меняет значение флагов isRangeZoomableиisDomainZoomable. Нажатие на пробел сбрасывает масштаб в исходное состояние.

```
@Override
		publicvoidkeyPressed(KeyEvent e) {
			switch (e.keyCode) {
			case SWT.SHIFT: {
				setRangeZoomable(true);
				return;
			}
			case SWT.CTRL: {
				setDomainZoomable(true);
				return;
			}
			case SWT.SPACE:
				restoreAutoBounds();
				return;
			}
		}
@Override
		publicvoidkeyReleased(KeyEvent e) {
			if (e.keyCode == SWT.SHIFT) {
				setRangeZoomable(false);
			} elseif (e.keyCode == SWT.CTRL) {
				setDomainZoomable(false);
			}
		}
	}
```

Переменная e.count отвечает за направление прокрутки колеса мыши.

```
	classPlotMouseWheelListenerimplementsMouseWheelListener {

		@Override
		publicvoidmouseScrolled(MouseEvent e) {

			if (e.count> 0 &&isRangeZoomable()) {
				zoomInRange(e.x, e.y);
			}
			if (e.count< 0 &&isRangeZoomable()) {
				zoomOutRange(e.x, e.y);
			}
			if (e.count> 0 &&isDomainZoomable()) {
				zoomInDomain(e.x, e.y);
			}
			if (e.count< 0 &&isDomainZoomable()) {
				zoomOutDomain(e.x, e.y);
			}
			if (e.count> 0 &&isDomainZoomable() &&isRangeZoomable()) {
				zoomInBoth(e.x, e.y);
			}
			if (e.count< 0 &&isDomainZoomable() &&isRangeZoomable()) {
				zoomOutBoth(e.x, e.y);
			}
		}
	}
```

6.3. Реализация перевода координат точки графика в координаты виджета.

Метод `plotToSwt(doublex, doubley)` понадобится для решения задачи с выводом всплывающей подсказки с координатами точек.

Обьектrectanglecодержит в себе прямоугольную область внутри осей графика.

```
Rectangle rectangle = PlotFrame.this.getScreenDataArea();
```

Обьекты класса ValueAxisdomainAxis и rangeAxis содержат в себе оси графика.

```
finalValueAxisdomainAxis = getChart().getXYPlot().getDomainAxis();
finalValueAxisrangeAxis = getChart().getXYPlot().getRangeAxis();
```

width_in_axis_units содержит в себе ширину абсцисс в единицах оси.

```
doublewidth_in_axis_units = domainAxis.getUpperBound() - domainAxis.getLowerBound();
```

height_in_axis_units содержит в себе высоту ординаты в единицах оси.

```
            doubleheight_in_axis_units = rangeAxis.getUpperBound() -
rangeAxis.getLowerBound();

            doublerelativeX = (x - domainAxis.getLowerBound()) /
width_in_axis_units;
            doublerelativeY = (rangeAxis.getUpperBound() - y) /
height_in_axis_units;
            doublescreenX = relativeX * rectangle.width + rectangle.x;
            doublescreenY = relativeY * rectangle.height + rectangle.y;

            returnnew Point((int) Math.round(screenX), (int) Math.round(screenY));
      }
```

Может нарисовать картинку для наглядности?

6.4. Реализация вывода окошка с координатами точек на графике

Задаем параметры окошка в конструкторе PlotFrame:

Определяемтипвиджета: `toolTip = newDefaultToolTip(this, SWT.BALLOON, true);`

Задаем цвета окошка:

```
toolTip.setBackgroundColor(new  Color(comp.getDisplay(),  255,  255,  255));
toolTip.setForegroundColor(new Color(comp.getDisplay(), 128, 128, 128));
```

Задаем смещение:`toolTip.setShift(new Point(0, -50));`

Параметры появления/исчезновения окошка:

```
toolTip.setRespectDisplayBounds(true);
toolTip.setHideDelay(0);
toolTip.setHideOnMouseDown(false);
toolTip.setPopupDelay(0);
```

Регистрируем слушатель движений мышки:

```
addSWTListener(newPlotMouseMoveListener());
```

Список используемых источников

1. **Емельянов В.В., Ясиновский С.И.** Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. - М.: "Анвик", 1998. - 427 с., ил. 136.
2. **Документация по языку РДО** [http://www.rdostudio.com/help/index.html]
3. **Брюс Эккель**. Философия Java
4. **Герберт Шилдт.** Java 8. Руководство для начинающих
5. **SWT Documentation** [https://www.eclipse.org/swt/docs.php]
6. **AWT Documentation** [https://docs.oracle.com/javase/8/docs/api/]
7. **Map Documentation** [ps://docs.oracle.com/javase/8/docs/api]

Список использованного программного обеспечения

1. RAO-Studio
2. Eclipse IDE for Java Developers Mars 2.0
3. openjdkversion "1.8.0_40-internal"
4. UMLet14.2
5. Inkscape 0.91
6. Microsoft® Office Word 2010
7. Microsoft® Visio 2010