Л.Н. Лядова[1]

Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал)

LyadovaLN@hse.perm.ru

МЕТАМОДЕЛИРОВАНИЕ КАК ОСНОВА СРЕДСТВ ОПЕРАТИВНОЙ РАЗРАБОТКИ ПРОФЕССИОНАЛЬНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Введение

Развитие информационных технологий (ИТ), расширение их возможностей привело к тому, что средства ИТ находят применение в новых областях. Возросшие потребности в создании новых информационных систем (ИС) для различных предметных областей, автоматизирующих всё новые виды деятельности, уже не удовлетворяются при использовании традиционных средств разработки ИС.

Кроме того, использование ИС в динамично развивающихся организациях, с постоянно меняющимися потребностями автоматизации бизнес-процессов, а также различных категорий пользователей ИС обусловило необходимость в создании новых технологий, позволяющих осуществлять динамическую адаптацию информационных систем к меняющимся условиям функционирования и требованиям бизнес-систем. Адаптируемость становится одним из ключевых свойств ИС, гарантирующим эффективность вложений в создание и внедрение ИС, определяющим их живучесть.

Создание ИС с использованием современных инструментальных средств основано на разработке различных моделей, описывающих предметную область ИС, определяющих структуры данных и алгоритмы функционирования. При реализации *модельно-ориентированного подхода* к разработке ИС (или *Model-Centric Software Development* – MCSD) модели становятся центральным звеном на всех этапах созда-

[1] Работа выполнена при поддержке РФФИ (проект № 12-07-00763-а)

ния ИС. Однако максимальная гибкость системы достигается при использовании различных технологий, основанных на моделировании, на применении моделей не только в процессе разработки ИС, но и в ходе эксплуатации: система функционирует в режиме интерпретации построенных моделей, управляющих её поведением, настройка ИС осуществляется через изменение моделей.

Как отмечается, эксперты, специалисты в соответствующих предметных областях должны принимать активное участие не только в разработке моделей на этапе анализа предметной области и проектирования системы, но и в её настройке в ходе эксплуатации, адаптируя систему к своим потребностям. Это возможно только при использовании соответствующих инструментальных средств, основанных на *предметно-ориентированном моделировании* (DSM – *Domain Specific Modeling*), а также специальных *предметно-ориентированных языков* (DSL – *Domain Specific Languages*) или *предметно-ориентированных языков моделирования* (DSML – *Domain Specific Modeling Languages*), позволяющих разработчикам и пользователям работать с формальными моделями в привычных для них терминах.

Существует «специализация» разработчиков ИС: одни занимаются проектированием структур данных, другие – анализом и моделированием бизнес-процессов и т.д. На разных этапах разработки ИС используются различные средства формализации результатов работы. Крупные корпоративные системы автоматизируют различные виды деятельности предприятий, используются специалистами во многих областях. Таким образом, в одной системе должны «сосуществовать» различные модели, созданные с использованием разных языков. Здесь речь идёт уже об использовании нескольких предметно-ориентированных языков, предназначенных для профессионально-ориентированной настройки системы. Традиционные подходы к созданию ИС, основанные на использовании для создания моделей одного языка моделирования (например, UML), и реализующие их технологии не позволяют решить эти задачи.

Развивается относительно новый подход к разработке информационных систем, основанный на *мультимоделировании* (DSMM – *Domain-Specific Multimodeling*), парадигме *программирования, ориентированного на языки* (*Language-Oriented Programming* – LOP), разработка системы с использованием нескольких предметно-ориентированных языков. Этот подход существенно повышает производительность при разработке ИС, обеспечивает преемственность при переходе от одного этапа разработки к другому, облегчает её сопровождение, позволяет привлекать к работе экспертов в соответствую-

щих предметных областях на всех этапах жизненного цикла ИС. Однако при этом встаёт проблема преобразования и согласования различных моделей, созданных с использованием различных языков – проблема трансформации моделей (в традиционных CASE-системах, например, на основе построенных визуальных моделей генерируется программный код, но не всегда обеспечивается обратная связь – при внесении изменений в код визуальные модели не меняются). Определить правила, в соответствии с которыми выполняются преобразования моделей, предлагается также с помощью специальных языков – *языков трансформации моделей* (*Model Transformation Language* – MTL).

В данной статье рассматривается подход к созданию инструментальных средств, предназначенных для разработки динамически адаптируемых профессионально-ориентированных систем, основанный на метамоделировании и интеграции CASE-средств и языковых инструментариев, создании DSM-платформ.

Моделирование и технологии программной инженерии

При использовании традиционных подходов к созданию ИС требования, спецификации, используемые при разработке, фиксируются в документах, которые могут быть неполными, неоднозначными, просто неправильно понятыми разработчиками. Следствием являются ошибки в реализации системы, устранение которых требует времени, является затратным. Применение формальных моделей на всех стадиях проекта позволяет преодолеть эти трудности. Разработчики получают возможность выполнять валидацию моделей, уточнять их, осуществлять тестирование на различных этапах реализации проекта. Однако при этом требуется обеспечить как разработчиков – ИТ-специалистов, так и заказчиков – экспертов в предметных областях – средствами, которые позволяли бы строить формальные модели, понятные и однозначно интерпретируемые всеми участниками проекта.

Модель – это абстрактное описание системы (в частности, программного обеспечения), в котором отражены существенные с точки зрения решаемых её разработчиком задач характеристики, особенности функционирования. В модели фиксируется информация о системе в форме, пригодной для понимания и интерпретаций людьми и обработки инструментальными средствами, применяемыми на различных этапах жизненного цикла ИС. Модель создаётся с использованием определённых *языков моделирования* (обычно используются визуальные средства моделирования, основанные на совокупности графических

элементов, которые применяются в модели и могут быть легко интерпретированы людьми).

Метамодель описывает понятия и нотации, используемые в модели. Таким образом, метамодель можно рассматривать как язык, используемый для разработки моделей. Предметно-ориентированные языки моделирования позволяют при создании моделей учитывать специфику решаемых задач, описываемой предметной области.

Вся информация, представленная в модели, фиксируется в виде *метаданных*, которые могут быть обработаны применяемыми разработчиками инструментами.

В настоящее время основной тенденцией реализации проектов в области ИТ является применение технологий, базирующихся на подходе, называемом «*разработка, управляемая моделями*» (MDE – *Model Driven Engineering*). Под MDE понимают подход к созданию программного обеспечения (ПО), когда модели становятся основными артефактами на всех этапах разработки ИС: на основе моделей генерируется программный код, определяются структуры данных и создаётся и настраивается пользовательский интерфейс системы. Используют также понятия «разработка, основанная на моделях» (MBE, MBSE – *Model Based System Engineering* или *Model Based Software Engineering*), «проектирование, управляемое моделями» (MDD – *Model-Driven Design*). Во многих публикациях отмечается близость этих понятий. В русскоязычных публикациях устоявшихся терминов ещё нет (используется, например, аббревиатура МУР (модель-управляемая разработка) и др.).

Технологии MDE сфокусированы на архитектуре системы и соответствующих средствах автоматизации, позволяют работать на более высоких уровнях абстракции при проектировании программного обеспечения, а это позволяет строить более простые модели, сосредоточить усилия на предметной области, поставленных задачах. Внедрение в модели семантики, используемой во время выполнения, позволяет поднять общий уровень автоматизации управления жизненным циклом ИС.

Консорциум OMG (Object Management Group) разработал набор стандартов, названных MDA (*Model-Driven Architecture*), ставших основанием для реализации такого подхода, сфокусированного на архитектуре. Основная цель MDA (или MOA – модель-ориентированная архитектура) – это минимизация затрат на создание информационных систем, программного обеспечения, связанных с привязкой ПО к конкретным системным платформам и программным инфраструктурам. Для этого вводятся более высокие уровни описаний программной си-

стемы и её предметной области, на основе которых создаются структуры данных, выполняется генерация программного кода системы, осуществляется настройка её параметров и т.п. для различных платформ.

Основная идея MDA – отделение абстрактной архитектуры системы от архитектуры конкретной платформы. *Платформенно-независимые модели* (*Platform Independent Model* – PIM) описывают структуру и функции системы, но не особенности ее реализации. Модели PIM содержат достаточно информации, чтобы перейти от них к *платформенно-зависимым моделям* (*Platform Specific Model* – PSM), которые могут включать исходный код, конфигурационные файлы и др. элементы, специфичные для целевой платформы.

На приведённом ниже рис. 1 показана наиболее общая связь между описанными выше понятиями.

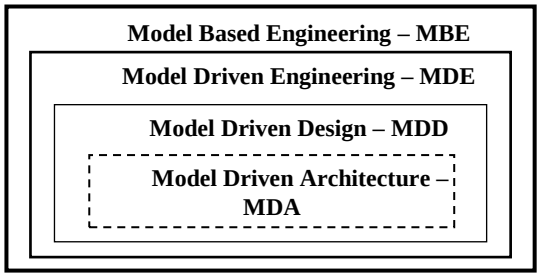

Рис. 1. Технологии, основанные на использовании моделей

Классический пример использования моделей и их разделения – моделирование баз данных, когда на основе одной и той же концептуальной модели данных можно получить и поддерживать несколько физических моделей для различных СУБД.

Стандарты MDA – это наилучший путь к управлению сложностью проекта, достижению высокого уровня переносимости решений, переиспользования результатов и значительному сокращению трудоёмкости реализации программных проектов, интеграции этапов анализа и реализации. Хотя при использовании MDA создаётся две модели вместо одной, MDA упрощает разработку системы на основе абстракций, её реализацию для различных целевых платформ. MDA позволяет определить *трансформации*, которые дают возможность отобразить модели PIM на PSM, автоматизировать генерацию платформенно-зависимой модели по платформенно-независимой – этот процесс в значительной степени формализован.

Кроме того, уменьшается риск ошибок проектирования: на относительно простой и полностью основанной на требованиях заказчика платформенно-независимой модели проще находить ошибки и исправлять их, в то время как традиционные модели «загромождены» деталями реализации, которые мешают восприятию и пониманию модели. Разделение PIM и PSM упрощает также разработку документации, интеграцию систем, создание гетерогенных систем и т.д. Таким образом, уже не потребуется создавать модель PSM вручную, и процесс разработки существенно ускорится, так как значительное количество элементов, специфичных для технологии реализации будет вноситься в эту модель автоматически.

Описание перехода от PIM к PSM может быть произведено и отлажено всего один раз для каждой технологии, а потом может использоваться во всех проектах, использующих данную технологию.

При использовании автоматизированного перехода от платформенно-независимых моделей к платформенно-зависимой системе *цикл разработки ИС с использованием MDA* включает следующие этапы:

1. Постановка задачи, создание сценариев использования ИС и списка формальных требований к ней.
2. Создание PIM.
3. Автоматизированная трансформация PIM в PSM с использованием стандартного или заранее разработанного описания трансформации для данной предметной области и выбранной технологии реализации.
4. Уточнение моделей, их доработка, добавление необходимых деталей (при модификации моделей должно поддерживаться соответствие между ними).
5. Автоматизированная генерация кода по детальной PSM.
6. Ручная доработка кода, компиляция.

Переход от PSM к программному коду достаточно хорошо разработан, поддерживается всеми CASE-системами (в частности, для работы с UML при создании ИС для наиболее широко используемых платформ, языков и технологий). Автоматизированный же переход от PIM к PSM ещё недостаточно проработан.

Отмечается, что не всякий способ задания трансформации моделей может быть эффективно применён в разработке ПО с использованием MDA. Формулируется ряд требований, которым должны соответствовать средства трансформации моделей:

- *Формальность и полнота*: язык описания трансформаций должен обеспечивать возможность определения любой необходимой трансформации для любой платформы, при этом описание

трансформации должно быть формализованным, чтобы была возможность её автоматического выполнения.

- *Универсальность правил трансформации*: необходимо, чтобы язык описания трансформации позволял задавать трансформации, применимые для множества проектов, а не только для одной конкретной модели (должны быть предусмотрены средства для настройки и параметризации трансформации).
- *Поддержка целостности при модификации*: после трансформации как исходные модели, так и модели, созданные в процессе трансформации, могут быть изменены, при этом должно поддерживаться соответствие между этими моделями, достигнутое в процессе трансформации.
- *Наглядность правил трансформации*: описание трансформации должно быть понятно не только создавшему его разработчику-программисту, но и тому, кто будет его использовать, т.е. язык описания трансформации должен иметь легко читаемую нотацию, допускать эффективное структурирование описания трансформации.
- *Взаимосвязанная трансформация нескольких моделей*: язык трансформации должен предусматривать наличие более чем одной исходной и генерируемой модели.

MDA ориентирована на использование моделей, основанных на UML, а также генерации кода приложений. Существуют также проекты, в которых применяются другие подходы к созданию моделей и языков моделирования.

В настоящее время не существует единого универсального визуального языка моделирования: активно используются на практике такие языки визуального моделирования, как SDL и MSC – для моделирования телекоммуникационных систем, IDEF, DFD, EPC, BPEL и BPML – для моделирования бизнес-процессов на разных уровнях и т.п.

Язык UML часто рассматривается как стандарт языка моделирования, однако этот язык имеет ряд существенных недостатков:

- диаграммы UML сложны для понимания не только для экспертов, которые принимают участие в разработке системы, но в некоторых случаях даже и для профессиональных программистов;
- диаграммы UML не могут адекватно представить понятия предметной области, т.к. работа ведётся в терминах «класс», «ассоциация», «агрегация» и т.п., а не в терминах этой предметной области.

Решение о выборе языка полностью лежит на разработчиках системы, выбирающих средства разработки. «Универсальные» языки могут стать основой для разработки предметно-ориентированных языков, «заточенных» для моделирования систем в определённой предметной области. Язык, используемый для создания других языков, будем называть *метаязыком*. Процесс создания модели может стать *итеративным*: создав некоторый язык, мы можем использовать его как метаязык для создания другого языка, который, в свою очередь, также может быть использован как метаязык.

Потребности разработчиков в более удобных и мощных средствах моделирования предметных областей привели к появлению нового направления в создании инструментальных средств разработки ИС – *семантического моделирования*. Основная цель исследований в области семантического моделирования – сделать инструментальные средства разработки более «разумными», позволяющими представить более полно «смысл» описываемых процессов и данных, реализовать систематический подход к проектированию ИС.

Общая *схема проектирования ИС* при использовании семантического моделирования включает следующие два этапа:

1) семантическое моделирование для создания концептуальной модели предметной области системы в терминах этой области;

2) отражение спецификаций созданной модели предметной области на возможности конкретной платформы.

Для семантического моделирования предметных областей широко используется *онтологический подход*, а средства создания и редактирования онтологий становятся основой для создания инструментальных средств разработки ИС.

Большинство инструментальных средств создания ИС ориентированы в первую очередь на *статическое моделирование* предметных областей ИС, что не учитывает их динамику. Практикуется «долгая и тщательная», требующая больших трудозатрат разработка моделей, которые затем фиксируются и реализуются в информационной системе с использованием соответствующих инструментальных средств. Созданная система допускает ввод и изменение данных, отражающих состояние предметной области, но не изменение модели. Однако такие изменения могут оказаться необходимыми в ходе эксплуатации ИС при изменении условий функционирования или требований бизнес-процессов и пользователей.

Таким образом, создаваемые инструментальные средства должны решать некоторые общие задачи, удовлетворять общим требованиям:

– Системы должны *основываться на моделях, семантически бо-*

лее полных, чем базовая модели, используемые в традиционных ИС для реализации функций системы (например, реляционная модель). Новые модели должны опираться на все преимущества существующих базовых моделей, быть их естественным расширением.

– Должна *обеспечиваться динамика эксплуатации ИС*: должны учитываться и поддерживаться изменчивость и расширяемость моделей предметных областей, а также темпоральная изменчивость данных в период эксплуатации ИС. Это требование позволяет осуществлять адаптацию ИС, их развитие в процессе функционирования системы. Семантические модели, созданные в терминах предметных областей, позволяют вести непрерывную разработку ИС с участием специалистов в прикладных областях, пользователей ИС.

– Должна быть создана *интегрированная инструментальная среда* – система должна совмещать инструменты для построения моделей и обработки данных, документов в рамках единого универсального средства. Должна быть обеспечена возможность эффективной реализации как *оперативной, так и аналитической обработки данных* в ИС, работы как со *структурированными данными, так и с неструктурированной информацией*, получаемой из различных гетерогенных источников.

Реализация перечисленных требований позволит значительно облегчить и ускорить разработку ИС, упростить их сопровождение.

В общем случае при создании и эксплуатации ИС разработчики и пользователи имеют дело с набором различных инструментов: собственно инструментарий СУБД, среда проектирования БД ИС и среда разработки приложений, инструментальные средства аналитической обработки данных и генераторы отчётов, средства поиска информации во внешних источниках, анализа полученных результатов. Эти средства объединяются в рамках CASE-систем и языковых инструментариев, BI-инструментария. Для построения и использования ИС необходимо владеть всеми этими инструментами.

Кроме того, система должна включать не только традиционные средства управления документами, генераторы отчётов и пр., но и средства интеллектуального анализа документов, которые могут стать основой для реализации средств автоматизации создания моделей [1].

Решение – интеграция различных инструментариев в одной системе на основе использования общих моделей [2]. Ставится задача создания DSM-платформы, объединяющей инструментальные средства различного назначения, ориентированные как на решение задач создания ИС и их динамической настройки, так и на работу с ними различных категорий пользователей.

Описываемый инструментарий ориентирован на создание распределённых информационных систем, обеспечивает их интеграцию и взаимодействие, синхронизацию данных и моделей, с которыми работают пользователи. Обеспечивается возможность создания систем с удалённым доступом на основе Web-технологий. Инструментарий предусматривает использование компонентов автоматического документирования процесса разработки и настройки ИС. Кроме описанных выше средств должны быть реализованы средства обеспечения безопасности ИС, многоуровневой защиты доступа к её ресурсам.

Основа DSM-платформ – средства предметно-ориентированного моделирования (Domain Specific Modeling, DSM), языковые инструментарии, предназначенные для создания предметно-ориентированных языков (Domain Specific Languages, DSL).

Обзор предметно-ориентированных языков и DSM-платформ приведён в работах [3, 4].

Архитектура интегрированной CASE-системы

Языковой инструментарий, или *DSM-платформа*, – это инструментальное программное обеспечение, предназначенное для поддержки разработки и сопровождения языков предметной области.

Разрабатываемая платформа (рис. 2) основана на *метамоделировании* [2, 5, 6]. В системе создаётся *несколько уровней моделей*. Данные, хранящиеся в БД системы, представляют собой *модель состояния* предметной области ИС. Их описание, обеспечивающее интерпретацию данных, – это *метамодель*, т.е. модель более высокого уровня, создаваемая с помощью CASE-средства. Для разработки этой модели применяется специальный формальный язык, позволяющий работать в терминах соответствующей предметной области, – *мета-метамодель*. Таким образом, строятся метамодели, которые определяют предметно-зависимые языки моделирования.

Процесс построения языков является *итерационным*: на основе одних языков создаются новые языки. Языки, используемые для создания новых языков, выступают в качестве *метаязыков*. Для создания этих языков также необходимы соответствующие инструментальные средства («мета-CASE-средства») – языковой инструментарий, называемый также DSL-инструментарием. Мета-CASE- и CASE-средства объединяются в одной CASE-системе, интегрирующей «традиционные» CASE-средства и DSL-инструментарий.

Средства формализации знаний об ИС и автоматизируемых ею процессах позволяют для каждой предметной области, каждого вида деятельности, поддерживаемого ИС, разрабатывать свои предметно-

ориентированные языки, «понятные» пользователям – специалистам в этой области, разработчикам.

Ещё одна задача – *снижение трудоёмкости разработки и настройки ИС через автоматизацию построения моделей*. В процессе эксплуатации системы могут появляться новые объекты и свойства, существенные для данной предметной области. Таким образом, встаёт задача динамического изменения построенных моделей.

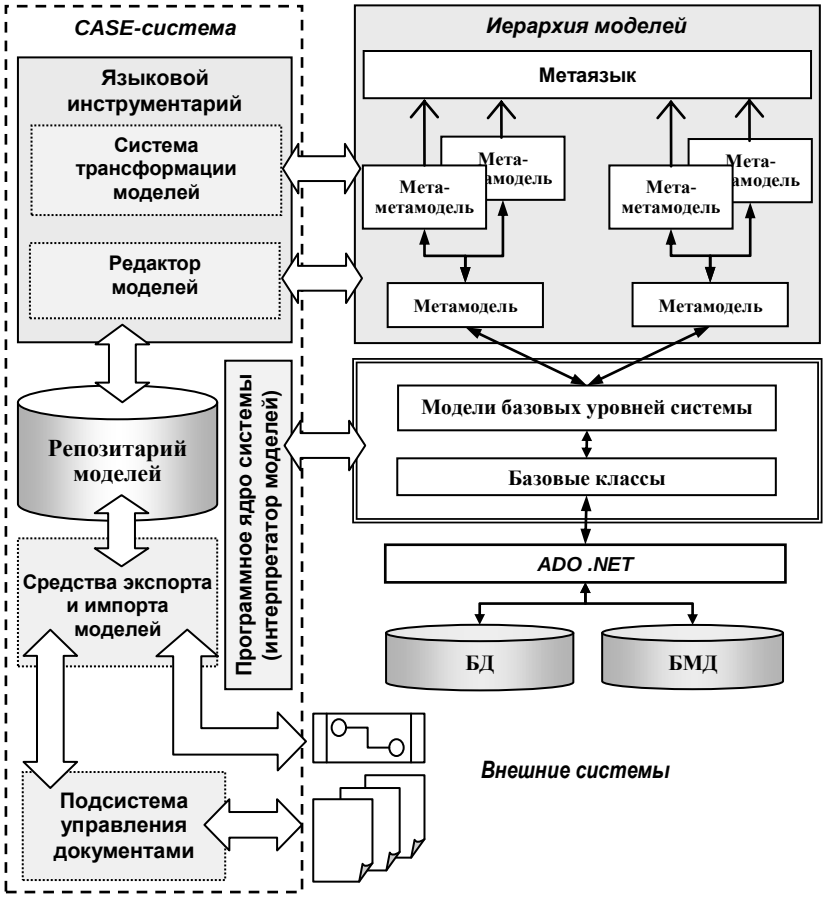

Рис. 2. Упрощённая структура информационной системы, основанной на DSM-платформе с интерпретацией моделей

Изменения в модели вносятся на основе анализа информации о предметной области, получаемой из различных источников (программной документации, нормативных и распорядительных документов и пр.). Важнейшей проблемой при решении этих задач является переход от текстовых описаний объектов и процессов предметной области к формальным нотациям. Решение этой задачи связано с поиском и анализом документов, выделением из них нужной информации. Предлагается решать эту проблему на основе *онтологического подхода* [1]. Разработанные при создании ИС модели являются основой для поиска документов, относящихся к предметной области ИС, их анализа, а результаты анализа найденных документов служат основой для внесения изменений в существующие модели. Таким образом создаётся «интеллектуальная» система с высокой степенью обратной связи.

Заключение

В работе представлен проект DSM-платформы, которая поддерживает все этапы жизненного цикла ИС от анализа предметной области и разработки предметно-ориентированных языков моделирования до создания на их основе моделей, которые могут использоваться в конкретной реализации ИС, функционирующей в режиме интерпретации моделей, либо могут быть экспортированы в другую систему.

Модификация моделей может производиться на любом этапе создания и функционирования ИС. При этом после внесения изменений в метамодель система автоматически внесёт все необходимые изменения в модели, созданные на основе этой метамодели.

Для того чтобы данный подход оказался применимым на практике, необходимо чтобы мета-метамодель была максимально выразительной, что обеспечивает высокую степень гибкости ИС, возможности её настройки на потребности пользователей. Фактически создание системы сводится к разработке предметно-ориентированных языков. Использование DSL и языковых инструментариев позволяет существенно упростить процесс создания моделей. В разработке могут принимать активное участие эксперты – специалисты в различных предметных областях. Выразительность языков и производительность созданных на их основе систем зависит от свойств моделей базовых уровней, выбора математического аппарата для описания формальных свойств языков, алгоритмов их интерпретации.

При работе в режиме интерпретации моделей реализация нескольких уровней моделирования в ИС приводит к потере производи-

тельности, но позволяет построить систему с уникальными возможностями динамической адаптации.

Кроме того, потеря производительности практически не растёт с ростом объёма обрабатываемых данных, т.к. сложность моделей зависит только от сложности предметных областей, а не от объёма данных. Ещё одно средство повышения производительности – реализация кэширования при интерпретации моделей.

Библиографический список

1. *Ланин В.В., Лядова Л.Н.* Управление документами в динамически адаптируемых системах, основанных на метамоделировании // Труды Конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'10» Научное издание в 4-х томах. Т. 1. – М.: Физматлит, 2010. С. 510-518.
2. *Лядова Л.Н., Сухов А.О.* Визуальные языки и языковые инструментарии: методы и средства реализации // Труды Конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'10» Научное издание в 4-х томах. Т. 1. – М.: Физматлит, 2010. С. 374-382.
3. *Сухов А.О.* Классификация предметно-ориентированных языков и языковых инструментариев // Математика программных систем: Межвуз. сб. научн. тр. / Перм. гос. нац. исслед. ун-т. Пермь, 2012. С. 74-83.
4. *Сухов А.О.* Сравнение систем разработки визуальных предметно-ориентированных языков // Математика программных систем: Межвуз. сб. научн. тр. / Перм. гос. нац. исслед. ун-т. Пермь, 2012. С. 84-111.
5. *Сухов А.О.* Среда разработки визуальных предметно-ориентированных языков моделирования // Математика программных систем: Межвуз. сб. научн. тр. / Перм. ун-т. Пермь, 2008. С. 84-94.
6. *Sukhov A.O., Lyadova L.N.* MetaLanguage: a Tool for Creating Visual Domain-Specific Modeling Languages // Proceedings of the 6th Spring/Summer Young Researchers' Colloquium on Software Engineering, SYRCoSE 2012. – Perm, 2012. – P. 42-53.