САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Специальность 010400 «Информационные технологии»

Кафедра системного программирования

Интеграция средств раскладок KIELER в инфраструктуру GMF

Выпускная работа бакалавра

Скородумов Кирилл Владимирович

461 группа

Руководитель	……	Д.В.Кознов
Рецензент	……	А.В.Сорокин
«Допущен к защите»	……	д.ф.-м. н., проф.
Заведующий кафедрой		А.Н.Терехов

Санкт-Петербург

2012

Saint-Petersburg State University

Mathematics and Mechanics Faculty

Software Engineering Department

Integration of KIELER layout tools into GMF

Graduate paper by

Kirill Skorodumov

461 group

Scientific advisor	……	D.V.Koznov
Reviewer	……	
"Approved by"	……	Professor
Head of Department		A.N.Terekhov

Saint-Petersburg

2012

Оглавление

Введение

Визуальное моделирование программного обеспечения – это использование чертежей и диаграмм при проектировании и сопровождении программного обеспечения по аналогии с чертёжным проектированием в строительстве и машиностроении [10]. Наиболее распространённым и развитым средством визуального моделирования является UML [10] – язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но в средствах выполнения UML-моделей как интерпретируемого кода возможна кодогенерация.

Более гибким средством являются предметно-ориентированные языки (Domain-Specific Languages – DSL). Эти языки разрабатываются для определенной предметной области (как правило, группа сходных проектов в рамках одной компании) и позволяют упростить и удешевить процесс разработки ПО.

В настоящее время существуют различные средства, позволяющие интегрировать инструменты визуального моделирования в среду разработки ПО: GMF[7], Microsoft DSL Tools[9], Microsoft Visio[11]. Основным требованием к подобным средствам является удобство использования и красота получаемых диаграмм.

GMF (Graphical Modelling Framework) – технология, предоставляющее компоненты и инфраструктуру для создания графические редакторов в среде Eclipse [8]. Эта технология позволяет задать метамодель будущего DSL, определить как графически будут задаваться его элементы, и создать инструменты для редактирования диаграмм. Однако данная технология не

поддерживает разработку так называемых навигационных сервисов [12], позволяющих просматривать и изучать модели, отображая/скрывая ее детали на диаграммах без изменения самой модели.

Технология KIELER (Kiel Integrated Environment for Layout Eclipse Rich Client) направлена на создание инструментов, позволяющих разрабатывать визуальные модели для сложных систем. Основная идея состоит в применении алгоритмов автоматической раскладки ко всем графическим компонентам диаграмм в пределах среды разработки, что открывает как новые возможности для создания и редактирования диаграмм. Технология является надстройкой над GMF, имеет фиксированную метамодель, и, следовательно, поддерживает набор диаграмм. Для поддержки новых типов диаграмм потребуется вручную написать код, работающий с ними. К тому же применение раскладок не решает проблему читаемости больших диаграмм.

В настоящее время на кафедре системного программирования мат.-мех. факультета СПбГУ развивается проект под названием V2V-трансформации, предлагающий решение для данной проблемы при помощи реализации навигационных сервисов в редакторах GMF [12]. Его подход заключается в интерпретации навигационных сервисов как трансформаций над диаграммой. Разработчику графических редакторов, пользующемуся GMF, предлагаются дополнительные возможности по проектированию таких сервисов.

1.Постановка задачи

При трансформации диаграммы удобно иметь возможность оптимально разложить полученный результат. KIELER предоставляет фреймворк для работы с раскладками, однако его раскладки работают с классами метамодели проекта KIELER, что не позволяет раскладывать диаграммы произвольного редактора GMF. В связи с этим в рамках данной работы были поставлены следующие задачи.

- Изучить инфраструктуру проекта KIELER.
- Спроектировать и реализовать библиотеку, позволяющую средствами KIELER раскладывать диаграмму, созданную с помощью любого редактора GMF.
- Интегрировать её в готовый GMF редактор.

2.Обзор используемых технологий

2.1 GMF

GMF предоставляет множество средств, которые облегчают использование GEF (Graphical Editing Framework) и EMF (Eclipse Modeling Framework) – стандартных библиотек Eclipse Modeling Project для создания графических редакторов и моделирования соотвественно[1]. Оно позволяет генерировать графические редакторы по метамодели предметно-ориентированного языка. При этом производится создание сущностей доменной модели, графических фигур и инструментов редактора, затем создаётся отображение между ними и происходит генерация редактора.

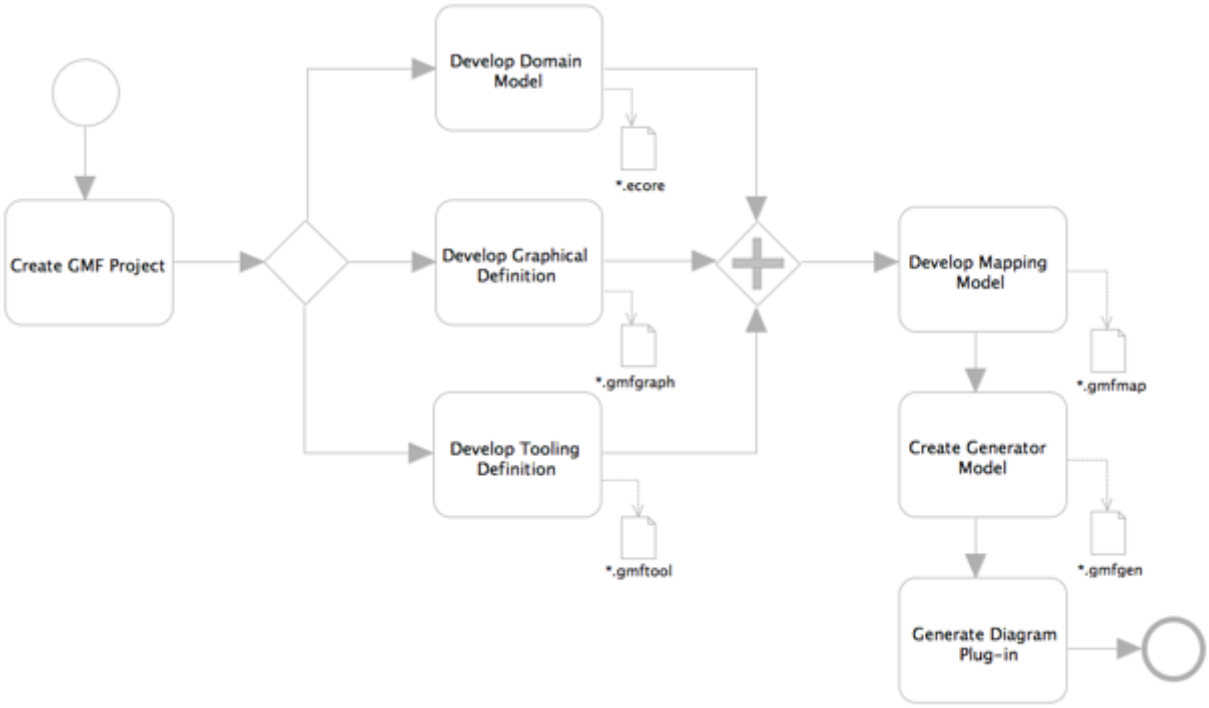


Рис. 1

Всё начинается с доменной модели – метамодели, определяемой в объектно-ориентированных терминах, таких как классы с атрибутами и

[1] Eclipse Modeling Project – платформа для разработки DSM решений для Eclipse [15].

ссылками на другие классы. Код для работы с этими классами генерируется прямо по доменной модели.

Графическая модель определяет символы, используемые для изображения различных сущностей, а также метки для отображения и редактирования значений атрибутов. Классы и ссылки на другие классы могут отображаться в соединения между узлами, соответствующими классам, или, например, ссылки могут интерпретироваться как агрегирование и позволять выделять составляющие из классов на диаграмме.

Модель инструментов редактора определяет сервисы для создания новых узлов и соединений между ними.

Модель отображения соединяет все три вышеописанные модели и определяет для каждой сущности доменной модели то, как она будет отображаться и какими средствами создаваться.

GMF редакторы используют для хранения своих диаграмм Ecore и EcoreDiag форматы. Ecore предназначается для хранения доменных моделей, а EcoreDiag – для хранения моделей представления. Формат EcoreDiag является расширением формата Ecore и содержит в себе также информацию о видимых свойствах элемента диаграммы, таких как толщина линий, шрифты и другие.

Все редакторы GMF компилируются в виде Eclipse плагинов.

2.2 KIELER

2.2.1 Обзор

Проект KIELER является по сути набором GMF редакторов, имеющих различную функциональность: от построения UML диаграмм, таких как диаграммы последовательностей или состояний, до создания программ для системы Lego Mindstorms. Все проекты KIELER реализованы в виде ECLIPSE-плагинов и имеют открытый исходный код.

Проект KIELER состоит из нескольких проектов, которые можно разбить на три группы как изображено на рисунке 2.

1. Семантика – KIEM, KlePto, KLOTS.
2. Синтаксис – ThinKCharts, KEG, KAOM, KITS, KIES.
3. Прагматика – KIML, KLoDD, KLay, KIVi, KSBasE, KEV, KIViK, KEX, KViD.

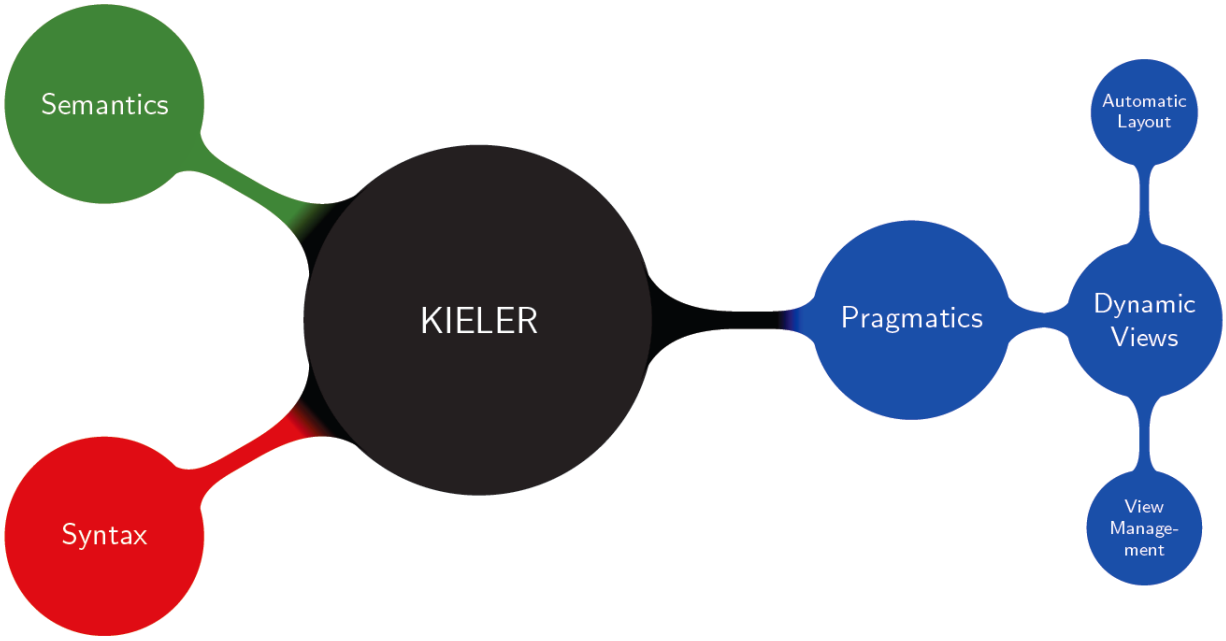


Рис. 2

Рассмотрим эти проекты более подробно.

2.2.2 Состав KIELER

KIEM (KIELER Execution Manager) – это проект, реализующий интерфейс проекта KIELER для моделирования и выполнения представлений моделей предметной области (например, EMF моделей). Сам по себе этот проект не производит никакого моделирования, но соединяет компоненты моделирования, визуализации и пользовательский интерфейс внутри платформы KIELER Eclipse.

KlePto вводит понятия моделирования и семантики в графические средства моделирования платформы Eclipse, такие как EMF, GEF, GMF.

Проект KLOTS (KIELER Lego On-line Testing System) предоставляет интегрированную среду разработки (IDE) для создания программ для систем Lego Mindstorms NXT на языке Java. Основная цель KLOTS – выполнение и тестирование программ, написанных на диалекте Synchronous Java, а также программ, разработанных на обычном языке java.

Проект ThinKCharts (Thin KIELER SyncCharts Editor) – это окружение, основанное на платформе Eclipse, созданное для моделирования SyncCharts диаграмм с дополнительными возможностями, такими как редактирование структуры, автоматическая раскладка и динамическое моделирование и визуализация.

Проект KEG (KIELER Editor for Graphs) – это графический редактор, созданный на основе платформы Eclipse для разработки графических алгоритмов, особенно алгоритмов раскладок. Для выполнение своей цели KEG предоставляет большинство структур, применяемых в теории Графов, и инструменты для построения и верификации графов (например, узлы, направленные и ненаправленные ребра и т д).

Проект KAOM (KIELER Actor Oriented Modeling) направлен на связывание и адаптацию идей проекта KIELER actor oriented modeling languages, таких как Ptolemy, который является самым ярким примером для проекта KAOM. В основном, KAOM нужен для адаптации требований различных языков и расширений под платформу Eclipse, что предоставляет намного больше возможностей для расширений, чем использование обычных приложений Java.

Проект KITS (KIELER Textual SyncCharts) содержит текстовый язык для SyncCharts, разработанный в рамках проекта KIELER и редактор для этого языка. В данный момент редактор еще не поддерживает все возможности языка и находится в процессе разработки.

Проект KLoDD (KIELER Layout of Dataflow Diagrams) направлен на разработку алгоритмов раскладки диаграмм для языков графических потоков данных.

Проект KLay (KIELER Layouters) нацелен на реализацию различных алгоритмов раскладок на языку Java для проекта KIML. В данный момент существует три реализации:

- Klay Layered – реализация раскладки, основанной на иерархиях;
- KLay Force – реализация, основанная на физической модели;
- KLay Planar – implementation of the planarization-based "topology-shape-metrics" approach.

Проект KIVi (KIELER View Management) – это движок для динамических визуализаций диаграмм (например, для графического моделирования). Основная функция проекта – это возможность проведения автоматической раскладки диаграмм, как это предусмотрено в проекте KIML. В KIVi реализовано интерактивное взаимодействие с пользователем, основанное на автоматическом применении раскладок. Рассмотрим ряд возможностей этого проекта.

1. Функции для фокусировки на элементе и получении контекста позволяют осуществлять навигацию в сложных моделях.
2. Выполнение и моделирование результатов используются для интерактивной визуальной отладки диаграмм.
3. Выполнение редактирования исполняемой трансформации моделей on the domain model while view management updates the diagram view

Проект KSBasE (KIELER Structure Based Editing) позволяет разработчикам добавлять новые структурные элементы в редактор, основанный на EMF. Отметим, что добавленные элементы должны быть основаны на метамодели редактора EMF.

Проект KEV (KIEL Environment Visualization) реализует графическую репрезентацию окружения моделирования, например, механические части. Любое SVG изображение может быть использовано как основа для анимации, в которой можно управлять цветом, прозрачностью и т д. во время моделирования. Также разработаны интерфейсы для различных

инструментов моделирования (Matlab, SCADE, Esterel Studio) и любых языков программирования, поддерживающих взаимодействие через TCP/IP. В рамках этого проекта Java приложение ModelGui было перенесено на платформу Eclipse и встроено в проект KIELER.

Проект KIViK (KIELER Visual Komparison) – реализует метод для сравнения моделей в графическом режиме в среде Eclipse. Также существует плагин EMF Compare, позволяющий сравнивать модели в текстовом режиме, схожий с методом diff текстовых редакторов. В данный момент KIViK не поддерживается и не является частью последних версий проекта KIELER. Исходные коды доступны только для Eclipse версии 3.3

Проект KEX (KIELER Example Management) – позволяет добавлять “примеры” в рабочую среду KIELER. Примерами обычно являются модели, шаблоны для SyncCharts, KIELER Actor Oriented Modeling и т д.

Проект KViD (KIELER Visualization of Data) нацелен на разработку новых механизмов отображения процессов моделирования и измерения данных модели. KViD позволяет отображать содержимое пула данных в таблице. Однако, поскольку для некоторых диаграмм было бы приемлемым прямое отображение данных (например, для диаграмм потоков данных), данный проект позволяет интегрировать новые методы отображения и соотносить их с графическими объектами.

Проект KIML (KIELER Infrastructure for Meta Layout) был создан для работы с автоматическими раскладками графических моделей. Термин Meta Layout в расшифровке аббревиатуры обращает внимание на идею описания раскладки диаграмм на абстрактном уровне в то время как конкретные раскладки рассчитываются специальными алгоритмами.

2.2.3 Раскладки

Проект KIML содержит основную функциональность, ответственную за раскладку диаграмм. В нём есть базовый класс для реализации обёрток над сторонними библиотеками раскладок, который позволяет абстрагироваться

от подробностей реализации конкретной библиотеки и использовать общий интерфейс для работы с алгоритмами этой библиотеки.

В проекте KIELER реализованы подобные «обёртки» для таких библиотек, как GraphViz, OGDF и ZEST.

KIML определяет несколько типов раскладок.

- Раскладка слоями – направляет максимальное количество рёбер в одну сторону, а узлы раскладывает слоями.
- Ортогональная раскладка – пытается разложить граф так, чтобы его максимальное количество его рёбер пересекалось под прямыми углами.
- Круговая раскладка – выделяет сильно связные компоненты графа, группируя их и располагая в форме круга.
- Силовая раскладка – имеет физический смысл и рассматривает узлы графа как притягивающиеся и отталкивающиеся объекты.
- Раскладка дерева – используется для раскладки графов, являющихся деревьями.

Также KIML реализует функциональность, позволяющую указывать настройки раскладки и раскладывать только указанную часть графа. Таким образом механизм выполнения раскладок проекта KIELER является достаточно мощным и удобным инструментом, а также может быть выделен из остальной функциональности. Этим и обосновано его применение в данной работе.

2.3 Проект V2V

Проект V2V расширяет возможности GMF, предоставляя возможности для разработки навигационных сервисов. Основой данного проекта является разделение модели и диаграммы – конкретного визуального представления модели. Навигационные сервисы реализуются как трансформации диаграммы, оставляя модель неизменной. Для спецификации трансформаций в проекте V2V используется язык ATL (Atlas Transformation Language) [11]. Этот язык разработан в рамках проекта Eclipse Modeling Project и имеет интерпретатор, позволяющий выполнять трансформации.

Структура V2V проекта изображена на рисунке 3. К структуре редактора GMF добавляются компоненты для создания и выполнения навигационных сервисов: настройка интерфейса и спецификация самих трансформациий, реализующих эти сервисы. И информация для библиотеки раскладки.

В ходе создание V2V-трансформации есть возможность указать алгоритм раскладки для итоговой диаграммы. Механизм раскладки диаграмм реализован в данной работе с помощью проекта KIELER, поскольку последний предоставляет фреймворк, позволяющий работать с различными библиотеками раскладок, а также имеет открытый исходный код.

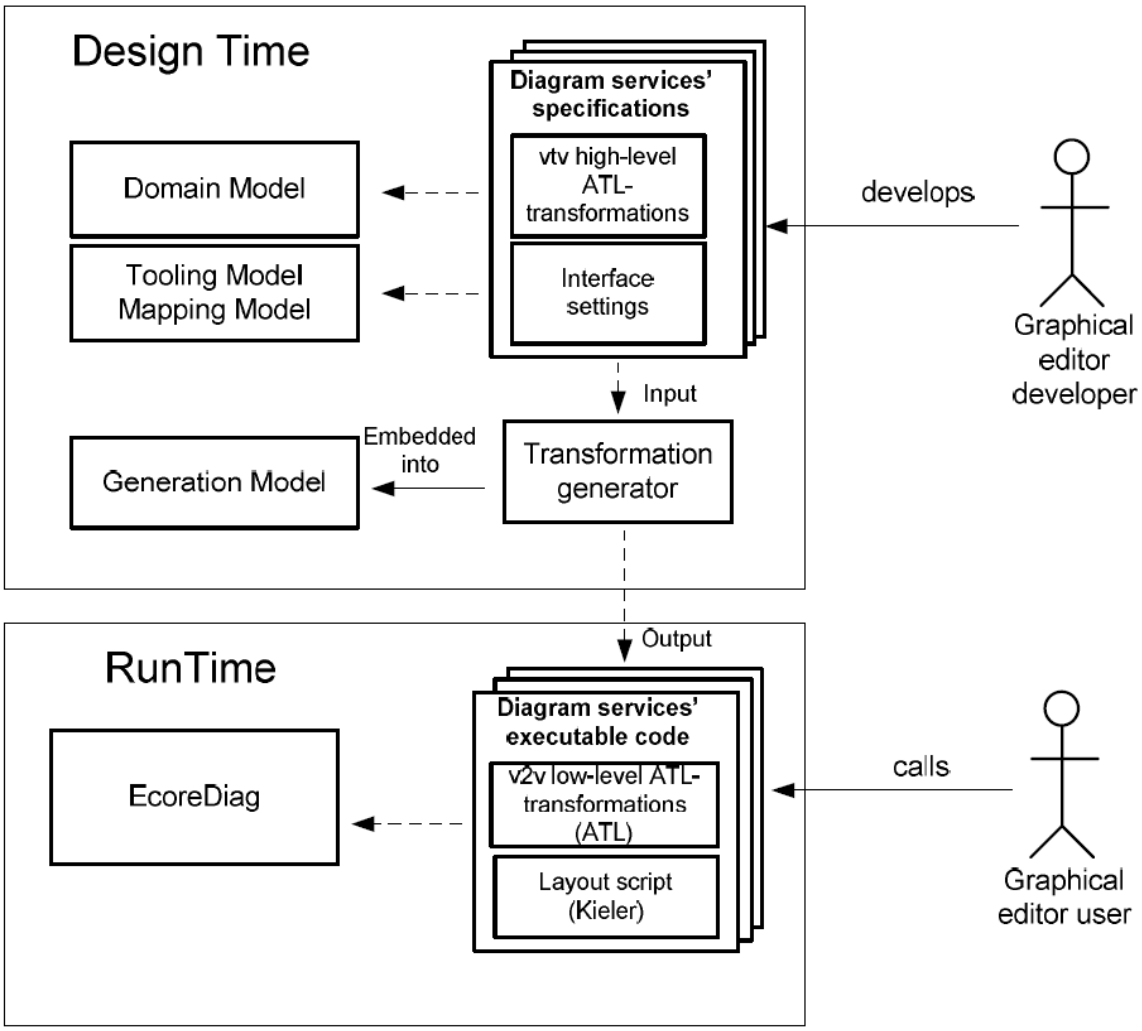

Design Time
Diagram services' specifications
vtv high-level ATL-transformations
Interface settings
Domain Model
Tooling Model Mapping Model
develops
Graphical editor developer
Input
Embedded into
Generation Model
Transformation generator
Output
RunTime
Diagram services' executable code
v2v low-level ATL-transformations (ATL)
Layout script (Kieler)
EcoreDiag
calls
Graphical editor user

Рис. 3

3 Реализация

При выполнении поставленной задачи потребовалось выделить необходимые для выполнения раскладок компоненты KIELER'a, а также реализовать два основных класса: адаптер, занимающийся конвертацией GMF сущностей в сущности KIELER, и менеджер раскладок, выполняющий запуск самого процесса раскладки.

На рисунке 4 изображена диаграмма классов созданной реализации.

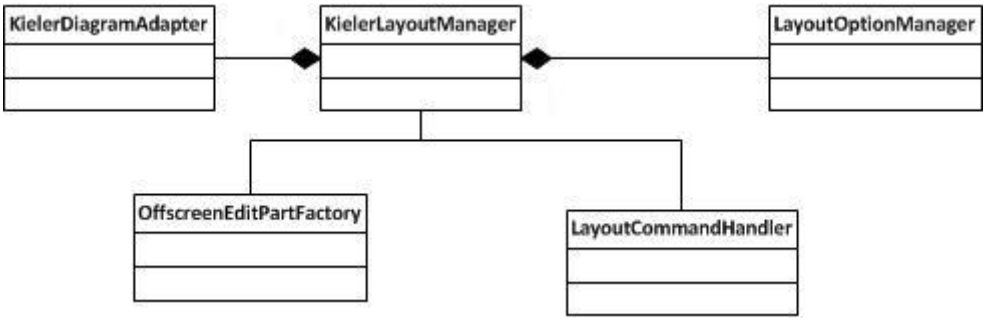


Рис. 4

3.1 Перенесённые компоненты

В проект V2V из проекта KIELER были перенесены следующие компоненты:

- de.cau.cs.kieler.core – содержит базовые классы, необходимые для работы инфраструктуры KIELER;
- de.cau.cs.kieler.core.kgraph – этот проект содержит элементы для работы с графовой структурой KGraph, в которую диаграммы преобразовываются при раскладке, а также фабричные методы для их создания.
- de.cau.cs.kieler.kiml – этот проект содержит класс AbstractLayoutProvider, базовый для всех раскладчиков, классы определяющие алгоритм раскладки, классы позволяющие определить

настройки раскладки, а также классы хранящие специфическую информацию времени раскладки;

- de.cau.cs.kieler.kwebs – этот проект содержит диалоги, позволяющие выбирать нужный алгоритм раскладки.

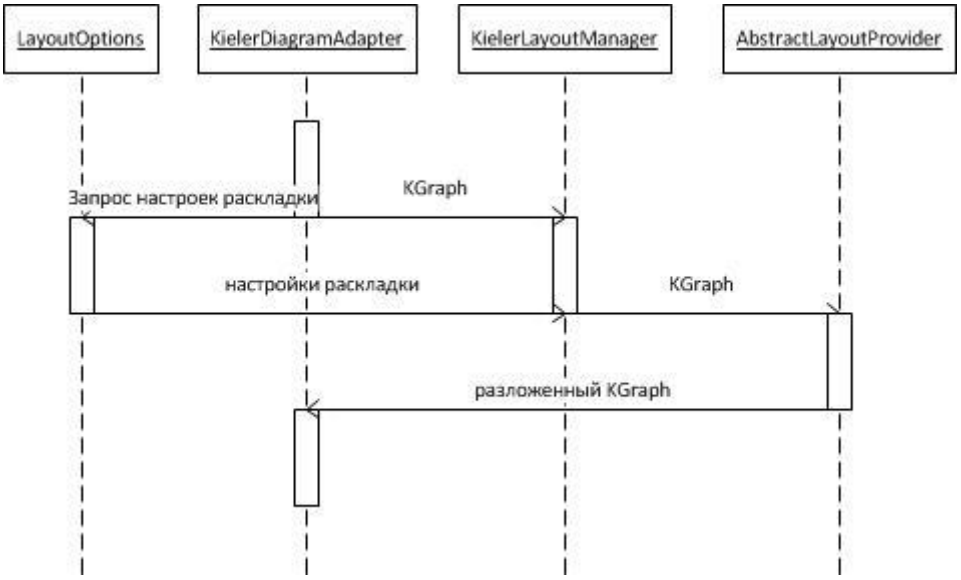
3.1.1 Адаптер

Для того, чтобы преобразовывать сущности KIELER'а в сущности GMF и обратно, был реализован класс KielerDiagramAdapter. GMF использует для представления элементов на диаграмме базовый класс Eobject. KIELER же имеет своё внутренне представление, используемое при раскладке – KGRaph. Адаптер создаёт отображение, определяющее соответствие между элементами этих двух представлений. Граф, соответствующий диаграмме, далее раскладывается классом из KIELER - AbstractLayoutProvider. Этот класс – абстрактный и конкретная его реализация указывается в настройках. После выполнения раскладки имеющееся отображение используется для восстановления исходной диаграммы EObject. Далее разложенная диаграмма сохраняется в файл. После чего она автоматически перерисовывается средствами GMF.

3.1.2 Менеджер раскладок

Выполнением раскладок занимается реализованный мною класс KielerLayoutManager. Этот класс является обёрткой над классом AbstractLayoutProvider. Он получает управление при вызове команды из пользовательского интерфейса, запускает адаптер, получает от него структуру KGRaph и запускает процесс раскладки в KIELER с заданным пользователем набором настроек, для хранения которых используется класс проекта KIELER LayoutOptions.

Процесс работы всей системы изображён на рисунке 6.



3.2 Особенности реализации

3.2.1 Сильная связность KIELER

Компоненты проекта KIELER обладают большим количеством взаимосвязей, необходимость которых зачастую неочевидна. При попытке переиспользования различных частей проекта KIELER приходится копировать в свой проект много проектов, не имеющих прямого отношения к выполняемой задаче. В частности, проект de.cau.cs.kieler.kwebs был включен в реализацию полностью, несмотря на то, что нужные формы с диалогами выбора алгоритмов содержит его единственный пакет de.cau.cs.kieler.kwebs.servicedata

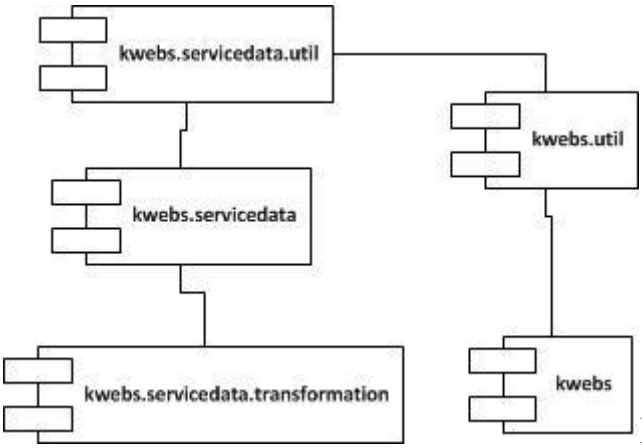


Рис. 5

3.2.1 Избыток функциональности используемых классов проекта KIELER

Классы проекта KIELER предназначены для работы в контексте KIELER и имеют зачастую ненужную для наших целей функциональность. В частности, такой функциональностью являются ProgressMonitor, необходимые для поддержки отображения прогресса выполнения трансформаций и анимация выполнения раскладки. Класс KielerProgressMonitor не реализует стандартный интерфейс Eclipse IProgressMonitor, поэтому его использование приводит к появлению лишней зависимости от проекта de.cau.cs.kieler.cs.core.alg.

3.2.2 Инфраструктура команд Eclipse

Взаимодействие с пользовательским интерфейсом происходит через фреймворк команд Eclipse, что обеспечивает расширяемост, инкапсуляцию от интерфейса и реализации ряда полезной функциональности, такой как отмена/повтор раскладки.

3.3 Апробация

Описанная выше реализация была выполнена в сгенерированном GMF-редакторе, который позволяет создавать диаграммы из простых геометрических фигур и соединяющих их линий. Она не зависит от конкретного редактора и может быть подключена к любому другому в виде Eclipse-плагина. Она позволяет выполнять раскладку создаваемых диаграмм как показано на рисунках 7 и 8.

В дальнейшем планируется интегрировать её в V2V проект и реализовать встраивание в процесс создания GMF-редакторов.

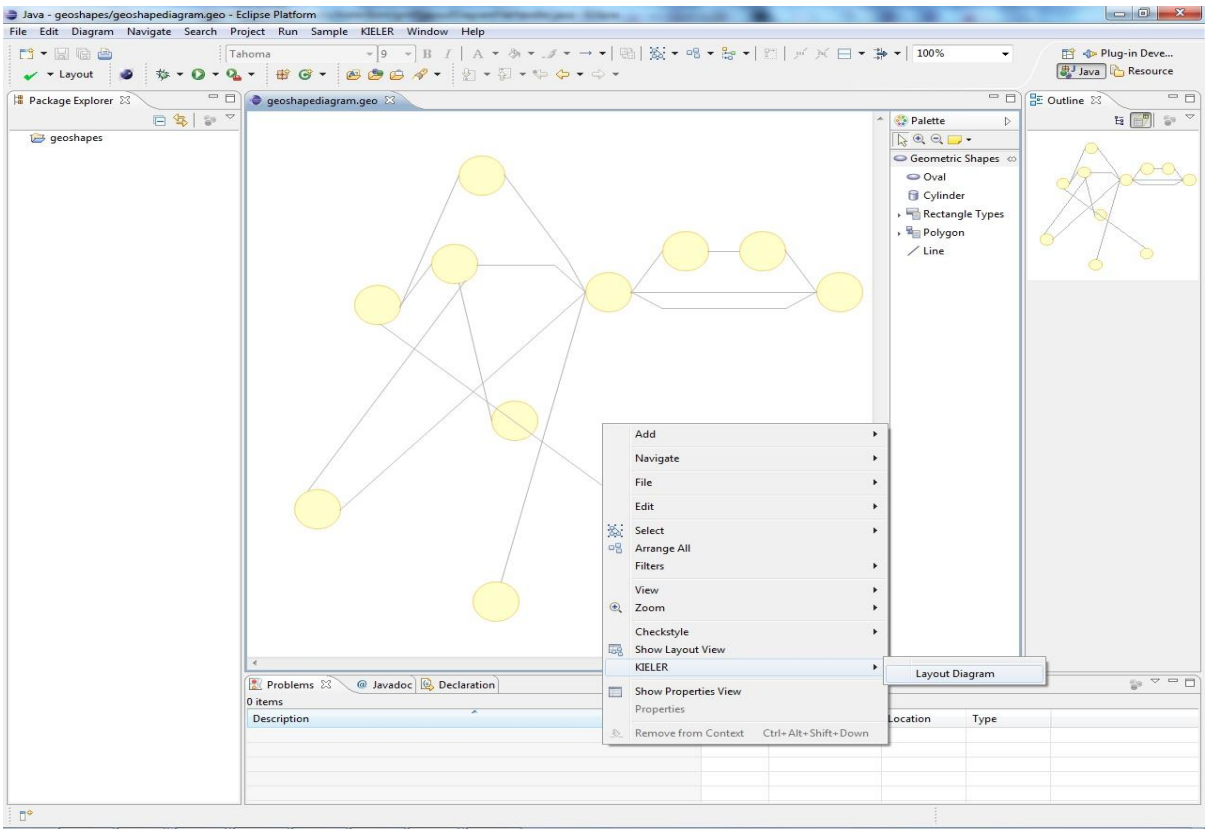

Java - geoshapes/geoshapediagram.geo - Eclipse Platform
File Edit Diagram Navigate Search Project Run Sample KIELER Window Help
Layout
Package Explorer
geoshapes
geoshapediagram.geo
Palette
Geometric Shapes
Oval
Cylinder
Rectangle Types
Polygon
Line
Outline
Add
Navigate
File
Edit
Select
Arrange All
Filters
View
Zoom
Checkstyle
Show Layout View
KIELER
Layout Diagram
Show Properties View
Properties
Remove from Context Ctrl+Alt+Shift+Down
Problems
Javadoc
Declaration
0 items
Description
Location
Type

Рис. 7

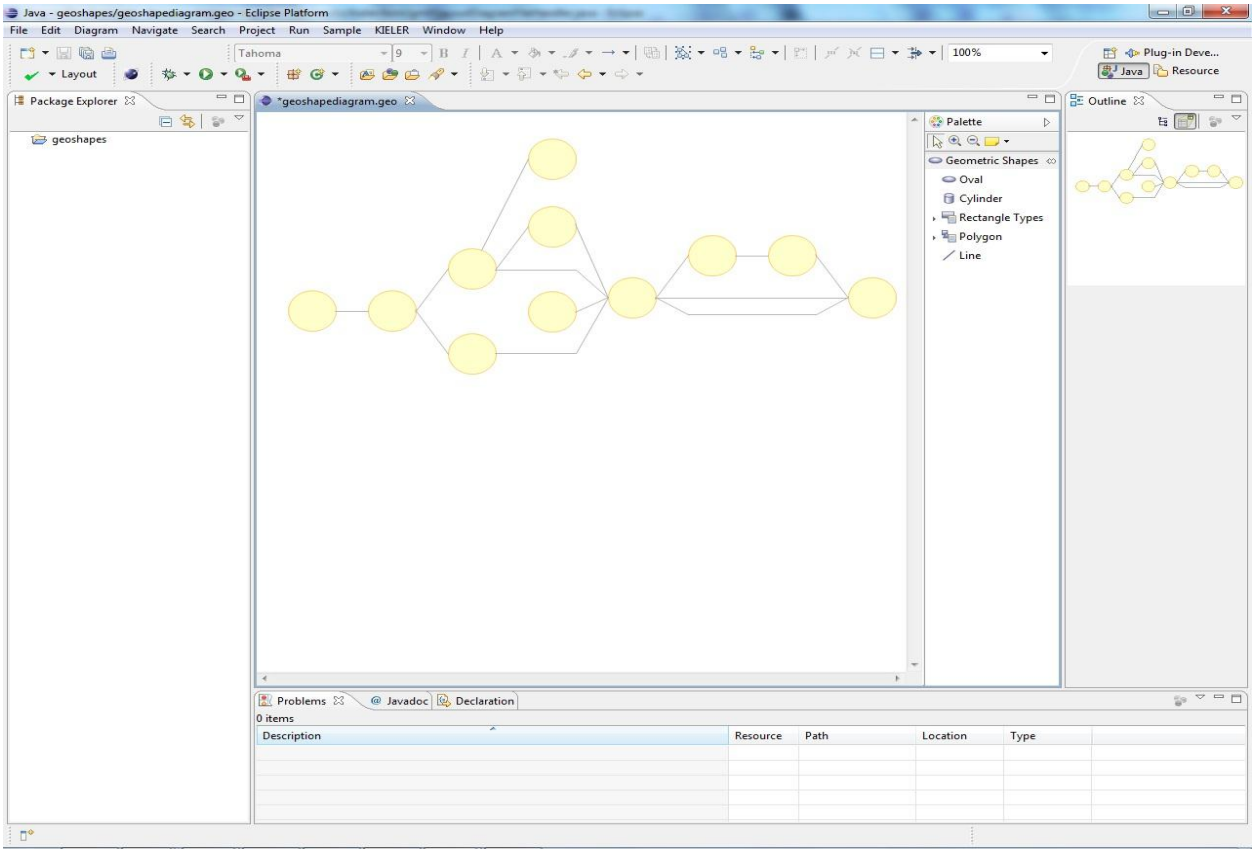

Java - geoshapes/geoshapediagram.geo - Eclipse Platform
File Edit Diagram Navigate Search Project Run Sample KIELER Window Help
Layout
Package Explorer
geoshapes
*geoshapediagram.geo
Palette
Geometric Shapes
Oval
Cylinder
Rectangle Types
Polygon
Line
Outline
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location
Type

Рис. 8

4. Результаты

В ходе выполнения данной работы были достигнуты следующие результаты.

1. Проведено исследование проекта KIELER и выделены необходимые компоненты для реализации раскладок.
2. Реализована интеграция механизма раскладок в готовый GMF редактор:
    a. процесс раскладки абстрагирован от модели KIELER;
    b. сохранена расширяемость KIELER’a, позволяющая абстрагироваться от конкретной библиотеки раскладок.
3. Проведены эксперименты, показавшие удобство применения раскладок при выполнении трансформаций над диаграммами.

В дальнейшем планируется реализовать генерацию кода описанной функциональности при создании редактора GMF.

5. Литература

1. Hauke Fuhrmann and Reinhard von Hanxleden. Taming graphical modelling. Real-Time and Embedded Systems Group, Department of Computer Science Christian-Albrechts-Universität zu Kiel
2. Olshausenstr Martin Muller. View Management for Graphical Models. Christian-Albrechts-Universität zu Kiel 2010
3. Björn Duderstadt, Evolutionary Meta Layout for KIELER. Christian-Albrechts-Universität zu Kiel, 2011
4. Stephan Wersig, Ein Web Service für das automatische Layout von Graphen, 2011
5. Miro Spönemann, Hauke Fuhrmann, and Reinhard von Hanxleden. Automatic Layout of Data Flow Diagrams in KIELER and Ptolemy II. Technical Report 0914, Christian-Albrechts-Universität zu Kiel, Department of Computer Science, 2009
6. Hauke Fuhrmann, Miro Spönemann, Michael Matzen, and Reinhard von Hanxleden. Automatic Layout and Structure-Based Editing of UML Diagrams. In Proceedings of the 1st Workshop on Model Based Engineering for Embedded Systems Design (M-BED'10), Dresden, 2010.
7. Graphical Modelling Project URL: http://www.eclipse.org/modeling/gmp/
8. Eclipse http://www.eclipse.org/
9. Microsoft DSL Tools http://msdn.microsoft.com/en-us/library/bb126259.aspx
10. UML 2.4.1 http://www.omg.org/spec/UML/2.4.1/
11. ATL http://wiki.eclipse.org/MMT/Atlas_Transformation_Language_(ATL)
12. Microsoft Visio http://visio.microsoft.com/en-us/Pages/default.aspx
13. Д.В.Кознов О спецификации диаграммных преобразований в графических редакторах. Вестник Санкт-Петербургского Университета, Сер. 10. Вып. 3. 2011. С. 100-111.

14. Д.В.Кознов. Основы визуального моделирования. Учебное Пособие. Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. 246 с.
15. Д.В.Кознов. Разработка и сопровождение DSM-решений на основе MSF // Системное программирование / Вып. 3, под ред. А.Н.Терехова и Д.Ю.Булычева. СПб.: Изд. СПбГУ, 2008. С. 80-96.
16. А.Сорокин, Д. Кознов. Обзор проекта Eclipse Modeling Project // Сб. Системное программирование./ Вып. 5, под ред. А.Н.Терехова и Д.Ю.Булычева. СПб.: Изд. СПбГУ, 2010. С. 6-31.