Новосибирский государственный университет

Институт автоматики и электрометрии СО РАН, г. Новосибирск «Разработка программного модуля визуализации диаграмм процессов по спецификации на языке Reflex»

Выполнила Беленькая София Евгеньевна

Научный руководитель: зав. кафедрой КТ, д. т. н., доцент, зав. лаб. ИАиЭ СО РАН, Зюбин В.Е.

Соруководитель: Розов А. С., старший преподаватель кафедры КТ.

# Оглавление

[Введение 2](#_Toc37633584)

[Обзор диаграмм, предназначенных для статического анализа кода, а также IDE и прочих средств анализа кода с возможностью построения диаграмм. 5](#_Toc37633585)

[Средства построения диаграмм по коду для языков общего назначения. 6](#_Toc37633586)

[Средства разработки систем управления и ПО для встраиваемых систем. 11](#_Toc37633587)

[Разработка диаграмм для отображения связи процессов 16](#_Toc37633588)

[Обзор средств визуализации графов 20](#_Toc37633589)

[Обзор форматов представления графов. 25](#_Toc37633590)

[Архитектура модуля 28](#_Toc37633591)

[Реализация 29](#_Toc37633592)

[Список используемой литературы 30](#_Toc37633593)

# Введение

Существует несколько моделей разработки программного обеспечения, и в последнее время наибольшей популярностью пользуется итерационная модель разработки, при работе с которой возникает необходимость создания текущей рабочей документации, в частности, построении различных диаграмм по коду. Также подобная необходимость возникает при реверсивном инжиниринге и создании новых версий существующего программного продукта, так как диаграммы позволяют наглядно описать программу в выходной документации.

Для популярных языков общего назначения (таких, как Java, C++, С# и другие) реализовано множество программных средств автоматического построения диаграмм по коду, часть из которых встроены в интегрированные среды разработки (IDE). В то же время, при промышленной автоматизации, при написании управляющих алгоритмов и при программировании встраиваемых систем используются специализированные средства и языки: процесс-ориентированные языки (в частности, Reflex), или языки стандартов МЭК 61131-3. При этом процесс-ориентированное программирование (ПОП) показывает хорошие результаты и выигрывает по многим критериям, а стандарт МЭК 61131-3, в свою очередь, оказывается неудобным, как видно из работ [1], [2], [3].

Сейчас для ПОП диаграммы рисуются вручную, что занимает значительное время и может быть причиной ошибок ввиду человеческого фактора. В связи с этим возникает потребность автоматизации процесса создания и визуализации диаграмм для ПОП.

Цель работы: разработка программного модуля визуализации диаграмм процессов по спецификациям на языке Reflex. В рамках означенной цели поставлены следующие задачи:

* провести анализ специфики ПОП на языке Reflex с точки зрения необходимости визуализации диаграмм, анализ существующих видов диаграмм, использующихся для анализа кода, их сравнение;
* сформулировать требования к создаваемому программному модулю, определить формат представления данных;
* разработать архитектуру модуля, удовлетворяющую обозначенным требованиям;
* реализовать модуль визуализации;
* провести тестирование созданной реализации, опробовать ее на практике.

При визуализации диаграмм будет использоваться синтаксическое дерево разбора, построенное IDE или сторонним приложением. Таким образом, обеспечивается возможность встраивания модуля в состав IDE помимо возможности его независимой работы. Автоматическая визуализация диаграмм позволит улучшить такие критерии качества ПО, как расширяемость и масштабируемость, а также повысит сопровождаемость готовых проектов и упростит рецензирование кода (code review).

Работа состоит из трех глав, в первой из которых приведен анализ специфики ПОП и обзор средств автоматической генерации графических представлений по коду. Также в этой главе сформулированы требования к создаваемому программному модулю, определен формат представления данных и ключевые параметры, которые необходимо визуализировать. Во второй главе описывается разработанная архитектура модуля: транслятор, форматы промежуточного представления данных, выбранные средства визуализации. В третьей главе описана реализация программного модуля и приведены результаты его тестирования на практике.

# Обзор диаграмм, предназначенных для статического анализа кода, а также IDE и прочих средств анализа кода с возможностью построения диаграмм.

Рассмотрим рейтинг IDE, составленный на основе информации о частоте запросов загрузочной страницы в поисковике Google в 2019 году (см. Таблицу 1).

Таблица 1. Top IDE index [4] © Pierre Carbonnelle, 2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rank** | **Change** | **IDE** | **Share** | **Trend** |
| 1 | http://pypl.github.io/Up.pnghttp://pypl.github.io/Up.png | Android Studio | 23.38 % | +8.8 % |
| 2 | http://pypl.github.io/Down.png | Visual Studio | 21.81 % | -2.1 % |
| 3 | http://pypl.github.io/Down.png | Eclipse | 18.1 % | -5.3 % |
| 4 |  | NetBeans | 5.54 % | -0.5 % |
| 5 | http://pypl.github.io/Up.pnghttp://pypl.github.io/Up.pnghttp://pypl.github.io/Up.png | pyCharm | 4.89 % | +1.0 % |
| 6 | http://pypl.github.io/Up.pnghttp://pypl.github.io/Up.pnghttp://pypl.github.io/Up.png | Visual Studio Code | 4.88 % | +1.0 % |
| 7 | http://pypl.github.io/Down.pnghttp://pypl.github.io/Down.png | IntelliJ | 4.8 % | -0.0 % |
| 8 | http://pypl.github.io/Up.pnghttp://pypl.github.io/Up.png | Xcode | 3.91 % | +0.5 % |
| 9 | http://pypl.github.io/Down.pnghttp://pypl.github.io/Down.pnghttp://pypl.github.io/Down.png | Sublime Text | 3.66 % | -0.7 % |
| 10 | http://pypl.github.io/Down.pnghttp://pypl.github.io/Down.pnghttp://pypl.github.io/Down.png | Atom | 3.2 % | -0.9 % |
| 11 |  | Code::Blocks | 1.38 % | -0.4 % |
| 12 | http://pypl.github.io/Up.png | Vim | 0.89 % | -0.2 % |
| 13 | http://pypl.github.io/Down.png | Xamarin | 0.75 % | -0.4 % |

Проанализируем данный рейтинг с точки зрения наличия возможности построения UML-диаграмм. Зеленым цветом помечены строки с теми средствами разработки, в которых такая возможность есть, и красным – те, в которых она отсутствует. Из таблицы видно, что первая десятка популярных IDE в большинстве своем имеет встроенную возможность построения UML-диаграмм. Отсюда можно сделать вывод, что данный инструмент является востребованным. Рассмотрим подробно различные средства их построения.

## Средства построения диаграмм по коду для языков общего назначения.

1. Утилита Class Designer для Visual Studio способна динамически генерировать диаграмму классов по исходному коду, производить реверсивный инжиниринг и отслеживать наследственные связи с другими классами, а также реализует возможность перетаскивания элементов диаграммы и настройки видимости полей классов на ней. Для коммерческого использования юридическим лицом необходима платная лицензия при условии создания проекта с закрытым кодом. Visual Studio поддерживает широкий спектр языков и технологий, которые могут быть встроенными или предоставляться в виде расширений, в частности: C#, F#, Visual Basic, C++, Python, JavaScript, PHP, HTML, CSS, JavaScript и JSON.
2. Программная платформа моделирования Star UML, представляющая собой бесплатный проект с открытым кодом для 32-разрядных систем Windows, позволяет создавать диаграммы прецедентов, классов, сообщений, коллабораций, состояний, действий, компонентов, диаграммы развертывания, структурные диаграммы, и другие.
3. Бесплатная программная платформа Astah UML позволяет создавать следующие диаграммы: Class diagram (Object, Package, Subsystem and Robustness Diagrams are included), Use case diagram, Sequence diagram, Collaboration Diagram, State diagram, Activity diagram, Deployment diagram, Component diagram, способна генерировать Java 1.4 sourcecode из модели, а также импортировать Java 1.4 source files для генерации модели.
4. MagicDraw - визуальный UML , SysML , BPMN и UPDM инструмент моделирования, предоставляет возможности реверсивного инжиниринга, кодогенерации для следующих языков программирования: J2EE, C#, C++, CORBA IDL programming languages, .NET, XML Schema, WSDL, Java, EJB, DDL. Создает class, deployment, object, profile, package diagram, composite structural diagram, и другие. Требует платной лицензии.
5. Software Ideas Modeler - легкий и мощный инструмент для создания диаграмм UML и некоторых других диаграмм. Он поддерживает все 14 типов диаграмм UML, BPMN 2.0, SysML, ArchiMate, JSD, HTA, смешанную диаграмму, диаграмму потока данных, диаграмму последовательности операций, диаграмму надежности, диаграмму отношений сущностей, диаграмму требований, моделирование пользовательского интерфейса, диаграмму карты CRC, диаграмму параллелизма, дерево поведения, структурную диаграмму и т.д. Генерирует код для ActionScript, C#, C++, Delphi (Object Pascal), Java, JavaScript, PHP, Python, Ruby, SQL, Visual Basic, Visual Basic .NET, XML Schema, и осуществляет реверсивный инжиниринг для C#, C++, Java, PHP, Ruby, Visual Basic .NET. Требует платной лицензии для коммерческого использования.
6. BOUML – бесплатный конструктор UML-диаграмм, поддерживающий реверсивный инжиниринг для языков C++, Java, PHP, MYSQL, и кодогенерацию из диаграммы классов в них, а также в Python и IDL. Программа может строить диаграммы классов, последовательности, кооперации, объектов, прецедентов, компонентов, состояний, деятельности, диаграммы компонент и схем развертывания.
7. Пакет инструментов моделирования бизнес-процессов, ПО и баз данных Visual paradigm поддерживает 14 видов диаграмм: Class diagram, Use case diagram, Sequence diagram, Communication diagram, State machine diagram, Activity diagram, Component diagram, Deployment diagram, Package diagram, Object diagram, Composite structure diagram, Profile diagram, Timing diagram, Interaction overview diagram. Требует платной лицензии. Способен генерировать код для Java, C++, CORBA IDL, PHP, XML Schema, Ada, Python, C#, VB .NET, Object Definition Language (ODL), Flash ActionScript, Delphi, Perl, Objective-C и Ruby. Также осуществляет реверсивный инжиниринг для Java, C++, CORBA IDL, PHP, XML Schema, Ada, Python, C#, Java class, .NET dll и exe, JDBC.
8. Rational Rose представляет собой набор инструментов моделирования UML для разработки программного обеспечения. Есть функции кодогенерации и реверсивного инжиниринга для таких языков, как C++, Ada, Java/J2EE, Visual C++, Visual Basic и XML. Также существует возможность построения следующих диаграмм: Use Case diagram, Logical diagram, Component diagram, Deployment diagram, State diagram, Activity diagram, Iteration diagram. Больше не поддерживается.
9. Enterprise Architect – инструмент дизайна и визуального моделирования. Поддерживается кодогенерация и реверсивный инжиниринг для ActionScript, С, C #,C ++, Delphi, Java, PHP, Python, Visual Basic, Visual Basic .NET. ПО работает со следующими диаграммами: диаграмма классов, объектов, составная структурная диаграмма, диаграмма компонент, размещения, пакетов, взаимодействия, деятельности, функций, состояний, последовательностей, обзорные диаграммы потоков управления, коммуникационные диаграммы, временные диаграммы. Требует платной лицензии.
10. IntelliJ Idea – платная IDE, поддерживающая такие языки, как Java, Scala, Groovy, Kotlin, JavaScript, TypeScript, SQL и способная создавать диаграмму классов по коду. Также среда поддерживает module dependencies diagram.
11. Sybase PowerDesigner - средство UML проектирования с поддержкой кодогенерации для Java, C++, C#, VB .NET, Hibernate, EJB3, NHibernate, JSF, WinForm (.NET and .NET CF), PowerBuilder, также присутствуют возможности реверсивного инжиниринга для этих языков, исключая C++. Строит use-case, activity, sequence, диаграммы классов и компонентов, а также многое другое. Требует платной лицензии.
12. NetBeans - свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада, и ряда других. Поддерживаются следующие виды диаграмм: activity, class, sequence, state, use-case diagram, есть возможности реверсивного инжиниринга.
13. Lab view - это среда разработки и платформа для выполнения программ. Позволяет генерировать диаграммы классов. Поддерживает кодогенерацию в С-код. Требует платной лицензии.
14. NClass – это проект с открытым кодом. Поддерживает диаграммы классов и кодогенерацию для языков C# и Java. Полностью бесплатен.
15. Altova UModel 2008 – инструмент моделирования UML-диаграмм. Поддерживает кодогенерацию и реверсивный инжиниринг, генерацию документации для Java, C++, C#, Visual Basic и .NET. Требует платной лицензии. Работает с Use case diagrams, Class, object, Composite structure diagrams, Component, Deployment, Package, Profile diagram, Sequence, Activity, State machine diagrams, Interaction overview diagrams, Timing, Communication diagrams.

Более наглядно сравнение средств построения UML-диаграмм приведено в таблице 2.

Таблица 2. – Сравнение средств построения UML-диаграмм

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Название** | **Реверсивный инжиниринг** | **Кодогенерация** | **Возможность модификации** | **Динамическое построение** | **Возможность скрывать компоненты классов** | **Количество диаграмм** |
| Class Designer для Visual Studio | + | - | + | + | + | 1? |
| Star UML | - | - | + |  |  | 11 |
| Astah UML | + | + | + |  |  | 9 |
| MagicDraw | + | + | + | + \* |  | 24 |
| Software Ideas Modeler | + | + | + | +? | + | 14 |
| BOUML | + | + | + | -? | + | 10+ |
| Visual paradigm | + | + | + |  |  | 28 |
| Rational Rose | + | + | + | + | +? | 5+ |
| Enterprise Architect. | + | + | + |  |  | 12 |
| IntelliJ Idea | + | - | + | +? | + | 3 |
| Sybase PowerDesigner | + | + | + | -? | + | 3+ |
| NetBeans | + | - | + | -? | -? | 5+ |
| Lab view | + | + | + |  | - | 1 |
| NClass | - | + | + |  |  | 1 |
| Altova UModel 2008 | + | + | + |  |  | 14+ |

Были рассмотрены наиболее известные средства построения UML-диаграмм для языков общего назначения, таких, как Java, C# и другие. Исходя из специфики поставленной задачи, наибольший интерес представляют продукты, способные осуществлять реверсивный инжиниринг, так как в разрабатываемом программном модуле предполагается генерация диаграмм по коду. Средства построения, не удовлетворяющие данному критерию, выделены красным.

Из таблицы видно, что наиболее широкий спектр возможностей предоставляют такие программные продукты, как Class Designer для Visual Studio, Rational Rose и IntelliJ Idea, двое из которых также находятся в первой семерке популярности по количеству поисковых запросов в Google их загрузочной страницы. Отсюда можно сделать вывод, что наличие в IDE такого средства анализа кода, как визуализация диаграмм, дает преимущества и упрощает разработку. Все рассмотренные средства реализуют возможности модификации построенных диаграмм (drag-and-drop, изменение подписей и другие). Возможности скрывать компоненты отдельных классов и динамическое построение реализуется лишь частью рассмотренных средств.

Отсюда можно сформулировать следующие требования к решению:

* Автоматическое построение диаграмм процессов по коду.
* Обновление диаграммы по нажатию кнопки (по явному вызову), так как в данной области нет необходимости в динамическом построении диаграмм.
* Возможность редактирования диаграммы.
* Возможность сохранения диаграммы в отдельный файл для дальнейшего использования.
* Исходя из анализа ПОП, необходима визуализация следующих диаграмм:
* Диаграмм состояний процесса.
* Диаграмм связи процессов по данным.
* Диаграмм связи процессов по управлению.

## Средства разработки систем управления и ПО для встраиваемых систем.

Рассмотрим средства разработки систем управления и ПО для встраиваемых систем.

1. MATLAB: это высокоуровневый язык и интерактивная среда для программирования, численных расчетов и визуализации результатов. Требует платной лицензии. MATLAB Coder работает с Simulink Coder и Embedded Coder для генерации C кода из моделей Simulink, которые содержат код MATLAB.

Stateflow - это среда для моделирования и симуляции комбинаторной и последовательной логики принятия решений, основанных на машинах состояний и блок-схемах. Матрица перехода состояний генерируется из таблицы перехода состояний. Используя расширения для генерации кода, вы можете генерировать код PLC, C и C++ или HDL непосредственно из вашей диаграммы состояний.

1. CoDeSys - инструментальный программный комплекс промышленной автоматизации. Распространяется бесплатно демонстрационная версия, полная платно. Поддерживает IL, ST, LD, FBD, SFC, CFC. Поддерживаются основные 16- и 32-разрядные процессоры: Infineon C166, TriCore, 80x86, ARM (архитектура), PowerPC, SH, MIPS (архитектура), Analog Devices Blackfin, TI C2000/28x и другие. В качестве UML-редакторов интегрируются два типа диаграмм – классов (Class Diagram) и состояния оборудования (State Machine Diagram).
2. ISaGRAF — инструментальная графическая среда разработки прикладных программ для программируемых логических контроллеров на языках стандарта IEC 61131-3 и IEC 61499, позволяющая создавать локальные или распределенные системы управления. Поддерживает кодогенерацию для ISA86M, ISA68M, SCC, CC86M (те С-код).
3. YAPLC - свободная система программирования ПЛК. Для программирования используются пять языков стандарта IEC-61131-3, для расширения программ можно использовать С. В состав входят Beremiz - интегрированная среда разработки программных ПЛК на языках IEC-61131-3; matiec - транслятор языков програмрования IEC-61131-3, генерирует программный ПЛК на С; GNU ARM Embedded Toolchain - набор инструментов разработчика на С и С++.
4. LogicLab – программа, позволяющая симулировать и отображать логические условия, используя логические операции, сегментационные декодеры, бинарное кодирование и декодирование, ALU и магнитудное сравнение. Программа предназначается для проведения логических операций на основе событий, которые происходят. При помощи LogicLab пользователь может задать цепочку действий, которые нужно предпринять в зависимости от наступления тех или иных событий. При этом логические данные могут быть визуализированы в форме сегментационного отображения. Генерация машинного кода высокой эффективности для наиболее распространенных процессоров на рынке.
5. SOFTLOGIC в TRACE MODE® - это мощная система программирования промышленных контроллеров, полностью интегрированная со SCADA/HMI. Поддержка языков программирования стандарта IEC МЭК 6-1131/3. Полная версия платная.
6. SIMATIC STEP 7 — программное обеспечение фирмы Siemens для разработки систем автоматизации на основе программируемых логических контроллеров Simatic S7-300/S7-400/M7/C7 и WinAC. Поддерживаются LAD, FBD, STL, SCL, GRAPH 7, HiGraph 7, SFC. Требует платной лицензии.
7. TRACE MODE - программный комплекс класса SCADA HMI. Предназначен для разработки программного обеспечения АСУТП, систем телемеханики, автоматизации зданий, систем учёта электроэнергии (АСКУЭ, АИИС КУЭ), воды, газа, тепла, а также для обеспечения их функционирования в реальном времени. Обладает функциями программирования промышленных контроллеров. Полная версия платная. Встроена генерация документов. Шаблоны документов создаются в удобном визуальном редакторе Интегрированной среды разработки TRACE MODE. Редактор позволяет осуществлять форматирование текста, работать со списками, таблицами, вставлять растровые изображения, тренды, столбчатые и круговые гистограммы и задавать для них тип кривых, цвет фона, линии сетки, масштаб по осям Х и Y, колонтитулы, легенду и т.п.
8. MasterSCADA — программный пакет для проектирования систем диспетчерского управления и сбора данных. возможностью вывода ретроспективы на графики. Построение графиков зависимости параметра от времени, параметра от параметра. Режим просмотра графиков совмещенный с журналом событий, возможность навигации по графикам как по времени, так и по событиям.
9. KB-IDE-это открытая платы ESP32. поддерживает визуальное программирование, программирование на Arduino. Система с открытым исходным кодом. При блочном кодировании генерируется код на С/С++.
10. Arduino IDE – бесплатная IDE. Программирование осуществляется на приближенном к C/C++ языке программирования Wiring
11. IDE компании Cygnal (MCS-51), работает с языками С и ассемблер. В окне дизассемблера в пошаговом режиме отображаются команды на языке 8051 ассемблера при исполнении программы, написанной на С.
12. Atmel Studio – основанная на Visual Studio бесплатная проприетарная интегрированная среда разработки (IDE) для разработки приложений для 8- и 32-битных микроконтроллеров семейства AVR и 32-битных микроконтроллеров семейства ARM от компании Atmel. Программа позволяет работать с ассемблером, C, C++. Распространяется бесплатно
13. CodeVisionAVR – интегрированная среда разработки приложений для AVR микроконтроллеров. Работает с ассемблером и С. Полная версия платная. Содержит генератор начального кода программы, позволяющего произвести инициализацию периферийных устройств;
14. Keil MDK–ARM (Keil uVision) – среда разработки, представляющая собой набор утилит для выполнения полного комплекса мероприятий по написанию приложений для микроконтроллеров. Работает с С или ассемблером. Полная версия платная.
15. Atollic TrueSTUDIO – интегрированная среда разработки программ для ARM микроконтроллеров на базе Eclipse. Поддерживает языки C/C++ и ассемблер. Полная версия платная.
16. IAR Embedded Workbench – среда разработки приложений для целого ряда микроконтроллеров. Позволяет писать программы на языках C, C++ и ассемблере. Полная версия платная.
17. CooCox CoIDE – Высокоинтегрированная программная среда, предназначенная для разработки кода микроконтроллеров архитектуры ARM. Программа основана на базе Eclipse и имеет все ее достоинства. Данная среда разработки совершенно бесплатна и имеет открытый код.
18. Scilab - пакет прикладных математических программ, предоставляющий открытое окружение для инженерных (технических) и научных расчётов. Это самая полная общедоступная альтернатива MATLAB. Содержит интерфейс к Fortran, Tcl/Tk, C, C++, Java, LabVIEW. Распространяется бесплатно.
19. Quantium platform (qm) предназначен для моделирования диаграмм состояний UML, поддерживает кодогенерацию на C или C ++.

Таблица 3. - Cредства разработки систем управления и ПО для встраиваемых систем

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Реверсивный инжиниринг** | **Кодогенерация** | **Возможность модификации диаграмм** | **Платно для коммерческого использования** |
| MATLAB: Stateflow, Simulink | -? | + |  | + |
| CoDeSys | -? | -? |  | + |
| ISaGRAF | -? | + |  | + |
| YAPLC | -? | + |  | -? |
| AXEl Logiclab | +? | + | ? | +? |
| SOFTLOGIC TRACE MODE | - | -? |  | + |
| SIMATIC STEP 7 | - | - |  | + |
| SCADA TRACE MODE | + | - | + | + |
| MasterSCADA | + | -? | + | + |
| KB-IDE | -? | + |  | - |
| Arduino IDE | - | -? |  | - |
| IDE компании Cygnal (MCS-51) | -? | + |  | -? |
| Atmel Studio | -? | -? |  | - |
| CodeVisionAVR | -? | + \* |  | + |
| Keil MDK–ARM (Keil uVision) | -? | -? |  | + |
| Atollic TrueSTUDIO | +? | +? | + | - |
| IAR Embedded Workbench | -? | +? |  | + |
| CooCox CoIDE | +? | +? | + | - |
| Scilab | +? | + | + | - |
| Quantium platform (qm) | -? | + |  | -? |

На основе изученных программных средств можно сделать вывод о необходимости адаптировать UML-диаграммы к визуализации диаграмм процессов.

# Разработка диаграмм для отображения связи процессов

Такой подход к визуализации процессов позволит быстро оценить связи между ними по переменным и вызовам. Имеющийся код управления микроволновой печью из пособия [8] был дополнен реализацией часов, показывающих текущее время, когда микроволновая печь не используется, а также добавлена возможность задания времени. Кроме того, было изменено поведение алгоритма при открытии дверцы во время разогрева: теперь время готовки не сбрасывается, и при закрытии дверцы разогрев продолжается до конца заданного времени. Если же на протяжении 30 минут дверца не была закрыта, время разогревания сбрасывается, и система переходит в начальное состояние. Также был осуществлен переход на англоязычную версию синтаксиса языка.

Для отображения состояний процесса Разогрев использовалась диаграмма состояний UML [9][10], пример на рис. 1. Внутри вершин расположены названия состояний, соединенные стрелками переходов, над которыми указывается в квадратных скобках ограждающее выражение, т.е. условие перехода, а после косой черты указываются действия, выполняемые при переходе в данное состояние.

Для построения диаграмм связи процессов по данным предложено использовать нотацию, разработанную на основе модифицированной диаграммы деятельности и состояний UML. В вершинах прямоугольной формы отображаются названия процессов, соединенных с овальными вершинами, внутри которых указаны названия разделяемых переменных. Над стрелкой подписывается тип взаимодействия – импортирование (с ключевым словом “import”) или объявление (с ключевым словом “declare”). Пример приведен на рис. 3.

Для построения диаграмм связи процессов по управлению были модифицированы:

* диаграмма деятельности UML, откуда заимствован общий вид вершин диаграммы,
* диаграмма состояний UML, из нотации которой использованы:
  + идея подписей над стрелками;
  + обозначение точки входа.

Внутри вершин указаны названия процессов. Вершины соединены стрелками, отражающими связь по управлению, причем тип связи указан над стрелкой с помощью ключевых слов “start” и “stop”. Пример диаграмм приведен на рис. 2.



Рис. 1. Диаграмма связи процессов по данным.



Рис. 2. Диаграмма связи процессов по управлению.



Рис. 3. Диаграмма состояний для процесса Разогрев.

# Обзор средств визуализации графов

1. GraphViz [15][19][20][21][22] - программное обеспечение визуализации графов с открытом исходным кодом. Пакет состоит из набора утилит командной строки и программ с графическим интерфейсом, способных обрабатывать файлы на языке DOT, а также виджетов и библиотек, облегчающих создание графов и программ для их построения, включая следующие:

* dot — инструмент создания многоуровневого графа с возможностью вывода изображения результирующего графа во множестве форматов (PNG, PDF, PostScript, SVG и ряда других). Предлагается основной, если ребра имеют направление. Dot располагает ребра в одном направлении (сверху вниз или слева на право), пытается избежать пересечений ребер и уменьшает их длины.
* neato — инструмент создания графа на основе «пружинной» модели («spring model», «energy minimised»). Предлагается основной для небольших графов (около 100 узлов), о структуре которых ничего не известно.
* twopi — инструмент создания графа на основе «радиальной» модели. Расположение узлов на концентрических окружностях зависит от их дистанции от данного корневого узла. Положение корневого узла можно выбрать или предоставить это сделать программе.
* circo — инструмент создания графа на основе «круговой» модели.
* fdp — инструмент создания ненаправленного графа на основе «fdp»-модели.
* dotty — графический интерфейс для создания графов.
* lefty — программируемый графический виджет (на языке EZ[3][4]).
* Sfdp - мультимасштабная версия fdp для верстки больших графов.

GraphViz имеет множество опций для индивидуализации диаграмм. Среди них: выбор цветов, шрифтов, изменение стилей линий и узлов, создание гиперссылок и т.д. Имеет 55 видов выходных форматов файлов. Есть API для взаимодействия со многими языками, такими, как java и python. Форма узлов может быть различной. Dotty предоставляет возможность перемещать узлы, но при этом затираются элементы, лежащие на пути перетаскивания узла.

1. Gephi [16] - это программный пакет сетевого анализа и визуализации с открытым исходным кодом. В нем есть инструменты раскраски вершин и рёбер по их свойствам, настройка подписей, размеров и прочих параметров. Есть экспорт в svg, png, pdf. Поддерживает импорт из БД (SQLite, MySQL, PostgreSQL и Teradata). Gephi реализует 12 алгоритмов компоновки графа. Импортирует такие форматы представления графов, как GEXF, GDF, GML, GraphML, Pajek NET, GraphViz DOT, CSV, UCINET DL, Tulip TPL, Netdraw VNA, Spreadsheet. Есть API. Поддерживает только круглые узлы. Реализует возможность перетаскивать узлы.
2. Igraph [23][24] - это библиотека с открытым исходным кодом для создания и управления графами и анализа сетей. Она написана на C, а также существует в виде пакетов Python и R. Находится в свободном доступе под лицензией GNU General Public License Version 2. Метки узлов в графе могут быть любыми. Граф может быть представлен в форматах Pickle, pajek, net, ncol, graphmlz, graphml, gml, edgelist, edges, edge, dl(только чтение), dimacs, adjacency, lgl. Leda, graphviz (только запись), dot (только запись). Реализует 13 видов укладки графа. Поддерживает PostScript, PDF files, XFig files, SVG files, JPG, PNG

Реализуются следующие алгоритмы расположения графов:

* circle - распределяет вершины равномерно по кругу.
* drl - уже упомянутый алгоритм для больших графов.
* fr - силовой алгоритм Fruchterman-Reingold.
* graphort - алгоритм на основе физической модели, разработанный Michael Schmuhl.
* gfr - модификация алгоритма Fruchterman-Reingold для больших графов.
* kk - алгоритм Kamada-Kawai.
* lgl - адгоритм LGL.
* random - равномерное распределение точек.
* rt - алгоритм Reingold-Tilford.
* rt\_circular - модификация предыдущего алгоритма для размещения дерева по кругу.

1. Graphistry [26] Реализует только одну укладку. Есть ограничение 800 тысяч на максимальное число вершин или рёбер. Распространяется под лицензией Apache License 2.0. Вычисления проводятся на сервере, и стоимость 10$/час
2. OGDF [13][27][28] - это автономная библиотека C++ для графических алгоритмов, в частности для автоматического рисования графов. Библиотека доступна под лицензией GNU General Public License. Реализует hierarchical layout, Hierarchical layout with predefined layering, energy-based layout, orthogonal layout. Поддерживает xml, gml, svg форматы. Не занимается непосредственной отрисовкой графов.
3. ZGRViewer [29][30] - это графический визуализатор, реализованный на Java и основанный на масштабируемой машине визуального преобразования. Он специально предназначен для отображения графиков на dot, выполняя построение svg файла с помощью GraphViz.
4. Yed. [14] Распространяется бесплатно под лицензией yEd Software License Agreement, предполагающей следующее: «Лицензиату предоставляется неисключительное и непередаваемое право на установку одной копии программного обеспечения и использование его в качестве приложения. Программное обеспечение не может использоваться как часть автоматизированного процесса» [31]. Поддерживает следующие форматы представления графов: Graphml, ygf, gml, xgml, xls, xlsx, tgf, ged, XML + XSL, edge list, node list ,экспорт в форматы png, jpg, svg, pdf, swf, gif, bmp, emf, eps, html. Реализует более 24 видов layout (таких, как tree layout, radial layout и другие).
5. NetworkX [17][18][32] - библиотека для манипулирования графами под python, распространяется свободно под BSD-new license. Узлы могут быть любыми (например, текстом, изображениями, записями XML). Ребра могут содержать произвольные данные (например, веса, временные ряды). Поддерживаются следующие форматы представления графов: GML, GraphML, edge list, GIS Shapefile, Pajek, Sparse6, Graph6, YAML, LEDA, JSON, Pickle, GEXF, Adjacency List. Алгоритмы укладки графов предоставляются GraphViz.
6. Tulip[44] – библиотека, распространяется бесплатно под лицензией [43] GNU LESSER GENERAL PUBLIC LICENSE. Поддерживает Basic, Force Directed, Hierarchical, Misc, Multilevel, Planar, Tree layout типы. Плагины используют OGDF.
7. LEDA[48] [49] – платная библиотека для C++ и Java, содержащая также модуль GraphWin [37]
8. NetDraw[45][46] – бесплатное приложение. Программа позволяет создавать структуру с оптимальным расположением узлов, трансформировать граф, перетаскивать отдельные узлы вручную.

Более наглядное сравнение представлено ниже (см. Таблицу 4). Сохраняет в форматы jpeg, bitmap, metafile

Таблица 4. - Сравнение средств визуализации графов.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название | Платно для коммерческого использования | Возможность перетаскивания | Auto layout | Количество алгоритмов укладки графа | Формат представления графов | Количество поддерживаемых форматов | Настройка параметров визуализации | Наличие API | Узлы как на требуемых диаграммах | WYSIWYG\*\*\* | Подписи над ребрами |
| GraphViz | - | +\* | + | 11 | dot (gv) | 55 | + | + | + | - | - |
| Gephi | - | + | + | 12 | GEXF, GDF, GML, GraphML, Pajek NET, gv, CSV, UCINET DL, Tulip TPL, Netdraw VNA, Spreadsheet | 9 | + | + | - | - | + |
| Igraph | - | - | + | 13 | Pickle, pajek, net, ncol, GraphMLz, GraphML, gml, edgelist, edges, edge, dl(только чтение), dimacs, adjacency, lgl.  Leda, graphviz, dot (только запись) | 6 | + | + | + | - | + |
| Graphistry | + | + | + | 1 | ? | ? | + | + | - | + | - |
| OGDF | - | - | + | 12 | Graph, gml, rome, leda, chaco, PMDissGraph, YGraph, Graph6, MatrixMarket, Rudy, BENCH, PLA, GD-Challenge, GraphML, DOT, GEXF, GDF, TLP, DL, STP, DMF, edgeList | 3 | + | + | + | - | + |
| ZGRViewer | - | + | + | 11 | Dot (gv), svg | ? | + | + | + | - | - |
| Yed | -\*\* | + | + | 24+ | GraphML, ygf, gml, xgml, xls, xlsx, tgf, ged, XML + XSL, edge list, node list, graphmlz | 12 | + | - | + | + | + |
| NetworkX | - | +\* | + | 11 | GML, GraphML, edge list, GIS Shapefile, Pajek, Sparse6, Graph6, YAML, LEDA, JSON, Pickle, GEXF, Adjacency List, gv | 8 | + | + | + | - | + |
| Tulip | - | ? | + | 7 | Tlp, gml, csv, gexf | ? | + | + | + | ? | + |
| LEDA | + | ? | + | ? | Gw, GML, LEDA | ? | ? | + | + | ? | + |
| NetDraw | - | + | + | 3+ | Nodelist, edgelist, fullmatrix, vna, Pajek, vna, uciNet. | 3 | + | ? | + | + | - |

\* Примечание. Только в dotty

\*\* Примечание. yEd Software License Agreement

\*\*\* Примечание. What you see is what you get.

# Обзор форматов представления графов.

Список наиболее распространенных форматов хранения графов:

* GraphML [33][34][35]
* Gv [36] [37]
* Xgml\* [38]
* Gml
* Node list
* Edge list
* Pajek[40]
* Leda[41]
* TLP [42]
* Gw
* GEXF [50][51]

Краткий обзор наиболее распространенных форматов представления графов приведен в таблице 5. Схема возможных преобразований форматов представлены на диаграмме (см. Рис 4.).

Таблица 5. Обзор форматов представления графов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Хранение координат | Форма вершин | Ориентированный граф | Подписи над ребрами | Поддерживающие средства |
| GraphML | + | + | + | + | Gephi, Igraph, OGDF, Yed, NetworkX |
| Gv | - | + | + | + | GraphViz, Gephi, Igraph\*, OGDF, ZGRViewer, NetworkX |
| Xgml\* | + | + | + | + | Yed |
| Gml | + | + | + | + | Gephi, Igraph, OGDF, Yed, NetworkX, Tulip, LEDA |
| Node list | - | - | + | - | Yed, NetDraw |
| Edge list | - | - | + | - | OGDF, Yed, NetworkX, NetDraw |
| Pajek | + | + | + | + | Gephi, Igraph, NetworkX, NetDraw |
| Leda | + | + | + | + | Igraph, OGDF, NetworkX, LEDA |
| TLP | + | + | + | + | OGDF, Gephi, Tulip |
| Gw | + | + | + | + | LEDA |
| GEXF | + | + | + | - | Gephi, OGDF, NetworkX, Tulip |

\* Примечание. XGML is an XML-ish variant of the GML file format where sections and attributes as listed in the various tables of section GML File Format are wrapped in <section> and <attribute> tags, respectively.



Рис. 4. Диаграмма возможных преобразований форматов представления графов.

В результате анализа были выделены два языка описания графов: GML и GraphML. Они оказались наиболее поддерживаемыми форматами, удовлетворяющими поставленным критериям. Также при отборе учитывалось количество форматов, в которые можно перевести рассматриваемый формат.

Для окончательного выбора одного из форматов рассмотрим более подробную сравнительную таблицу (см. Таблицу 6).

Таблица 6. Сравнение GML и GraphML.

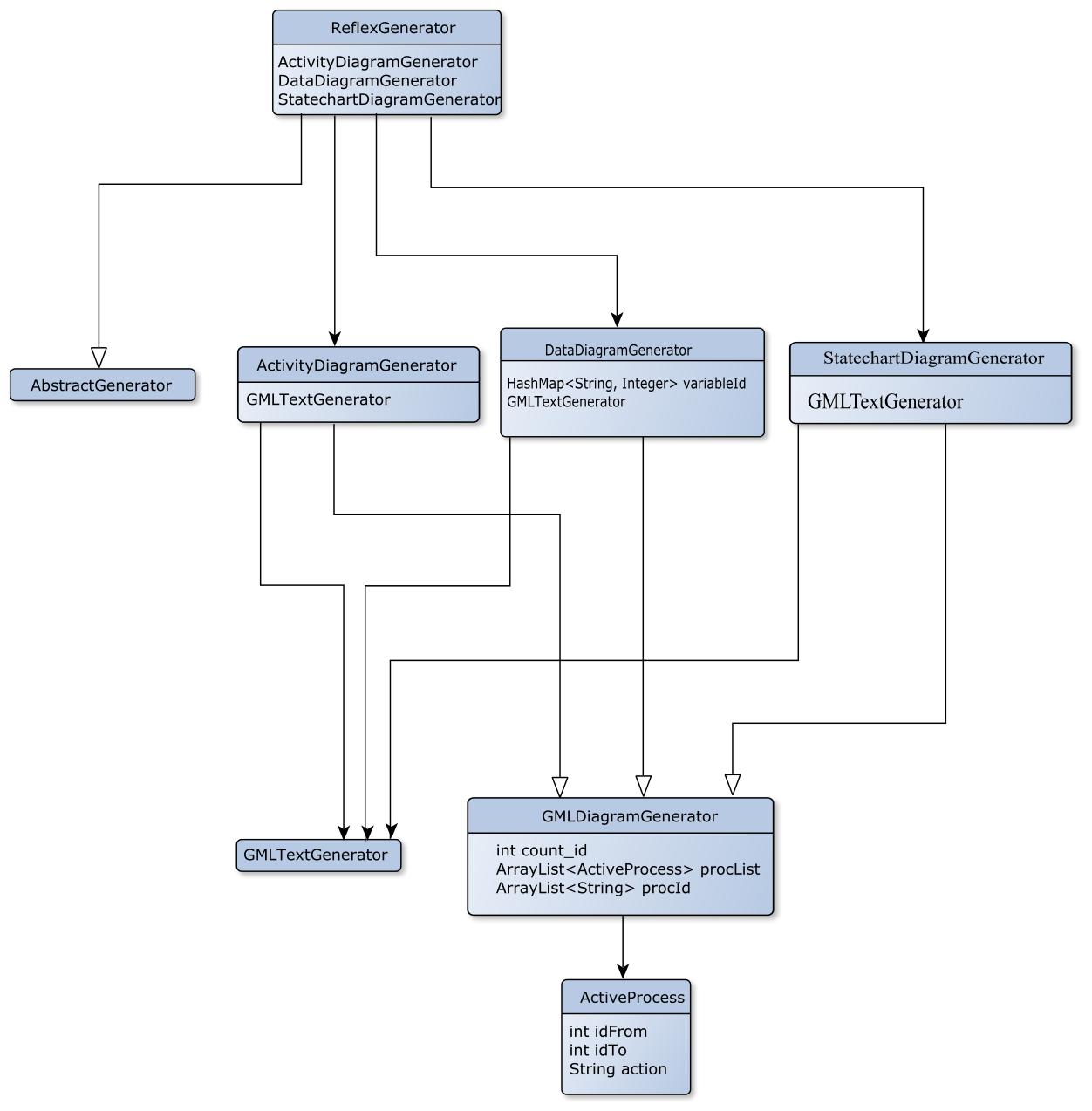
|  |  |  |
| --- | --- | --- |
| Критерий сравнения | GML | GraphML |
| Диаграмма связи по данным | + | + |
| Диаграмма связи по управлению | + | + |
| Диаграмма состояний процесса | + | + |
| Размер файла | x | 2x |
| Легок для восприятия человеком | + | - |
| Поддержка русского языка в надписях | +\* | + |

\* Примечание. Необходимо записывать ASCII-коды символов в 16-ричном виде.

# Архитектура модуля

На рис. 5 представлена диаграмма классов модуля визуализации диаграмм процессов. Основной класс ReflexGenerator, наследующийся от класса AbstractGenerator, использует три разных генератора для соответствующих типов диаграмм, использующих GMLTextGenerator, отвечающий непосредственно за генерацию кода в формате GML. В процессе генерации происходит проход по абстрактному синтаксическому дереву и создание модели диаграммы в виде списка вершин и связи между ними, элементы которого представлены классом ActiveProcess.

Рис. 5. Диаграмма классов модуля визуализации диаграмм процессов.

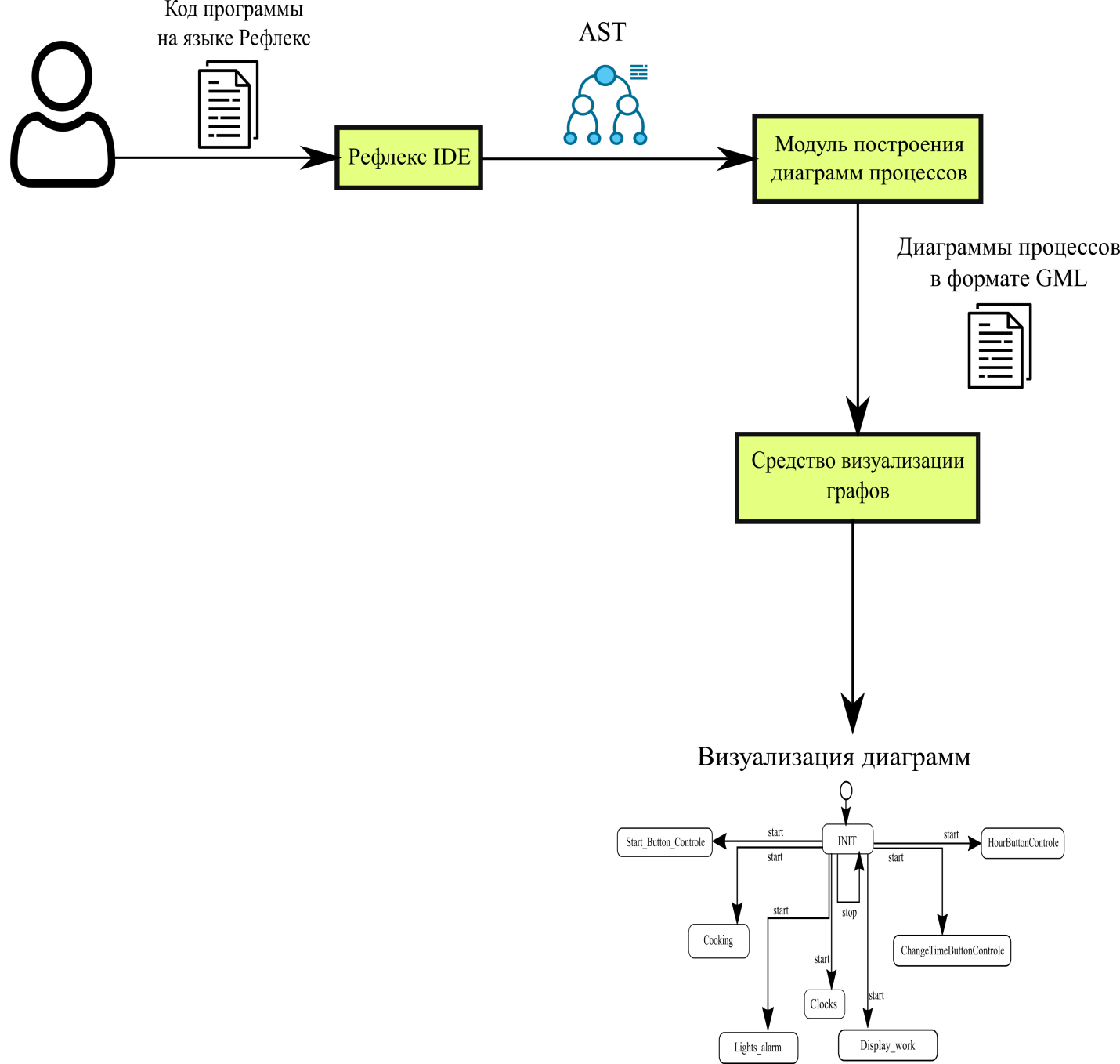


# Реализация

Реализацию модуля решено выполнить в виде Eclipse-плагина с использованием технологий xtext и xtend для написания генератора.

Схема работы представлена на рис. 6. После открытия Рефлекс IDE и написания кода, можно нажать кнопку сохранения файла, после чего автоматически сгенерируются диаграммы связи процессов по данным, по управлению и диаграммы состояний для каждого процесса, которые можно увидеть в левой части IDE в дереве файлов. Открывать их рекомендуется средством визуализации графов yed, поддерживающим автоматическую укладку графа на плоскость, а также возможность редактирования диаграмм.

Рис. 6. Схема работы с модулем визуализации диаграмм процессов.



# Список используемой литературы

* + 1. Розов А. С., В. Е. Зюбин, Нефедов Д. В. Программирование встраиваемых микроконтроллерных систем на основе гиперпроцессов//Вестн. НГУ. Серия: Информационные технологии. 2017. Т. 15, №4. С. 64-73.
    2. В. Е. Зюбин «Си с процессами»: язык программирования логических контроллеров //Мехатроника. 2006. № 12. C. 31–35
    3. В. Е. Зюбин Язык «Рефлекс» – диалект Си для программируемых логических контроллеров // Шестая международная научно-практическая конференция "Средства и системы автоматизации" CSAF’06 // Томск, 1-3 ноября 2005 г. Томск: ТУСУР, 2005
    4. TOP IDE index, URL: <http://pypl.github.io/IDE.html>
    5. Стойчев С. В. Интегрированные среды разработки программного обеспечения для микроконтроллеров. URL: <http://masters.donntu.org/2014/etf/stoychev/ind/index.htm>
    6. AXEL S.R.L. URL:<https://www.axelsoftware.it/en/>
    7. David Harel, “Statecharts: a visual formalism for complex systems”, 1986.
    8. Процесс-ориентированное программирование [Текст] : Учебное пособие / В. Е. Зюбин ; Новосиб. гос. университет, Институт автоматики и электрометрии СО РАН. - 2-е изд. / под ред. Е. В. Дубовцева. - Новосибирск: Ред. - издат. центр НГУ, 2011. - 194 с. - УДК 811.93+62-529+004.434. - тираж 100 экз.
    9. Википедия, Диаграмма состояний (UML), URL: https://ru.wikipedia.org/wiki/Диаграмма\_состояний\_(UML)
    10. Википедия, UML statemachine diagram, URL:<https://en.wikipedia.org/wiki/UML_state_machine>
    11. Хабр, «Визуализация больших графов для самых маленьких», URL:<https://habr.com/ru/company/ods/blog/464715/>
    12. Helen Gibson, Joe Faith and Paul Vickers, “A survey of two-dimensional graph layout techniques for information visualization”, URL: <http://leonidzhukov.net/hse/2018/sna/papers/gibson2013>
    13. M. Chimani, C. Gutwenger, M. Jünger, G. W. Klau, K. Klein, P. Mutzel. The Open Graph Drawing Framework (OGDF). Chapter 17 in: R. Tamassia (ed.), Handbook of Graph Drawing and Visualization, CRC Press, 2014.
    14. YWORKS, Yed, URL: <https://www.yworks.com/products/yed>
    15. Graphviz - Graph Visualization Software, URL:<http://www.graphviz.org>
    16. Gephi: The Open Graph Viz Platform, URL: <https://gephi.org>
    17. NetworkX [заметки о визуализации графов на python], URL: <http://blog.esemi.ru/2011/12/networkx-python.html>
    18. NetworkX, URL:<https://networkx.github.io>
    19. Хабр, Использование Graphviz для построения блок-схем, URL:<https://habr.com/ru/post/337078/>
    20. LinuxShare: Введение в GraphViz, URL: <http://linuxshare.ru/docs/misc/graphviz.html>
    21. Информационно-образовательный сайт для студентов, аспирантов и коллег Тушавина В. А., “Graphviz – рисуем графы”, URL: <https://tushavin.ru/graphviz/>
    22. GraphWiz: Node, Edge and Graph Attributes, URL: <http://www.graphviz.org/doc/info/attrs.html#d:ratio>
    23. Pyton-igraph Manual, URL: <https://igraph.org/python/doc/tutorial/tutorial.html>
    24. Package ‘igraph’, URL: <https://igraph.org/r/doc/igraph.pdf>
    25. run R online, URL:<https://rdrr.io/snippets/> - run R online
    26. Get Started Graphistry, URL: <https://www.graphistry.com/get-started>
    27. OGDF, URL:<https://ogdf.uos.de>
    28. OGDF repository, URL:<https://github.com/ogdf/ogdf>
    29. ZGRViewer, a GraphViz/DOT Viewer, URL: <http://zvtm.sourceforge.net/zgrviewer.html>
    30. E. Pietriga, A Toolkit for Addressing HCI Issues in Visual Language Environments, IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), pages 145-152, September 2005, Dallas, TX, USA
    31. yEd Software License Agreement, URL: <https://www.yworks.com/resources/yed/license.html>
    32. Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, “Exploring network structure, dynamics, and function using NetworkX”, in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008
    33. The GraphML File Format, URL:<http://graphml.graphdrawing.org>
    34. GraphML - язык описания графов, [Источник URL: <http://graphml.graphdrawing.org/>, Перевод: Шокоров В. П.], URL: <http://citforum.ru/internet/xml/graphml/>
    35. В.Н. Касьянов, ЯЗЫК ПРЕДСТАВЛЕНИЯ ГРАФОВ GRAPHML: БАЗОВЫЕ СРЕДСТВА
    36. Википедия, язык DOT, URL: <https://ru.wikipedia.org/wiki/DOT_(язык)>
    37. The DOT Language, URL: <https://graphviz.gitlab.io/_pages/doc/info/lang.html>
    38. XGML, Chapter 9. Input and Output, URL: <https://docs.yworks.com/yfiles/doc/developers-guide/xgml.html>
    39. GML: Chapter 9. Input and Output, URL: <https://docs.yworks.com/yfiles/doc/developers-guide/gml.html>
    40. Pajek: Exporting pictures from Pajek to EPS, Vladimir Batagelj, Andrej Mrvar, URL: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/doc/draweps.htm>
    41. Native File Format for Graphs, URL: <http://www.algorithmic-solutions.info/leda_guide/graphs/leda_native_graph_fileformat.html>
    42. Tulip software graph format (TLP), URL: <https://tulip.labri.fr/TulipDrupal/?q=tlp-file-format>
    43. LGPL: License, URL: <https://tulip.labri.fr/TulipDrupal/?q=licence>
    44. Tulip Screenshots, URL: <https://tulip.labri.fr/TulipDrupal/?q=screenshots>
    45. Компьютерная визуализация социальных сетей. Александр Прохоров, Николай Ларичев, URL:<https://compress.ru/article.aspx?id=16593#NetDraw>
    46. Borgatti, S.P., 2002. NetDraw Software for Network Visualization. Analytic Technologies: Lexington, KY
    47. GraphWin, URL: <https://graphics.stanford.edu/courses/cs368spring/TA/manuals/LEDA/GraphWin.html>
    48. LEDA Graphs for Java, URL: <https://algorithmic-solutions.com/index.php/products/leda-graphs-for-java>
    49. LEDA for C++, URL:<https://algorithmic-solutions.com/index.php/products/leda-for-c>
    50. GEXF File Format, URL: <https://gephi.org/gexf/format/schema.html>
    51. Gephi draft example, URL: <https://gephi.org/gexf/1.2draft/gexf-12draft-primer.pdf>