На правах рукописи

Литвинов Юрий Викторович

Разработка визуальных предметно-ориентированных языков

Специальность 05.13.11 —

математическое и программное обеспечение
вычислительных машин, комплексов, систем и сетей

Автореферат

диссертации на соискание учёной степени

кандидата технических наук

Санкт-Петербург
2015

Работа выполнена на кафедре системного программирования федерального государственного бюджетного образовательного учреждения высшего профессионального образования „Санкт-Петербургский государственный университет“.

Научный руководитель:	доктор физико-математических наук, профессор **Терехов Андрей Николаевич**
Официальные оппоненты:	**Фамилия Имя Отчество,** доктор физико-математических наук, профессор, Основное место работы с длинным длинным длинным длинным длинным длинным длинным названием, старший научный сотрудник
	Фамилия Имя Отчество, доктор физико-математических наук, Основное место работы с длинным длинным длинным длинным названием, старший научный сотрудник
Ведущая организация:	Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования с длинным длинным длинным длинным названием

Защита состоится DD mmmmmmmm YYYY г. в XX часов на заседании диссертационного совета NN на базе Название учреждения по адресу: Адрес.

С диссертацией можно ознакомиться в библиотеке Название библиотеки.

Автореферат разослан DD mmmmmmmm YYYY года.

Ученый секретарь
диссертационного совета
Номер совета, д.ф.-м.н. Фамилия Имя Отчество

Общая характеристика работы

Актуальность темы. Разработка сложных программных систем, несмотря на несколько десятилетий развития программной инженерии, до сих пор остаётся непростой задачей. Связано это отчасти с тем, что программное обеспечение невидимо и нематериально, его трудно себе представить. Бороться с этой проблемой помогает визуальное моделирование — подход, при использовании которого программа представляется в виде набора графических моделей, каждая из которых описывает её с разных точек зрения. Если при проектировании объектов реального мира используются чертежи, то при разработке сложных программных систем могут использоваться модели, описывающие разбиение системы на компоненты, протоколы взаимодействия объектов системы, и т.д. В отличие от чертежей, которые соотносятся с тем, как будет выглядеть в реальном мире проектируемый объект после того, как будет создан, вид визуальных моделей — лишь предмет договорённости между разработчиками. Каждый человек может представлять себе программу по-разному, что создаёт дополнительные трудности при применении визуального подхода. Тем не менее, благодаря наличию стандартных широко распространённых графических языков, визуальное моделирование повышает продуктивность труда и качество результирующего продукта при разработке. Существует довольно большое количество исследований, подтверждающих это экспериментально.

Сейчас визуальные модели используются в основном при анализе и проектировании, а также как средство документирования и передачи информации между разработчиками. Однако же программы целиком или их фрагменты возможно автоматически генерировать по набору визуальных моделей, что позволяет непосредственно использовать результаты анализа и проектирования и в значительной степени автоматизировать труд программистов.

При этом использование визуальных языков общего назначения, таких как UML, делает задачу разработки программного обеспечения только с помощью графических языков сложной в силу наличия семантического разрыва между кодом и моделями. Такие языки работают в тех же терминах, в которых пишется исходный код на традиционных текстовых языках (классы, объекты, компоненты и т.д.), поэтому, чтобы полностью специфицировать поведение системы и сделать возможной автоматическую генерацию, модель должна содержать в себе столько же информации, что и исходный код программы, но это противоречит самому понятию модели

как некоего упрощения моделируемого объекта. На самом деле, визуальная модель в этом случае даже менее удобна, чем код программы — визуальные символы занимают на экране больше места, чем текст. Если же визуальная модель будет изображать только важные аспекты функционирования системы, опуская излишние подробности, то её можно будет сохранить обозримой и полезной для человека, но это сделает её бесполезной для исполнителя (например, для интерпретатора или генератора исходного кода). Именно так, в основном, используется UML сейчас — как средство для анализа и дизайна системы, а сама система специфицируется ручным кодированием на текстовых языках. Большинство инструментов для рисования UML-диаграмм позволяют сгенерировать заглушки, куда предполагается дописывать код вручную, но существенного выигрыша для разработчиков это не даёт.

Существует принципиально другой подход к использованию визуального моделирования, называемый предметно-ориентированным моделированием. Он основан на том наблюдении, что иногда создать новый язык для какой-то узкой предметной области или даже для конкретной задачи и решить задачу на нём оказывается быстрее и эффективнее, чем решать эту задачу на языке общего назначения. В таком случае наличие у средств поддержки создаваемого языка знаний о предметной области позволяет добиться полной автоматической генерации программ про визуальным моделям.

Существует много широко известных текстовых предметно-ориентированных языков, например, язык для работы с данными SQL, язык для работы с текстами awk, средства описания контекстно-свободных грамматик для генераторов синтаксических анализаторов. В каждом из этих примеров программа на предметно-ориентированном языке работает в терминах той предметной области, для которой этот язык создан, что даёт возможность, не задумываясь о деталях реализации, решать требуемую задачу. Например, современные генераторы синтаксических анализаторов позволяют задавать грамматику в виде, очень похожем на формы Бэкуса-Наура, при этом программист может не думать о том, как будет реализован синтаксический анализатор для этой грамматики: каким-либо из автоматных методов или рекурсивным спуском. Это позволяет реализовывать синтаксические анализаторы даже людям, весьма поверхностно представляющим себе алгоритмы синтаксического анализа. В предметно-ориентированных языках знания о предметной области „спрятаны“ в инструментальные средства, что позволяет существенно расширить круг пользователей языка, вплоть до того, что на нём смогут программировать люди, далёкие от программирования.

Ряд исследований показывает, что продуктивность труда программистов при использовании предметно-ориентированных языков вырастает в 3-10 раз по сравнению с использованием языков общего назначения, поэтому такой подход представляется весьма перспективным.

Разумеется, создавать новый предметно-ориентированный визуальный язык и инструментальные средства его поддержки „с нуля“ для каждой узкой предметной области или конкретной задачи было бы неоправданно трудозатратно. Поэтому существуют специальные средства для автоматизации этой задачи, называемые „DSM-платформа“, или „MetaCASE-средство“. Такие средства позволяют задать синтаксис визуального языка, используя какой-либо формализм (как правило, это метамодели), и автоматически сгенерировать редактор этого языка и другие средства инструментальной поддержки (мы будем называть результат генерации термином „DSM-решение“). Это позволяет реализовывать технологии программирования, использующие новые предметно-ориентированные языки, за время порядка дней, что делает предметно-ориентированное моделирование оправданным даже для небольших проектов. Существуют зрелые исследовательские и промышленные DSM-платформы, такие как Eclipse Modeling Project, MetaEdit+ и другие; однако же, несмотря на значительные преимущества предметно-ориентированного моделирования, применяется оно довольно редко. Связано это, в частности, с недостатками существующих платформ и отсутствием развитой методологической базы для их применения. Во многих случаях для создания предметно-ориентированного решения требуется привлекать экспертов в создании языков, которыми зачастую оказываются авторы выбранной для реализации этого решения DSM-платформы, поэтому позволить себе это могут лишь крупные компании. Такая ситуация указывает на необходимость продолжения исследований в этой области, что и стало предметом данной работы.

Целью данной работы является исследование предметно-ориентированных визуальных языков, их свойств, процесса их разработки, создание методологии разработки предметно-ориентированных решений, достаточно простой в применении, чтобы свой предметно-ориентированный язык и инструментальные средства для него мог создавать даже человек, не имеющий опыта и специальной подготовки в создании визуальных языков. Также требуется создать технологию, эту методологию реализующую, которая позволяла бы неспециалистам создавать свои предметно-ориентированные решения в короткие сроки.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. провести обзор подходов к реализации визуальных языков и средств, их реализующих;
2. разработать в рамках DSM-платформы QReal простую в использовании технологию создания предметно-ориентированных языков;
3. реализовать разработанную технологию в виде промышленного продукта;
4. провести апробацию технологии путём создания нескольких DSM-решений с её помощью.

Научная новизна данной работы заключается в следующем:

1. Разработана новая методика и набор инструментальных средств для создания предметно-ориентированных языков с помощью графического языка метамоделирования и сопутствующих визуальных языков. Методика предполагает применение предметно-ориентированного подхода „самого к себе“, то есть предметно-ориентированные языки используются для описания всей функциональности разрабатываемых инструментальных средств для нового языка: редактора диаграмм, генераторов кода на текстовых языках по диаграммам, интерпретаторов, средства проверки ограничений на диаграммы, средства поддержки рефакторингов.
2. Предложен новый способ создания предметно-ориентированного языка: „метамоделирование на лету“. Способ предполагает изменение и дополнение визуального языка прямо в процессе создания диаграммы на нём, без использования отдельного метаредактора. В процессе разработки языка при таком подходе не требуется оперировать с понятиями „метамодель“ и „метаредактор“, что снижает требования к квалификации пользователей.
3. С использованием предложенных методик разработаны новые предметно-ориентированные языки и средства инструментальной поддержки для них: язык программирования роботов и среда QReal:Robots (также известная как TRIKStudio), средство программирования приложений для мобильных телефонов QReal:Ubiq, средство разработки аппаратных систем QReal:HaSCoL.

Практическая ценность данной работы определяется использованием полученных результатов при разработке DSM-платформы QReal, а также ряде

DSM-решений, созданных с её помощью, самым зрелым из которых стала среда программирования роботов QReal:Robots (TRIKStudio), предназначенная для обучения школьников основам информатики и кибернетики с использованием робототехнических конструкторов ТРИК, Lego Mindstorms NXT, Lego Mindstorms EV3.

Среда QReal разрабатывается в рамках деятельности научно-исследовательской группы по изучению визуального моделирования под руководством проф. А.Н. Терехова с 2007 года и базируется на более чем двадцатилетнем опыте коллектива кафедры системного программирования Санкт-Петербургского государственного университета в разработке графических языков. Проект имеет открытый исходный код[1], разрабатывается на языке C++ с использованием библиотеки Qt силами студентов и преподавателей кафедры, автор данной диссертации — один из руководителей проекта. QReal создаётся как средство визуального моделирования, поддерживающее ряд широкоизвестных визуальных языков (UML 2.0, BPMN, блок-схемы), и одновременно как DSM-платформа, позволяющая быстро и без специальных знаний создавать свои собственные визуальные языки и DSM-решения на их основе. На данный момент среда существует в виде работающего прототипа. Проект поддержан грантом Санкт-Петербургского государственного университета 6.39.1054.2012. DSM-платформа QReal использовалась для реализации ряда предметно-ориентированных решений, использовавшихся в проектах компании „ЛАНИТ-Терком“, связанных с разработкой информационных систем и систем компьютерного зрения.

Среда программирования роботов QReal:Robots (или TRIKStudio) — на данный момент наиболее зрелая предметно-ориентированная технология, созданная с помощью среды QReal. Первый прототип среды программирования был разработан автором данной диссертации с использованием системы QReal примерно за неделю и включал в себя визуальный язык из примерно 20 сущностей, редактор к нему и интерпретатор, позволяющий исполнить программу на компьютере, посылая команды роботу Lego Mindstorms NXT по интерфейсу Bluetooth. На данный момент система переименована в TRIKStudio и обладает возможностями управления роботами ТРИК, Lego Mindstorms NXT, Lego Mindstorms EV3 по WiFi, Bluetooth, USB, генерации программы с последующей загрузкой её на робот для автономного исполнения и исполнения программы на двухмерной модели робота на экране компьютера.

[1]Страница проекта и репозиторий с исходным кодом на GitHub, URL: https://github.com/qreal/qreal

Апробация работы заключается в следующем.

- Некоторые результаты данной работы были доложены на второй научно-технической конференции молодых специалистов „Старт в будущее“ (Санкт-Петербург, 2011). Доклад был отмечен наградой.
- Результаты, связанные с применением разработанной технологии при создании среды QReal:Robots, были доложены на VII Международной научно-практической конференции „Современные информационные технологии и ИТ-образование“ (Москва, 2012).
- Результаты, связанные с применением разработанной технологии для разработки предметно-ориентированного языка для платформы Ubiq, были доложены на международной конференции „10th Conference of Open Innovations Association FRUCT“ (Tampere, 2011).
- Результаты диссертации использовались при проектировании и реализации DSM-платформы QReal и ряда DSM-решений, созданных с её помощью, включая среду программирования роботов QReal:Robots. Среда QReal использовалась для создания нескольких предметно-ориентированных решений в ЗАО “ЛАНИТ-Терком”. Среда QReal:Robots демонстрировалась на Открытых состязаниях Санкт-Петербурга по робототехнике в 2012 году и на робототехническом фестивале „Робофест 2012“ в Москве. На данный момент эта среда переименована в TRIKStudio и используется как основное средство программирования кибертенического конструктора ТРИК, используется в нескольких робототехнических кружках в России и на мастер-классах по робототехнике, проводимых компанией „Кибернетические технологии“.
- По теме диссертации опубликовано пять научных работ (три из них — в сборнике из списка ВАК, две другие — в сборнике, входящем в РИНЦ) и десять тезисов докладов на конференциях (под авторством или в соавторстве с автором диссертации).

Структура и объём работы. Диссертация состоит из введения, пяти глав, заключения, списка сокращений и условных обозначений, списка литературы (81 наименование). Объем основной части работы — 149 страниц с 34 рисунками и 3 таблицами.

Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, формулируется цель работы, научная новизна и практическая значимость, приводятся сведения об апробации работы.

В **Главе 1** приводятся основные понятия, используемые в предметно-ориентированном визуальном моделировании. Обсуждается структура визуального языка: синтаксис (абстрактный, конкретный и синтаксис сериализации), уровни абстракции (предметная область, модель, метамодель, метаметамодель), вводится классификация формальных языков по важным для дальнейшего изложения свойствам.

По форме представления информации языки делятся на следующие группы: *текстовые* языки — используют в основном текст для представления информации, *графические* или *визуальные* языки, использующие визуальные символы, и *текстографические* языки, активно использующие и текст, и графику для представления программы. Графические языки, в свою очередь, делятся на графовые (в которых диаграмма представляет собой помеченный мультиграф, либо сводится к нему) и неграфовые (не сводящиеся к помеченному мультиграфу).

По используемой модели вычислений языки делятся на *статические* (задающие структуру разрабатываемой системы) и *динамические* (описывающие поведение), которые, в свою очередь, делятся на языки, модель вычислений которых основана на сетях Петри (с простыми токенами — ориентированные на поток управления, или с токенами, содержащими в себе существенную информацию — ориентированные на поток данных), и языки, в основе которых лежат диаграммы конечных автоматов.

Глава 2 содержит обзор существующих подходов к созданию DSM-решений. Рассматриваются существующие методологии разработки предметно-ориентированных языков, обсуждаются возможности, достоинства и недостатки существующих DSM-платформ, включая зрелые системы и академические разработки. Результаты анализа существующих DSM-платформ сведены в таблицу 1.

Делаются выводы о том, что все существующие среды не автоматизируют весь жизненный цикл создания предметно-ориентированного решения, особенно его начальные этапы — если автор DSM-решения уже „знает, что писать“ (то есть имеет в голове ясное представление о предметной области и даже метамодель создаваемого языка), то к его услугам множество существующих инструментов. Но если требуется вести разработку предметно-

Таблица 1: Основные возможности существующих DSM-платформ

Название	Метаязык	Метаредактор	Конкретный синтаксис	Ограничения
MetaEdit+	GOPPRR	Визуальный или диалоговые окна	Визуально	Только средствами метаязыка
Eclipse Modeling Project	Ecore (аналог MOF)	Визуальный, текстовый, импорт метамодели	Визуально или вручную поверх библиотек	Текстовые (OCL)
Generic Modeling Environment	Свой, довольно развитый	Визуальный	Настройкой существующих фигур	Текстовые (OCL)
PSL/PSA	Сущность-связь	Текстовый	Нет	Нет
AToM3	Сущность-связь	Визуальный	Визуально	Вручную, на Python
Microsoft Modeling SDK	Свой (диаграммы классов)	Визуальный	Настройкой существующих фигур или кодированием на C#	Вручную поверх библиотеки
Pounamu	Сущность-связь	Визуальный	Визуально	Визуальным языком общего назначения или Java
DOME	Свой (DSTL), довольно развитый	Визуальный	Настройкой существующих фигур или кодированием на Alter	Вручную, на Alter
MetaLanguage	Сущность-связь	Визуальный	Неизвестно	Да

ориентированного решения „с нуля“ (начиная с оценки осуществимости и анализа предметной области), очень многое придётся делать без какой-либо инструментальной поддержки и, как правило, без руководства к действию. Таким образом, явно имеется пробел в существующих исследованиях, который данная работа призвана помочь заполнить.

Глава 3 содержит описание предлагаемого подхода к разработке DSM-решений: приводятся этапы жизненного цикла DSM-решения, обсуждается

возможная степень автоматизации каждого этапа, формулируются требования на средства автоматизации, приводится описание предлагаемой технологии. Предлагается два варианта методологии разработки — „Классическая“ методология и методология „Метамоделирования на лету“, которая является вариантом классической и служит для упрощения первых этапов жизненного цикла DSM-решения. Схематически „Классическая“ методология изображена на рисунке 1.

Рис. 1: „Классическая“ методология разработки визуального предметно-ориентированного языка.

Фаза разработки и внедрения состоит из нескольких итераций, порядок действий для каждой из которых представлен на рисунке 2.

Такая методология называется классической, поскольку примерно такой схемы придерживается большинство существующих DSM-платформ и большинство авторов, описывающих процесс создания предметно-ориентированных языков и дающих рекомендации по этому процессу. Вклад данного исследования состоит в структуризации этапов жизненного цикла, описании действий на каждом этапе жизненного цикла и предложения технологии автоматизации каждого этапа. Основная идея, предлагаемая здесь — использование визуальных языков на каждом этапе жизненного цикла, вплоть до описания генераторов.

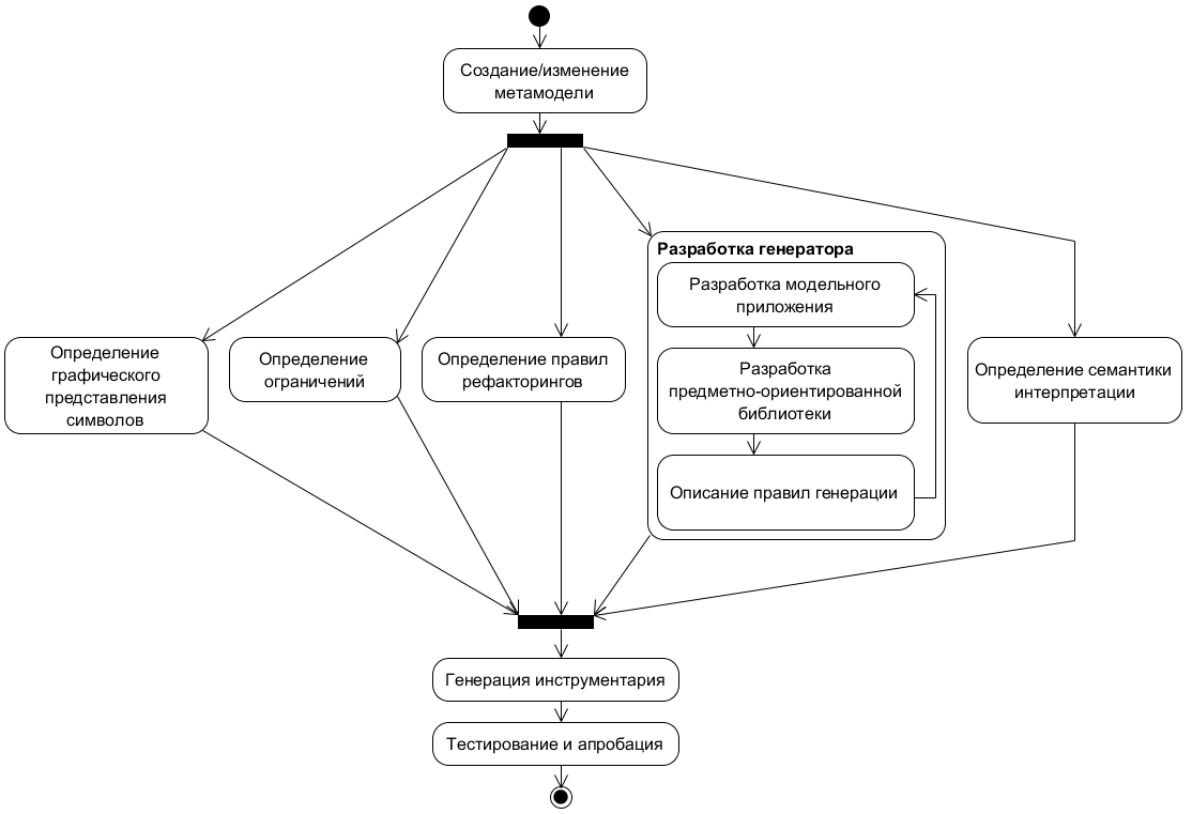

Рис. 2: Итерация проектирования и разработки языка.

С методологической точки зрения научную новизну имеет предлагаемый здесь подход „Метамоделирования на лету“. Ключевой принцип данной методологии состоит в том, что создание визуального языка проходит непосредственно в процессе рисования диаграммы, без использования метаредактора.

Основной этап предлагаемой методологии — прототипирование языка — начинается сразу после этапа анализа применимости. Разработчик языка и будущий пользователь работают за одним рабочим местом. При запуске DSM-платформы они видят канву для рисования и пустую палитру. Пользователь объясняет, что он примерно хотел бы нарисовать, разработчик языка добавляет на палитру новые элементы, определяет для них графическое представление, пользователь рисует. Пользователь может сказать, что такой-то элемент должен содержать такую-то дополнительную информацию, тогда разработчик добавляет элементу новое свойство, задаёт его тип и значение по умолчанию, и пользователь продолжает рисовать диаграмму, используя новое свойство. Через некоторое время пользователь может сам добавлять и редактировать типы элементов, и работа полностью передаётся ему, разработчик языка лишь следит за процессом и консультирует при необходимости пользователя. Работа заканчивается, когда модельное приложение полностью нарисовано,

после чего текущая интерпретируемая метамодель сохраняется в виде, пригодном для дальнейшего редактирования в метаредакторе. После этой фазы идут итерации „классической“ методологии по доработке созданного прототипа, дополнению его ограничениями, рефакторингами, интерпретатором и генератором, подготовки инсталляционного пакета, развертывания и сбора обратной связи. На этих этапах пользователи, как и в „классической“ модели, непосредственно в разработке не участвуют, поскольку этапы гораздо более продолжительны во времени и прямо при пользователе выполнены быть не могут.

Модель жизненного цикла языка, использующая „метамоделирование на лету“, представлена на рисунке 3.

Рис. 3: Методология „метамоделирования на лету“.

В **Главе 4** анализируются результаты реализации инструментальных средств поддержки предлагаемой технологии в проекте QReal. Описываются возможности системы QReal, связанные с поддержкой техник метамоделирования, включая метамоделирование на лету, принятые архитектурные решения.

Приложение А содержит примеры применения результатов, описанных в данной диссертации, для разработки DSM-решений. Описывается среда QReal:Robots, то, какие преимущества были получены от использования DSM-платформы QReal при её разработке, то, чем QReal помочь не смог, и почему. Также приводится описание среды разработки сервисов для мобильных телефонов QReal:Ubiq и среды разработки аппаратуры QReal:HaSCoL,

описываются их визуальные языки, достоинства и недостатки принятых при их создании подходов.

В **Приложении В** описывается визуальный метаязык системы QReal.

Реализация предложенных в данной диссертации методик осуществлялась коллективом студентов и аспирантов кафедры системного программирования СПбГУ в рамках проекта QReal. Хочется особо отметить вклад студентов, работавших над данным проектом под руководством автора диссертации: Абрамова Ивана Александровича, Дерипаска Анны Олеговны, Гудошниковой Анны Андреевны, Жуковой Беллы Юрьевны, Заболотных Елены Петровны, Занько Софьи Владимировны, Иванова Всеволода Юрьевича, Кузенковой Анастасии Сергеевны, Кузнецовой Марьи Юрьевны, Курбанова Рауфа Эльшад оглы, Назаренко Владимира Владимировича, Нефёдова Ефима Андреевича, Никольского Кирилла Андреевича, Осечкиной Марии Сергеевны, Птахиной Алины Ивановны, Пышновой Александры Витальевны, Савина Никиты Сергеевича, Соковиковой Натальи Алексеевны, Такун Евгении Игоревны, Тихоновой Марии Валерьевны, всех студентов, работавших под руководством Брыксина Тимофея Александровича, а также вклад Дмитрия Мордвинова и Ирины Брюхановой. Без них данная диссертация вряд ли была бы возможна.

В **заключении** приведены основные результаты работы.

Основные результаты работы

1. Проведён анализ различных подходов к реализации визуальных языков и различных технологических средств, их реализующих.
2. Разработана методика и набор инструментальных средств для создания предметно-ориентированных языков с помощью графического языка метамоделирования и сопутствующих визуальных языков.
3. Предложен новый способ метамоделирования: „метамоделирование на лету“.
4. Предложенные методики и технологии реализованы в виде промышленного продукта QReal.
5. Проведена апробация при создании редактора, генератора, средств проверки ограничений среды QReal:Robots и других предметно-ориентированных решений.

Публикации автора по теме диссертации

В изданиях из списка ВАК РФ

1. Средства быстрой разработки предметно-ориентированных решений в metaCASE-средстве QReal / А.С. Кузенкова, А.О. Дерипаска, К.С. Таран [и др.] // Научно-технические ведомости СПбГПУ, Информатика, телекоммуникации, управление. 2011. № 4 (128). С. 142–145.

2. Ю.В. Литвинов. Реализация визуальных средств программирования роботов для изучения информатики в школах // Компьютерные инструменты в образовании. 2013. № 1. С. 36–45.

3. Терехов А.Н. Брыксин Т.А. Литвинов Ю.В. QReal: платформа визуального предметно-ориентированного моделирования // Программная инженерия. 2013. № 6. С. 11–19.

В других изданиях

4. QReal DSM platform-An Environment for Creation of Specific Visual IDEs. / Anastasiia Kuzenkova, Anna Deripaska, Timofey Bryksin [и др.] // ENASE. 2013. С. 205–211.

5. Терехов А.Н., Брыксин Т.А., Литвинов Ю.В. Среда визуального программирования роботов QReal:Robots // III Всероссийская конференция "Современное технологическое обучение: от компьютера к роботу"(сборник тезисов). 2013. С. 2–5.

6. Кузенкова А.С., Литвинов Ю.В. Поддержка механизма рефакторингов в DSM-платформе QReal // Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада "Технологии Microsoft в теории и практике программирования". Изд-во СПбГПУ, 2013. С. 71–72.

7. Литвинов Ю.В. Визуальные средства программирования роботов и их использование в школах // Современные информационные технологии и ИТ-образование, сборник избранных трудов VII Международной научно - практической конференции. ИНТУИТ.РУ, 2012. С. 858–868.

8. Osechkina Maria, Litvinov Yuri, Bryksin Timofey. Multistroke Mouse Gestures Recognition in QReal metaCASE Technology // SYRCoSE 2012: Proceedings

of the 6th Spring/Summer Young Researchers' Colloquium on Software Engineering. ISPRAS, 2012. С. 194–200.

9. Кузенкова А.С., Дерипаска А.О., Литвинов Ю.В. Поддержка метамоделирования в среде визуального программирования QReal // Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада "Технологии Microsoft в теории и практике программирования". Изд-во СПбГПУ, 2011. С. 100–101.

10. Брыксин Т.А., Литвинов Ю.В. Технология визуального предметно-ориентированного проектирования и разработки ПО QReal // Материалы второй научно-технической конференции молодых специалистов «Старт в будущее», посвященной 50-летию полета Ю.А. Гагарина в космос. ОАО "КБСМ 2011. С. 222–225.

11. Кузенкова А.С., Брыксин Т.А., Литвинов Ю.В. Метамоделирование: современный подход к созданию средств визуального проектирования // Материалы второй научно-технической конференции молодых специалистов «Старт в будущее», посвященной 50-летию полета Ю.А. Гагарина в космос. ОАО "КБСМ 2011. С. 228–231.

12. Брыксин Т.А., Литвинов Ю.В. Среда визуального программирования роботов QReal:Robots // Материалы международной конференции "Информационные технологии в образовании и науке". Самарский филиал МГПУ, МГПУ, 2011. С. 332–334.

13. Ubiq Mobile + QReal: a Technology for Development of Distributed Mobile Services / Timofey Bryksin, Yuri Litvinov, Valentin Onossovski [и др.] // 10th Conference of Open Innovations Association FRUCT and the 2nd Finnish-Russian Mobile Linux Summit: Proceedings. State University of Aerospace Instrumentation (SUAI), 2011. С. 27–35.

14. Поддержка жестов мышью в мета-CASE-системах / М.С. Осечкина, Т.А. Брыксин, Ю.В. Литвинов [и др.] // Системное программирование. 2010. № 5. С. 52–75.

15. Архитектура среды визуального моделирования QReal / А.Н. Терехов, Т.А. Брыксин, Ю.В. Литвинов [и др.] // Системное программирование. 2009. № 4. С. 171–196.