

УДК 007:681.512.2

СРЕДСТВА ТИПИЗАЦИИ И ПРАГМАТИКА ЯЗЫКА МОДЕЛИРОВАНИЯ  
ОРГ-МАСТЕР<sup>1</sup>

Д.В. Кудрявцев, Д.В. Кознов, Л.Ю. Григорьев

В настоящий момент активно развивается область управления архитектурой предприятия (Enterprise Architecture Management), направленная на структуризацию и преобразование деятельности бизнес-компаний и на целостное управление бизнесом. Технология ОРГ-Мастер является российской разработкой в этой области, применяемой на практике в течение последних пятнадцати лет. В работе представлены новые возможности языка моделирования ОРГ-Мастер, реализованные в последней версии программных средств – более строгая типизация и уточненная прагматика.

**Ключевые слова:** архитектура предприятия, моделирование архитектуры предприятия, онтологический инжиниринг, онтологии, бизнес-процессы, BPMN, UML, предметно-ориентированное моделирование, визуальное моделирование, модельно-ориентированная разработка, DSM-подход.

Введение

В настоящее время бизнес-организации вынуждены непрерывно развиваться и трансформироваться, чтобы реагировать на изменения рынка, среды, технологий или же для того, чтобы инициировать эти изменения. Для проведения целенаправленных, управляемых и успешных трансформаций принято выделять так называемую архитектуру предприятия (Enterprise Architecture, EA), которая отражает организацию (устройство) предприятия и предназначена для его анализа, проектирования изменений и преобразований [1–3]. Для моделирования EA (Enterprise Architecture Modeling) и дальнейшей работы с этими моделями используются специальные программные средства, которые в последнее время принято называть «инструменты управления архитектурой предприятия»<sup>2</sup> [4, 5]. В этих инструментах активно применяется визуальное моделирование, т.е. используются «чертежи, позволяющие точно изображать хорошо структурированную информацию в понятной форме» [6], и, следуя традиции программной инженерии, для их разработки создаются, в основном, графовые нотации<sup>3</sup>. Данная область в настоящий момент активно развивается, о чем свидетельствует, например, отчет Gartner Group за 2012 год о существующих на рынке EAM-инструментах. Однако до стандартизации в этой области пока еще очень далеко: так, например, лидеры этого рынка, продукты IBM Rational System Architect [9] и Mega [10] используют разные методологии и наборы диаграмм<sup>4</sup>. В этой области используются многочисленные языки моделирования, как созданные непосредственно для моделирования EA, так и взятые из смежных областей: это стандарты моделирования бизнес-процессов [11], язык моделирования архитектуры предприятия Archimate [12], карты стратегий [13], шаблон бизнес-модели Остервальдера [14] и средства инженерии знаний (mind maps, concept maps) [15], а также средства моделирования программного обеспечения, например, UML (Unified Modeling Language) [16] и многое другое. В работе [17] предпринята попытка создать классификацию визуальных языков моделирования, в связи с чем найдено и рассмотрено около пятидесяти различных типов нотаций, в монографии [18] проведен обзор средств визуализации знаний и также идентифицировано огромное количество нотаций и подходов. Трудности со стандартизацией в EAM происходят из-за большого разнообразия прикладных областей, где используется данный подход: мировой бизнес очень вариативен, существует большое количество точек зрения и подходов к ведению бизнеса, а также различаются задачи, решаемые с помощью EA<sup>5</sup>.

<sup>1</sup> Работа выполнена при частичной поддержке гранта РФФИИ 12-01-00415-а.

<sup>2</sup> управление архитектурой предприятия – Enterprise Architecture Management, EAM; соответствующие программные продукты – EAM-tools (EAM-инструменты).

<sup>3</sup> Т.е. нотации, основывающиеся на концепции математического графа – набора вершин, которые могут по-разному изображаться и которые соединяются линиями, т.е. дугами, которые также часто называют ребрами [7].

<sup>4</sup> Самой стандартизованной подобластью EAM является моделирование бизнес-процессов (business process modeling, BPM) – здесь было создано несколько языков, которые в настоящий момент объединены стандарт комитетом OMG (Object Management Group) в рамках стандарта BPMN (Business Process Management Notation, обзор на русском языке стандартов в этой области и дальнейшие ссылки можно найти, например, в [11]).

<sup>5</sup> Если продолжить аналогию с программной инженерией, то даже в этой достаточно узкой по сравнению с бизнесом области есть проблемы со стандартизацией визуального моделирования (другое распространенное название – модельно-ориентированная разработка, используемые англо-язычные термины – model-driven development, MDD, model-driven architecture, MDA). Приятный в конце 90-х г.г. прошлого века стандарт UML [16] к началу 2000-х г.г. не оправдал надежд на универсальность [19] и в итоге так и не стал широко использоваться на практике. В итоге стал развиваться альтернативный подход – предметно-ориентированное моделирование (DSM-подход, Domain Specific Modeling, DSM) [20]; вот ряд исследований на эту тему, выполненных в России: [21–23].

В России в течение последних 20 лет активно развивается и используется на практике методология и комплекс программных средств ОРГ-Мастер [24–29]. ОРГ-Мастер не является коробочным программным продуктом и представляет собой программно-методический комплекс, предназначенный для оказания сервисов в области ЕАМ для российских предприятий, акцентируясь на бизнес-моделировании<sup>6</sup>. Преимуществами данного комплекса являются развитые возможности работы с репозиторием модели, позволяющие систематизировать и интегрировать множество частных моделей и точек зрения на предприятие в рамках создаваемой для него ЕА. Таким образом, становится возможным не только создавать красивые картинки для начальников, но оперативно разрабатывать сложные и детальные модели различных предприятий, отделяя уровень внутреннего (рабочего) представления от внешнего, презентационного [25, 28].

В настоящей работе представлены результаты ревизии языка моделирования ОРГ-Мастер. Эта ревизия была предпринята в связи с бурным развитием в последние годы ЕАМ-средств [8, 31], а также в контексте создания новой версии средства программной поддержки ОРГ-Мастера. При этом авторы руководствовались следующими соображениями:

- старались вносить в язык черты, существенно увеличивающие его выразительную силу;
- вместе с тем стремились максимально сохранить исходную концепцию языка; эта концепция хорошо себя зарекомендовала на практике, она обладает целостностью и гибкостью, являясь плодом более чем двадцатилетней работы большого коллектива аналитиков и экспертов;
- в связи с этим, в частности, ограниченно вносили в язык различный «синтаксический сахар», стремясь в большей степени к интеграции с другими языками и нотациями через импорт/экспорт;
- в задачи авторов входила также формализация методологии бизнес-моделирования на уровне языковых средств;
- наконец, стремились упорядочить и формализовать сценарии использования языка.

### Обзор языка

Язык моделирования ОРГ-Мастер является очень простым и компактным. В него не включены сущности предметной области архитектурного моделирования – они надстраиваются над языком «сверху», с помощью задания пиктограмм и типов, через структуру модели, а также с помощью библиотечных средств. Такой подход открывает широкие возможности по использованию ОРГ-Мастера как в различных проектах по разработке архитектуры предприятий (ЕА-проектах), так и в разных предметных областях. Фактически язык оказывается средством разработки произвольных онтологий [26, 28, 32, 33] не обязательно в области ЕАМ. Главными конструкциями языка являются классификаторы и проекции, которые рассмотрены ниже.

Классификатор – это контейнер, который содержит дискретные, упорядоченные и иерархизированные элементы информации, которые, в свою очередь, описывают некоторый связный и достаточно однородный фрагмент предметной области. Информация, которая попадает в один классификатор, является однородной, поскольку основная задача классификатора – классифицировать, и это делается по принципу близости, похожести. Кроме того, классификатор позволяет упорядочить и дискретизировать (т.е. разбить на части) исходно непрерывную информацию, существующую в виде текстового документа или набора документов, а также в устной форме. Дискретные части, так называемые атомарные информационные единицы, хранящиеся в классификаторе, называются позициями. Дискретизация предметной области является важным шагом при проектировании архитектуры организации.

Процесс дискретизации информации в ОРГ-Мастере тесно связан с построением иерархии – в рассматриваемом информационном фрагменте выделяются разделы, которые, в свою очередь, могут содержать другие разделы, и т.д. «На дне» этой иерархии находятся листовые позиции.

Еще одним важным моментом упорядочивания и дискретизации информации является возможность задавать атрибуты. Требуется не только структурировать процессы, но также и описать их, т.е. определить для них набор свойств и заполнить значения этих свойств – важность, уровень зрелости и т.д. Иначе говоря, у позиций должны быть атрибуты. Атрибуты позиций описываются на уровне классификатора, для которого можно определить вид атрибута, присвоить ему имя и задать тип (числовой, текстовый, комментарий и т.д.). После этого у всех позиций данного классификатора появляется возможность задавать соответствующие значения атрибута. Можно сказать, что еще один атрибут каждой позиции – это ее имя, и туда можно помещать произвольный текст, но он не должен быть очень большим (несколько строк – не более).

Бывает, что в один классификатор попадают и вовсе разнородные объекты, не объединяемые в одну иерархию, но по каким-то причинам тесно связанные между собой. Например, в одном классификато-

<sup>6</sup> Другим фокусом в области ЕАМ является разработка интегральной ИТ-архитектуры предприятия – единого видения всех ИТ-систем, которые должны автоматизировать деятельность данного предприятия. Сервисы в области бизнес-моделирования предприятий на основе ОРГ-Мастера оказываются Санкт-Петербургской компанией «Бизнес Инжиниринг Групп» [35].

ре вместе со стратегическими активами организации могут оказаться цели и стратегии, которые основываются на вышеуказанных стратегических активах – и это будут такие специальные позиции.

Для того чтобы отразить факт наличия в одном классификаторе неоднородной информации, в ОРГ-Мастере используются пиктограммы. Для всей модели отдельно определяется множество допустимых пиктограмм, с классификатором связывается набор пиктограмм, причем каждая позиция в классификаторе может иметь пиктограмму из этого множества.

Отметим, что классификатор с механизмом атрибутов и пиктограммы являются двумя разными способами типизации.

Рассмотрим теперь проекции и связи в ОРГ-Мастере. Проекция – это отношение между классификаторами, в рамках которого могут устанавливаться связи между позициями этих классификаторов. В этих связях могут участвовать как листовые позиции, так и разделы – в определении проекции не накладывается дополнительных ограничений на позиции, которые могут участвовать в связях. Например, классификатор «Процессы» может быть связан проекцией с классификатором «Ролевая структура», и это означает, что каждый процесс имеет ссылку на организационную роль, которая за него отвечает. Но также можно установить связь раздела «Процессы управления» классификатора «Процессы» и входящих в него элементов с организационными ролями. Это избыточность того же рода, что и атрибуты классификатора – одни и те же для позиций разного типа. Выходом здесь могло бы быть создание дополнительных ограничений в проекции на типы связываемых элементов (например, задание этих ограничений как ассоциаций между типами).

В ОРГ-Мастере можно создавать проекции произвольной арности – как правило, от 1 до 7, хотя количество участников в проекции не ограничено (оно, разумеется, не может быть нулем или отрицательным числом). Если несколько классификаторов соединены проекцией, то разделы и листовые объекты в них могут быть соединены связями (в этом смысле разделы от листовых объектов ничем не отличаются) – по одному от каждого классификатора. Иначе говоря, связь – это набор из элементов позиций соответствующих классификаторов.

### **Средства типизации**

Одним из главных направлений модернизации языка было усиление типизации. С одной стороны, это было необходимо для дифференциации используемых при моделировании конструкций базового языка: оказалось, что, например, пиктограммы используются в различных смыслах – и как перечислимые типы для атрибутов, и как метки для позиций классификатора и связей в проекциях. Кроме того, развитие средств визуального моделирования в новой версии ОРГ-Мастера потребовало сквозной типизации всех конструкций – графические средства должны точно «знать», какие именно сущности необходимо визуализировать.

Итак, пиктограммы были названы типами и разбиты на следующие виды:

- тип позиции – возможность типизировать элементы классификатора;
- тип связей – возможность указывать определенный идентификатор связям в проекции;
- свободные атрибуты – возможность выбирать значение атрибута в классификаторе из преопределенного списка значений.

В каждой модели ОРГ-Мастера имеется справочник типов позиций, из которого можно выбирать подходящие типы, а при отсутствии таковых создавать новые. Этот список является деревом по отношению тип/подтип, каждый элемент которого (конечный и узловой) обязательно имеет текстовое значение и может иметь связанную с ним иконку.

В каждой модели ОРГ-Мастера имеется справочник типов связей. При создании проекции с ней связывается один или несколько типов связей, каждой связи назначается некоторый тип из этого списка. Тип связей используется при фильтрации в процессе разработки и генерации отчетов по модели.

Третий вид типов в ОРГ-Мастере – это свободные атрибуты, которые являются повторно используемыми перечислимыми типами. Свободный атрибут можно задавать в качестве типа у атрибута классификатора, выбирая для каждой позиции классификатора подходящее значение из списка допустимых, связанных с этим свободным атрибутом.

В языках программирования уже с 70–80-х г.г. прошлого века стало принято использовать строгую типизацию переменных. Однако в средствах моделирования ситуация с типизацией более гибкая – так, например, атрибуты в диаграммах классов UML могут не иметь типов, и в общем случае это не является ошибкой<sup>7</sup>. Необходимость строгой типизации, в общем случае – однозначно заданных связей между различными элементами языка моделирования, типами диаграмм и пр., на наш взгляд, должна так сказать, дозироваться, так как этот подход, с одной стороны, конечно, гарантирует корректность специ-

---

<sup>7</sup>Хотя при использовании UML совместно со средами разработки ПО (программного обеспечения) (т.е. с поддержкой генерации кода на Java, C# и пр.) это, безусловно, ошибка, о которой UML-пакеты обычно сообщают пользователям.

фикаций<sup>8</sup>, но с другой стороны, делает процесс моделирования негибким, неудобным и очень громоздким<sup>9</sup>.

Типы в ОРГ-Мастере используются во многом как метки при составлении выборок из модели при разработке отчетов, и в связи с этим строгий контроль типов не требуется. Но в новой версии ОРГ-Мастера они используются внешними графическими редакторами<sup>10</sup> и различными сторонними плагинами, которым нужно точно указать, какие именно элементы следует отображать и связывать на тех или иных диаграммах. Причем неточности здесь становятся причиной ошибок при работе этих средств.

Повышенный (по сравнению с прошлой версией языка) уровень строгости типового контроля введен в ОРГ-Мастере во многом для поддержки графических редакторов. Однако есть тип «Не задан», который проставляется для созданной позиции/связи автоматически и фактически позволяет элементам не иметь конкретного значения типа, хотя это считается не очень хорошим стилем. Тем не менее, это удобно, поскольку такой подход позволяет, например, выполнять процедуру множественного импорта элементов классификатора из сторонних источников – баз данных, таблиц Microsoft Excel и т.д.; после выполнения такого импорта в модели одновременно появляется много разных позиций, и требовать от аналитика сразу же назначить каждому из них нужный тип нецелесообразно, так как в этот момент не ясно, какой у кого должен быть тип, останутся ли все они или некоторая их часть в данном классификаторе и т.д.; соответствующие типы лучше проставить позднее. В пакете ОРГ-Мастер предполагается реализовать валидацию модели на предмет наличия в ней типов «Не задан».

Еще одной новой чертой является разделение множества типов, связываемого с классификатором, на базовые и вспомогательные типы. Каждая позиция классификатора должна иметь строго один базовый тип и может также иметь несколько вспомогательных. Так, если получилось, что в одном классификаторе попали единицы оборудования и виды должностей сотрудников (например, если классификатор описывает целиком какой-то департамент компании, и этот департамент небольшой, то может оказаться целесообразно создать один классификатор для такого департамента и поместить туда все<sup>11</sup>), то два этих вида позиций должны иметь различные базовые типы. Вспомогательные типы используются так же, как и пиктограммы позиций в прежней версии языка, кроме того, они позволяют строить сложные и нелинейные классификации – например, элемент организационной структуры предприятия может быть департаментом, а может – конкретной должностью сотрудника. Аналогичная возможность в объектно-ориентированном проектировании реализуется с помощью множественного наследования.

Наш механизм работы с типами, безусловно, не является надежным, в отличие, например, от поддержки типизации в языках программирования. В последнем случае для программы компилятор строит общее дерево, в рамках которого осуществляется статический контроль корректности, в том числе и контроль правильного использования типов. Репозиторий модели до некоторой степени является аналогом такого дерева, но слабым аналогом, позволяющим пользователю ошибаться, например, неправильно выбирать тип позиции (список допустимых типов для выбора может быть большим). В программе на языке программирования такая ошибка выявляется сразу же – начинаешь присваивать переменной, которую ошибочно определил как символьную, целое значение – компилятор сразу же сигнализирует об ошибке. Типы же элементов в ОРГ-Мастере используются не столь интенсивно и строго, поэтому многие ошибки определяются не автоматически, а «вручную» – при генерации отчетов и диаграмм по модели. Более того, такая практика проверки моделей (не только в смысле типов) широко используется аналитиками, работающими с программными средствами ОРГ-Мастера: аналитик ожидает увидеть в отчете или на диаграмме определенную картину, и если он видит что-то другое, то начинает разбираться с моделью. Такой уровень поддержки корректности вполне адекватен в нашей ситуации, так как цена ошибки здесь иная, чем в программировании, хотя, безусловно, в будущем мы приложим максимум усилий для автоматического обеспечения корректности работы с типами.

### Уточнение прагматики: режимы использования языка

Следуя семиотической традиции, остановимся на измерении нашего языка моделирования, которое называется прагматикой и определяет способ его использования [38]. В ходе модернизации ОРГ-Мастера прагматика языка моделирования была уточнена и формализована путем выделения ролей пользователей этого языка (программных средств) и описания сценариев (режимов) их работы.

<sup>8</sup> Подробно достоинства строгой типизации в моделировании обсуждаются, например, в работе [34].

<sup>9</sup> Кроме того, возможны и так называемые «ленивые» подходы проверки корректности спецификаций – т.е. ошибки позволяют делать (кто знает, может быть, это не ошибки, а незавершенные пока спецификации?), а после окончания моделирования (или в какой-то другой важный момент) производится пакетная валидация/верификация спецификаций (см. примеры в работах [35, 36]).

<sup>10</sup> Внешне редакторы – это средства диаграммного моделирования, созданные в рамках пакета ОРГ-Мастер для взаимодействия с клиентами, а также для реализации полнофункциональной графики. Внешние редакторы разработаны на базе Microsoft Visio и детально описаны в работе [37].

<sup>11</sup> Моделирование небольших предприятий и разработка небольших ЕА-проектов является отдельной очень интересной темой ЕАМ. На практике оказывается, что в этом случае целесообразно использовать несколько иные средства и подходы, в отличие от тех, которые используются в больших ЕА-проектах.

Для уточнения режимов использования программного продукта необходимо разделить уровни моделирования и ввести некоторые дополнительные понятия.

Язык моделирования ОРГ-Мастер является, по сути, метаязыком – моделирование ЕА не производится непосредственно в терминах этого языка, с помощью классификаторов и проекций, а выполняется посредством дополнительных понятий и сущностей, таких как «Цель», «Процесс», «Операция», «Организационная роль». Последние задаются методологией ОРГ-Мастера, и в каждом конкретном проекте они могут уточняться и расширяться. Таким образом, достигается гибкость и компактность языка моделирования, а также гибкость средств бизнес-моделирования ОРГ-Мастера. Последние часто меняются в связи с бурным развитием области ЕАМ, а также уточняются и детализируются в связи с использованием ЕАМ в различных узких предметных областях (государственных корпорациях, правительственных органах, малом бизнесе и т.д.).

Методология (дополнительные понятия и сущности «поверх» языка ОРГ-Мастер) синтаксически оформляется с помощью опорных моделей [29, 39, 40]. Последние представляют собой специальные модели ОРГ-Мастера (т.е. их можно открывать и редактировать как обычные модели), но главное их предназначение – задавать структуру рабочих моделей, соответствующих конкретным ЕА-проектам (аналог схемы базы данных). Таким образом, при создании каждой новой модели в ОРГ-Мастере пользователю предлагается выбрать подходящую опорную модель, из которой в текущую, вновь создаваемую модель экспортируются состав и спецификации классификаторов, проекций, типов и отчетов.

На сегодняшний день создана опорная модель, определяющая основные сущности для моделирования архитектуры бизнес-предприятий [24, 40]. Было также создано 6 опорных моделей для моделирования федеральных органов власти и субъектов федерации [39]<sup>12</sup>. В разработке находится еще одна опорная модель, ориентированная на архитектуру городского хозяйства [41]. В [42] описан проект по применению ОРГ-Мастера при разработке Web-системы в области русско-финских приграничных отношений. Для этого проекта не нашлось подходящей опорной модели, и все понятия и сущности создавались разработчиками проекта.

Авторами выделены следующие роли (сценарии) для использования языка ОРГ-Мастер (они также совпадают с ролями пользователей программных средств ОРГ-Мастер):

- «Методолог» – разработка и изменение опорных моделей;
- «Архитектор» – выбор и адаптация опорной модели при разработке конкретного ЕА-проекта;
- «Аналитик» – разработка ЕА-проекта: активная работа с базовым пакетом ОРГ-Мастер, т.е. создание позиций, связей, отчетов и т.д.;
- «Клиент» – ввод данных в модель ОРГ-Мастера посредством диаграмм (т.е. с помощью внешних редакторов) в рамках каркаса модели, созданного в режиме «Аналитик»; данная роль предназначается для работников компаний, для которых разрабатывается ЕА-проект, хотя при наличии квалификации и желания они также могут работать в рамках любой из перечисленных выше ролей<sup>13</sup>.

В настоящий момент в новой версии пакета ОРГ-Мастер поддержаны только три режима – «Методолог», «Архитектор» и «Аналитик», причем первые два пока объединены в рамках одних и тех же функциональных возможностей ОРГ-Мастера и различаются только идеологически, в ролевом плане.

### **Заключение**

В настоящей работе представлены новые возможности языка моделирования ОРГ-Мастер:

- усиление типизации – разделение используемых типов на типы позиций классификатора, типы связей в проекциях и свободные атрибуты, а также введение обязательной типизации всех элементов классификатора и связей в проекциях с поддержкой, однако, отложенной типизации через стандартный тип по умолчанию «Не задан» и валидацию модели на наличие в ней таких типов;
- выделение различных ролей и сценариев использования языка моделирования ОРГ-Мастер – «Методолог», «Архитектор», «Аналитик», «Клиент». Все представленные в работе результаты реализованы в программном средстве ОРГ-Мастер 2.0.

Также кратко рассмотрена область ЕАМ и проведены параллели развития средств моделирования в этой области со средствами визуального моделирования в программной инженерии.

Отметим дальнейшие направления развития языка моделирования ОРГ-Мастер:

- развитие графических средств моделирования и тесная их интеграция с языком моделирования ОРГ-Мастера;
- разработка технологии автоматизированной генерации, настройки и сопровождения порталов на основе моделей ОРГ-Мастера;

---

<sup>12</sup> Подробнее см. <http://bigc.ru/government/modeling/>.

<sup>13</sup> кроме роли «Методолог» – пока не было случаев, чтобы клиенты участвовали в разработке методологии ОРГ-Мастера.

- развитие метаредактора для удобной модификации и адаптации опорных моделей и лежащих в их основе онтологий (в том числе графическое редактирование); интеграция со стандартами Semantic web;
- развитие механизма для работы со средствами накопления и передачи знаний (так называемые справочники и референтные модели);
- разработка механизма интеграции модели с внешними источниками данных (в том числе неструктурированными) – обычно знания любой организации распределены по различным хранилищам, документам и головам людей, и все эти знания нужно учитывать при создании и сопровождении ЕА.

### Литература

1. Данилин А., Слюсаренко А. Архитектура и стратегия. «Инь» и «Янь» информационных технологий предприятия // Интернет-Университет Информационных Технологий. – 2005. – 504 с.
2. Зиндер Е.З. Архитектура предприятия в контексте бизнес-реинжиниринга // Intelligent Enterprise. – 2008. – Ч. 1. – № 4. – С. 46; Ч. 2. – № 7. – С. 183.
3. Op'tLand M., Proper E., Waage M., Steghuis C. Enterprise Architecture: Creating Valueby Informed Governance. – Springer-Verlag, Berlin, Germany, 2009. – 146 p.
4. Калянов Г.Н. Архитектура предприятия и инструменты ее моделирования // Автоматизация в промышленности. – 2004. – № 7. – С. 9–12.
5. BucklS. 7 EAMtools: State-of-the-Art [Электронный ресурс]. – Режим доступа: [http://www.hpi.unipotsdam.de/hirschfeld/teaching/past/itua12/media/Itua12\\_07\\_EAMTools.pdf](http://www.hpi.unipotsdam.de/hirschfeld/teaching/past/itua12/media/Itua12_07_EAMTools.pdf), свободный. Яз. англ. (дата обращения 02.09.2013).
6. Ross D.T. Structured Analysis (SA): A Language for Communicating Ideas // IEEE Trans. Software Eng. – 1977. – V. 3. – № 1. – P. 16–34.
7. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – БХВ-Петербург, 2003. – 1104 с.
8. Bittler R.S. Magic Quadrant for Enterprise Architecture Tools. – Gartner, 2012. – G00234030. – 28 p.
9. Rational System Architect [Электронный ресурс]. – Режим доступа: <http://www-03.ibm.com/software/products/ru/ru/ratisystarch/>, свободный. Яз. англ. (дата обращения 02.09.2013).
10. MEGA Studio User Guide. – 2-nd Ed. – MEGA International, November 2011. – 114 p.
11. Артамонов И.В. Описание бизнес-процессов: вопросы стандартизации // Прикладная информатика. – 2011. – № 3 (32). – С. 20–28.
12. Jacob M., Jonkers H., Lankhorst M., Proper E. & Quartel D.A.C. ArchiMate 2.0 – Specification: The Open Group, 2012 [Электронный ресурс]. – Режим доступа: <http://pubs.opengroup.org/architecture/archimate2-doc/>, свободный. Яз. англ. (дата обращения 08.09.2013).
13. Каплан Р., Нортон Д. Стратегические карты. Трансформация нематериальных активов в материальные результаты. – М.: Олимп-Бизнес, 2012. – 446 с.
14. Остервальдер А., Пинье И. Построение бизнес-моделей. Настольная книга стратега и новатора: Пер. с англ. – М.: Альпина Паблишер, 2012. – 288 с.
15. Гаврилова Т.А., Лещева И.А., Кудрявцев Д.В. Использование моделей инженерии знаний для подготовки специалистов в области информационных технологий // Системное программирование. – Т. 7. – Вып. 1: Сб. статей / Под ред. А.Н.Терехова, Д.Ю.Булычева. – СПб: Изд-во СПбГУ, 2012. – С. 90–105.
16. ФаулерМ. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. – Символ-Плюс, 2011. – 192 с.
17. Kudryavtsev D., Gavrilova T. Diagrammatic Knowledge Modeling for Managers: Ontology-Based Approach. KEOD 2011 // Proceedings of the International Conference on Knowledge Engineering and Ontology Development. – 2011. – С. 386–389.
18. Самочадин А.В., Нурулин Ю.Р. Информационная поддержка публичных услуг. – СПб: БХВ-Петербург, 2013. – 160 с.
19. Dave A. Thomas: MDA: Revenge of the Modelers or UML Utopia? // IEEE Software. – 2004. – V. 21. – № 3. – P. 15–17.
20. Kelly S., Tolvanen J.-P. Domain-Specific Modeling: Enabling Full Code Generation. – John Wiley&Sons, 2008. – 340 p.
21. Кознов Д.В. Разработка и сопровождение DSM-решений на основе MSF // Системное программирование. – 2008. – Т. 3. – № 1. – С. 80–96.
22. Сорокин А.В., Кознов Д.В. Обзор Eclipse Modeling Project // Системное программирование. – 2010. – Т. 5. – № 1. – С. 6–32.
23. Гуров В.С., Мазин М.А., Шалыто А.А. UNIMOD – инструментальное средство для автоматного программирования // Научно-технический вестник СПбГУ ИТМО. – 2006. – № 7 (30). – С. 32–45.

24. Менеджмент по нотам: Технология построения эффективных компаний / Под ред. Л.Ю. Григорьева. – М.: Альпина Паблшер, 2012. – 694 с.
25. Grigoriev L., Kudryavtsev D. Non-diagrammatic method and multi-representation tool for integrated enterprise architecture and business process engineering // Proceedings of 15th IEEE Conference on Business Informatics (CBI 2013), 15–18 July. Vienna, Austria. – 2013. – P. 258–263.
26. Григорьев Л.Ю., Заблочный А.А., Кудрявцев Д.В. Технология наполнения баз знаний онтологического типа // Научно-технические ведомости СПбГПУ. Серия «Информатика. Телекоммуникации. Управление». – 2012. – № 3 (150). – С. 27–36.
27. Kudryavtsev D., Grigoriev L. Ontology-based business architecture engineering technology // The 10th International Conference on Intelligent Software Methodologies, Tools and Techniques, September 28–30. – 2011. – P. 233–252.
28. Григорьев Л., Кудрявцев Д. Организационное проектирование на основе онтологий // Научно-технические ведомости СПб ГПУ. Серия «Информатика. Телекоммуникации. Управление». – 2012. – № 1 (140). – С. 22–27.
29. Григорьев Л.Ю., Кознов Д.В., Кудрявцев Д.В. Обзор языка моделирования ОРГ-Мастер // Системное программирование. – Изд-во СПбГУ. – 2013. – Т. 8. – Вып. 1. – С. 5–34.
30. Бизнес-Инжиниринг Групп [Электронный ресурс]. – Режим доступа: <http://bigc.ru/instruments/bigmasterpro/bm/om/>, свободный. Яз. рус. (дата обращения 04.09.2013).
31. Verneaud M., Buckl S., Diaz-Fuentes A., et. al. Trends for Enterprise Architecture Management Tools Survey. – Technical report, 2012. – 144 p.
32. Гаврилова Т.А. Об одном подходе к онтологическому инжинирингу // Новости искусственного интеллекта. – 2005. – № 3. – С. 25–31.
33. Staab S., Studer R., eds. Handbook Ontologies. – Springer, 2009. – 811 p.
34. Terekhov A.N., Sokolov V.V. Implementation of the conformation of MSC and SDL Diagrams in the Real Technology // Programming and Computer Software. – 2007. – Т. 33. – № 1. – P. 24–33.
35. Кознов Д.В., Ольхович Л.Б. Визуальные языки проектов // Системное программирование. – 2005. – Т. 1. – № 0. – С. 148–167.
36. Вельдер С.Э., Шалыто А.А. Введение в верификацию автоматных программ на основе метода Model Checking // Научно-технический вестник СПбГУ ИТМО. – 2007. – № 42. – С. 33–48.
37. Кознов Д.В., Кудрявцев Д.В., Григорьев Л.Ю., Гагарский Р. Концепция средств графического моделирования в технологии ОРГ-Мастер // Программная инженерия. – 2014. – № 2. – С. 15–24.
38. Моррис Ч. Основы теории знаков / Под ред. Ю.С. Степанова // Семиотика. – М.: Радуга, 1983. – С. 37–90.
39. Кудрявцев Д.В., Григорьев Л.Ю., Кислова В.В., Жулин А.Б. Административное моделирование на основе онтологий // Вопросы государственного и муниципального управления. – 2009. – № 1. – С. 157–169.
40. Кудрявцев Д.В. Разработка моделей и методов обработки знаний в области организационного проектирования на основе онтологий: Автореф. Дисс. канд. техн. наук: 05.13.01; 05.13.11. – СПб: СПбГПУ, 2009. – 20 с.
41. Костырко А., Кудрявцев Д., Григорьев Л., Кислова В., Жулин А., Сиятуллина Л., Ермаков Р. Моделирование комплексов городского хозяйства для системного развития ИКТ города // Сборник трудов конференции «Инженерия знаний и технологии семантического веба–2012», 1–9 октября 2012. – СПб: СПбГУ ИТМО, 2012. – С. 81–88.
42. Кознов Д.В. Предметно-ориентированное визуальное решение для сбора и упорядочивания информации при разработке информационной Web-системы. Компьютерные инструменты в образовании, – 2013. – № 5 – С. 15–27.

- Кудрявцев Дмитрий Вячеславович** – Россия, Санкт-Петербург, Санкт-Петербургский государственный политехнический университет, доцент; компания «Бизнес Инжиниринг Групп», ведущий консультант; кандидат технических наук, доцент, [dmitry.ku@gmail.com](mailto:dmitry.ku@gmail.com)
- Кознов Дмитрий Владимирович** – Россия, Санкт-Петербург, Санкт-Петербургский государственный университет, кандидат физ.-мат. наук, доцент, [dkoznov@yandex.ru](mailto:dkoznov@yandex.ru)
- Григорьев Лев Юрьевич** – Россия, Санкт-Петербург, компания «Бизнес Инжиниринг Групп», генеральный директор, [griglev@gmail.com](mailto:griglev@gmail.com)