

ПРОЕКТИРОВАНИЕ ИНТЕГРИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ
С ИСПОЛЬЗОВАНИЕМ DSM-ПЛАТФОРМЫDESIGNING OF INTEGRATED INFORMATIONAL SYSTEMS
WITH DSM-PLATFORM

Аннотация: Описываются средства DSM-платформы MetaLanguage, позволяющие разрабатывать предметно-ориентированные языки и модели для различных областей, выполнять трансформации моделей при интеграции систем. Приведены примеры.

Abstract: The tools of MetaLanguage DSM-platform, allowing to design domain specific languages and models for different domains, to transform models at systems integration, are described. Examples are presented.

Ключевые слова: моделирование, предметно-ориентированные языки, языковой инструментарий, трансформации.

Keywords: modeling, domain-specific languages, language workbench, transformations.

Введение

В настоящее время все более актуальной становится проблема повышения оперативности разработки новых информационных и аналитических систем и настройки существующих систем на решение новых задач, их адаптации к меняющимся условиям эксплуатации и потребностям пользователей. Решить эти задачи можно только на основе применения моделей, представляющих формализованные описания предметных областей, требований к разрабатываемым информационным системам (ИС). Использование моделей не только на этапе анализа предметной области и формализации требований к создаваемым ИС, но и в течение всего их жизненного цикла позволяет создать гибкие системы с максимальными возможностями их адаптации. В связи с этим все более востребованными становятся инструментальные средства, основанные на применении *модельно-ориентированного подхода* (МОП) к созданию ИС. Такие средства позволяют снизить трудоемкость создания и сопровождения систем за счет «переиспользования» ранее описанных моделей для генерации и настройки программного обеспечения (ПО) ИС.

Модельно-ориентированный подход к разработке ПО предполагает использование языков моделирования, с помощью которых производится построение моделей. При этом чаще используются *визуальные языки*, поскольку графические модели обладают большей наглядностью и «понятностью» не только для программистов, но и для экспертов в соответствующих предметных областях, конечных пользователей.

При проектировании информационных и аналитических систем выбор используемых CASE-средств и VI-платформ диктует необходимость применения соответствующих языков моделирования. При интеграции систем, созданных различными разработчиками, приходится перерабатывать модели, описанные с помощью различных языков, выполнять их *трансформации*, чтобы обеспечить возможность их интеграции при создании модели интегрированной системы.

Ещё одна задача – обеспечить участие в процессе разработки системы экспертов, специалистов в различных предметных областях. Это требует создания соответствующих условий: разрабатываемые модели должны быть понятны не только специалистам в области информационных технологий, системным аналитикам и программистам, но и экспертам, которые должны работать с описаниями, в которых использованы привычные для них понятия, термины соответствующей предметной области. Эта возможность обеспечивается, если для создания моделей применяются *предметно-ориентированные языки*.

Для разработки предметно-ориентированных языков и преобразования моделей используются специальные программные средства – языковые инструментарии, DSM-платформы.

Языки моделирования и DSM-платформы

В процессе разработки сложных систем создаётся множество моделей, используемых для решения различных задач, описывающих целевые системы с различных точек зрения, на различных уровнях абстракции. Таким образом, создаётся *иерархия моделей* [1], в которой модели описываются с помощью языков моделирования (*метамodelей*), языки моделирования описываются с помощью *метаязыков*.

Существуют реализации МОП, которые используют *языки моделирования общего назначения* для построения моделей. Так, язык моделирования общего назначения UML (Unified Modeling Language) вместе со стандартом MOF (Meta-Object Facility) формирует основу для концепции MDA (Model-Driven Architecture). Но языки моделирования общего назначения часто сложны для понимания не только для экспертов в конкретной предметной области, которые принимают участие в процессе создания системы, но и, в некоторых случаях, даже для профессиональных разработчиков. Кроме того, иногда бывает затруднительно адекватно выразить понятия предметной области, которыми оперируют пользователи ИС, с помощью языков общего назначения. Именно поэтому в модельно-ориентированной разработке ПО все более широко применяются визуальные *предметно-ориентированные языки моделирования* (*Domain-Specific Modeling Languages – DSML, Domain-Specific Languages – DSL*), предназначенные для решения определенного класса задач в конкретной предметной области. Предметно-ориентированные языки просты в применении и понятны различным категориям специалистов, поскольку они оперируют привычными для них терминами предметной области. При использовании DSL с задачей модификации моделей с помощью соответствующих инструментальных средств способны справиться даже пользователи, не являющиеся профессиональными разработчиками, эксперты в конкретной предметной области, которые могут стать активными участниками процесса создания и модификации моделей и, таким образом, участниками разработки и сопровождения ИС.

Для поддержки процесса создания DSL используется специальный вид ПО, получивший название *языковой инструментарий*, или *DSM-платформа* (DSM – *Domain Specific Modeling*). Существуют различные средства создания визуальных DSL с возможностью задания собственных графических нотаций: MetaEdit+ [2, 3], DSL Tools [4, 5], Eclipse GMF [6, 7], QReal [8, 9] и др. DSM-платформы позволяют разрабатывать языки и модели, а также осуществлять их преобразования, связанные с необходимостью перехода от одних нотаций к другим, – *трансформации моделей* (перевод моделей, описанных с использованием одного языка, на другой язык).

По направлению преобразования моделей различают вертикальные и горизонтальные трансформации. *Вертикальные трансформации* управляют ходом преобразования моделей при переходе от одного уровня иерархии моделей к другому, например, при отображении элементов метамодели на элементы модели. *Горизонтальная трансформация* – это преобразование, при котором исходная и целевая модель принадлежат одному уровню иерархии. Примером горизонтальной трансформации является преобразование модели, созданной с использованием одного языка моделирования, в модель, описанную на другом языке. Средства трансформации моделей позволяют снизить трудоёмкость создания ИС, обеспечивая, с одной стороны, возможность автоматической генерации структур данных и программного кода, а с другой – возможность экспорта и импорта моделей.

Существуют различные подходы к трансформации моделей: например, системы AGG, GReAT, VIATRA используют для выполнения трансформаций правила переписывания графов (graph rewriting), подход MTBE основывается на методе программирования по образцу [10] и т.д. Все реализованные в настоящее время подходы обладают теми или иными ограничениями, осложняющими их применение для описания трансформаций (в частности,

предъявляются высокие требования к квалификации пользователей). Таким образом, одной из основных задач, решаемых при создании языковых инструментариев, становится задача разработки средств трансформации, снимающих эти ограничения. Реализация таких средств существенно расширяет возможности применения DSM-платформ, делает их более доступными для использования различными категориями специалистов при разработке систем различного назначения и качестве основы для интеграции информационных и аналитических систем [11].

Создание и трансформация моделей с использованием DSM-платформы MetaLanguage

Система MetaLanguage – это *языковой инструментарий*, предназначенный для создания визуальных предметно-ориентированных языков и моделей с их использованием, а также выполнения трансформаций созданных моделей.

Процесс разработки моделей в MetaLanguage включает несколько этапов [12]:

- *разработка предметно-ориентированного языка* (метамодели) с помощью графического редактора моделей;
- *разработка моделей* с использованием созданного языка, их валидация.

При создании метамодели в первую очередь определяются базовые конструкции нового языка: сущности метамодели, отношения между ними, ограничения, налагаемые на сущности и отношения. После построения метамодели разработчик получает в распоряжение расширяемый, динамически настраиваемый визуальный язык моделирования. В процессе создания DSL первого уровня используются *базовые конструкции* MetaLanguage: сущность, отношение, ограничение. При этом созданные метамодели могут использоваться в качестве метаязыков для создания новых языков. Таким образом, строится *иерархия моделей*.

Используя полученный DSL, пользователь может создавать модели, содержащие объекты, описывающие конкретные сущности предметной области и связи между ними. При разработке модели необходимо также проверить, удовлетворяет ли она ограничениям, которые были на неё наложены, – выполнить валидацию созданной модели.

При внесении изменений в метамодель (в описание DSL) система автоматически вносит все необходимые изменения в модели, созданные с применением этой метамодели (предметно-ориентированного языка).

Для автоматизации преобразований моделей в MetaLanguage используется компонент *трансформации*.

Вертикальная трансформация – это преобразование модели, описанной на одном уровне иерархии, в модель, представленную на другом уровне. Трансформация модели вышестоящего уровня иерархии в модель нижестоящего уровня в MetaLanguage соответствует операции *создания модели*. Обратная трансформация позволяет производить *интерпретацию модели* нижестоящего уровня, определять типы ее элементов, выполнять различные операции над этой моделью. Алгоритмы выполнения соответствующих операций позволяют автоматически поддерживать согласованность моделей различных уровней.

Для того чтобы выполнить *горизонтальную трансформацию* моделей, необходимо задать соответствующие правила трансформации, которые описываются с использованием соответствующих метамodelей. Правила определяют соответствие между конструкциями языков (метамodelей). Описание трансформаций должно производиться с использованием визуальных языков, понятных различным категориям специалистов. В MetaLanguage существует возможность задания горизонтальных трансформаций типа «модель-модель» и «модель-текст».

В системе MetaLanguage трансформация моделей осуществляется на основе алгебраического подхода с одинарным выталкиванием, основанного на графовых грамматиках, в реализацию которого внесён ряд модификаций [13, 14], обеспечивающих возможность описания трансформации атрибутов элементов метамodelей, многоуровневого описания метамodelей из левой и правой части продукционного правила, описания правил трансформаций на одном уровне иерархии, а их применения – на другом.

Задача проектирования комплексной интегрированной информационно-аналитической системы

В последнее время большое внимание уделяется методам и средствам интеграции информационных систем, позволяющим объединить множество различных подсистем, расширить их функциональные возможности, повысить эффективность их использования. Например, для решения задач мониторинга деятельности промышленных предприятий имеется программное обеспечение, которое автоматизирует лишь некоторые операции процесса мониторинга, реализуемые отдельными подсистемами, значительная часть работ не автоматизирована. Таким образом, существует потребность в создании единой организационно-технической системы формирования и ведения сводной информации об объектах мониторинга, получаемой из распределённых, обособленных источников. Комплексная система должна объединить существующие подсистемы, что позволит решить проблему взаимодействия подсистем, объединения их функциональности (основное внимание уделяется функциональной интеграции) [15].

Перспективным подходом к решению вопросов функциональной интеграции систем является проектирование, управляемое моделями (MDE, Model-Driven Engineering), предполагающее систематическое использование моделей на протяжении всего жизненного цикла ИС: на основе моделей определяются структуры данных, генерируется программный код, создаётся и настраивается пользовательский интерфейс системы. Процесс создания программного продукта представляется в виде цепочки трансформаций исходной модели в готовую к использованию информационную систему. В основе MDE – концепция DSM (Domain Specific Modeling).

При разработке модели комплексной информационно-аналитической системы на основе существующих подсистем решаются задачи интеграции моделей подсистем, разработанных для различных предметных областей с использованием различных визуальных языков. Кроме того, различные категории специалистов требуют представления модели на различных языках, её описания с использованием различных графических нотаций, разных типов диаграмм.

Для снижения трудоёмкости решения поставленных задач предлагается использовать возможности DSM-платформы. Апробация подхода проведена при проектировании системы автоматизации деятельности службы мониторинга нефинансовых предприятий, интегрирующей несколько унаследованных систем.

Разработка и трансформация моделей бизнес-процессов в интегрированной информационно-аналитической системе с помощью MetaLanguage

Основой для разработки предметно-ориентированного языка моделирования деятельности службы мониторинга является общецелевой (универсальный) язык, предназначенный для разработки моделей в исследуемой предметной области. Этот язык, с одной стороны, должен обладать достаточной выразительной мощностью для описания различных видов деятельности, автоматизируемых унаследованными системами, а также для моделирования комплексной (интегрированной) системы. С другой стороны, он должен обеспечить возможность создания на его основе предметно-ориентированных языков для моделирования различных подсистем, учитывающих их специфику и потребности пользователей.

При реализации проекта решаются следующие задачи:

- на основе данных о деятельности предприятий создаётся метамодель предметной области (общецелевой язык системы);
- на основе метамодели общецелевого языка строятся языки моделирования работы предприятий различных отраслей;
- на разработанном языке описываются модели, представляющие конкретные бизнес-процессы, протекающие в исследуемой предметной области.

На рис. 1 показана метамодель общецелевого языка, созданная с помощью языкового

инструментария MetaLanguage. Созданный общецелевой DSL обладает следующими преимуществами: достаточная выразительная мощь, обеспечивающая адекватное представление бизнес-процессов в исследуемой области; простота метамодели для понимания, высокая степень её прозрачности; ориентация на решаемые задачи в области создания комплексной системы мониторинга, что даёт возможность пользователям легко освоить язык и работать с моделями с использованием привычных профессиональных терминов.

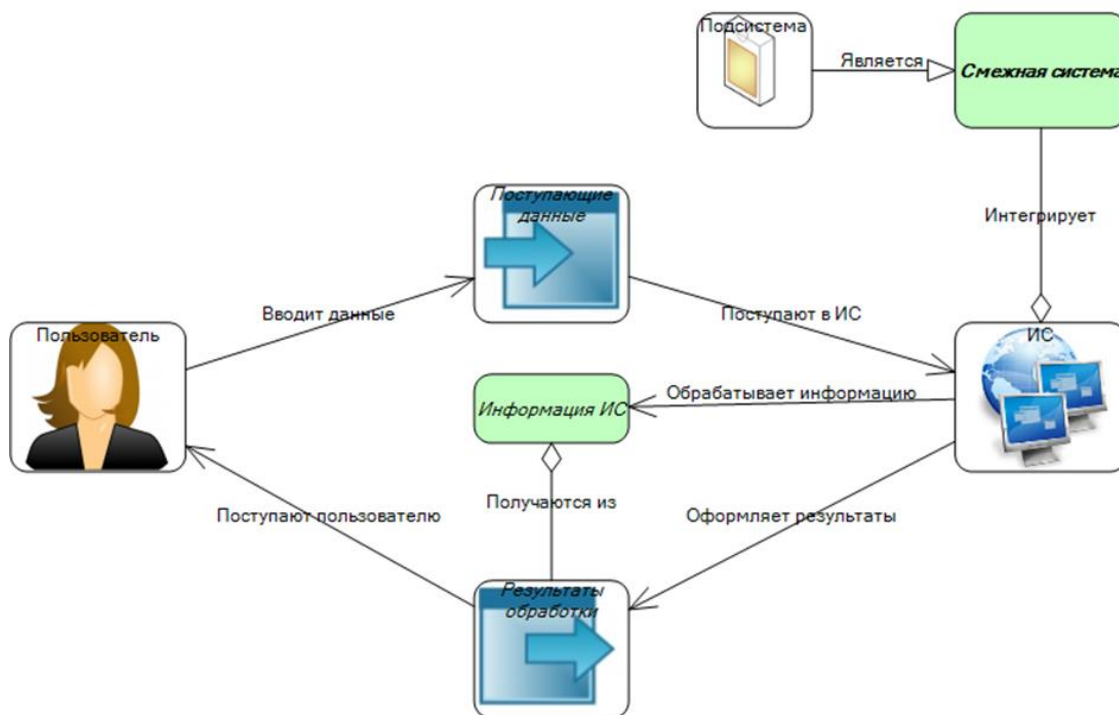


Рис. 1. Метамодель общецелевого языка моделирования, построенная с использованием системы MetaLanguage

При реализации метамодели языка были созданы следующие сущности (рис. 1):

- «Подсистема» представляет подсистемы, входящие в состав комплексной ИС;
- «Смежная система» – любая подсистема комплексной системы;
- «ИС» – комплексная (интегрирующая) система;
- «Пользователь» представляет все роли пользователей, имеющих доступ для работы с комплексной системой;
- «Поступающие данные» – входные данные комплексной системы;
- «Информация ИС» – промежуточные данные на различных этапах их обработки;
- «Результаты обработки» – выходные данных комплексной системы.

Сущности «Подсистема» и «Смежная система» имеют связь типа наследование «Является». Сущности «Смежная система» и «ИС» имеют связь типа агрегация «Интегрирует». Оставшиеся связи сущностей имеют тип ассоциация: между «ИС» и «Пользователь» – связь «Передаёт результаты», между «Пользователь» и «Поступающие данные» – «Вводит данные», между «Поступающие данные» и «ИС» – «Поступают в ИС», между «ИС» и «Информация ИС» – «Обработывает информацию», между «ИС» и «Результаты обработки» – «Оформляет результаты обработки», между «Пользователь» и «Результаты обработки» – «Получает результаты».

На основе общецелевого языка моделирования построена модель обработки заявки сотрудниками службы мониторинга. На рис. 2 показан её упрощённый вариант. Модель, построенная на основе разработанной метамодели общецелевого языка, не может детально представить все операции обработки заявок системы мониторинга. Кроме того, построенная модель не отражает разделение комплексной системы на конкретные подсистемы;

функциональные возможности каждой подсистемы и степень их использования в процессе обработки невозможно выделить. Таким образом, построенная модель имеет ряд недостатков: недостаточно наглядна, не отражает функциональные возможности конкретных подсистем, не отражает специфику работы конечных пользователей.

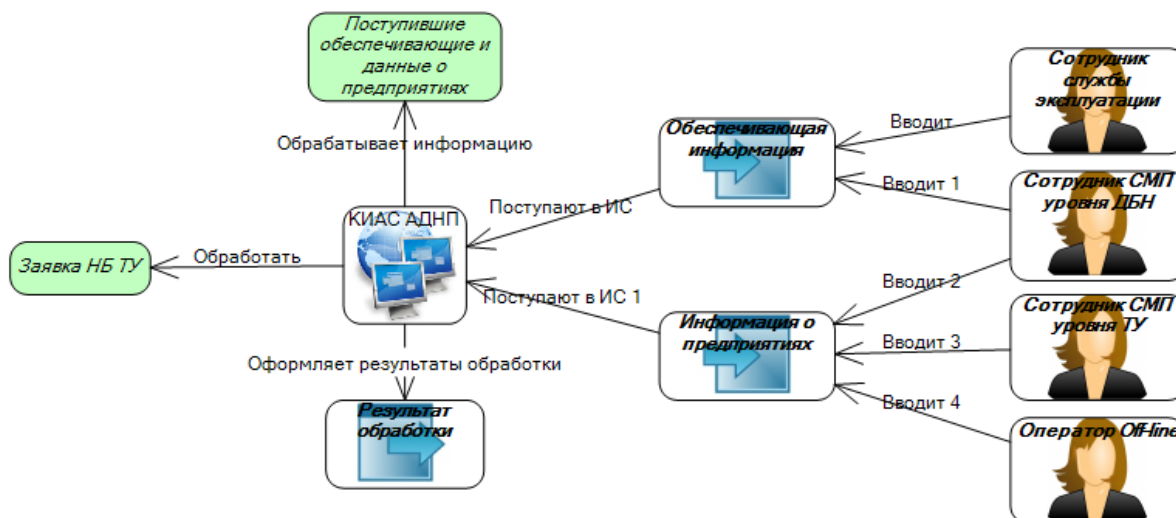


Рис. 2. Модель обработки заявки, построенная на основе общецелевой метамодели

Для устранения этих недостатков следует создать специфические языки для решения конкретных задач различных подсистем для конкретных категорий пользователей.

Разработанный общецелевой язык используется в качестве метаязыка при создании языков второго уровня, позволяющих более наглядно описать бизнес-процессы системы, автоматизирующей решение задач мониторинга. Новый предметно-ориентированный язык создаётся в нотации конкретного синтаксиса общецелевого языка моделирования (рис. 3).

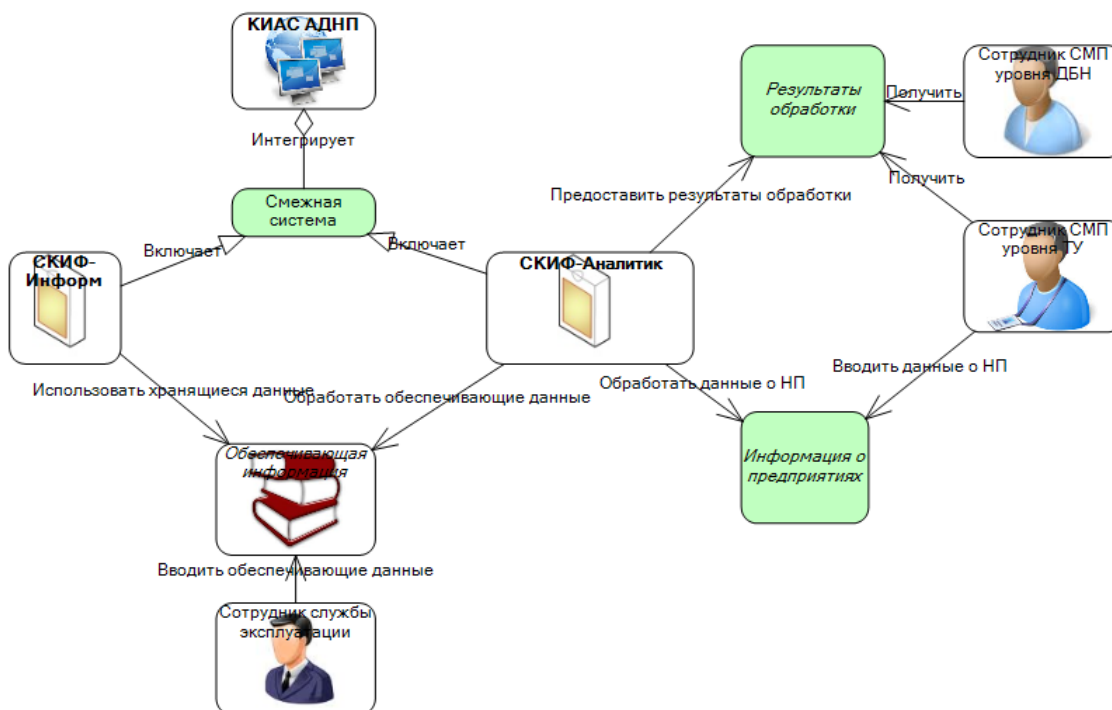


Рис. 3. Упрощённая метамодель языка второго уровня, построенная в системе MetaLanguage

На основе построенной предметно-ориентированной метамодели построена более детальная модель обработки заявки сотрудниками службы мониторинга (рис. 4).

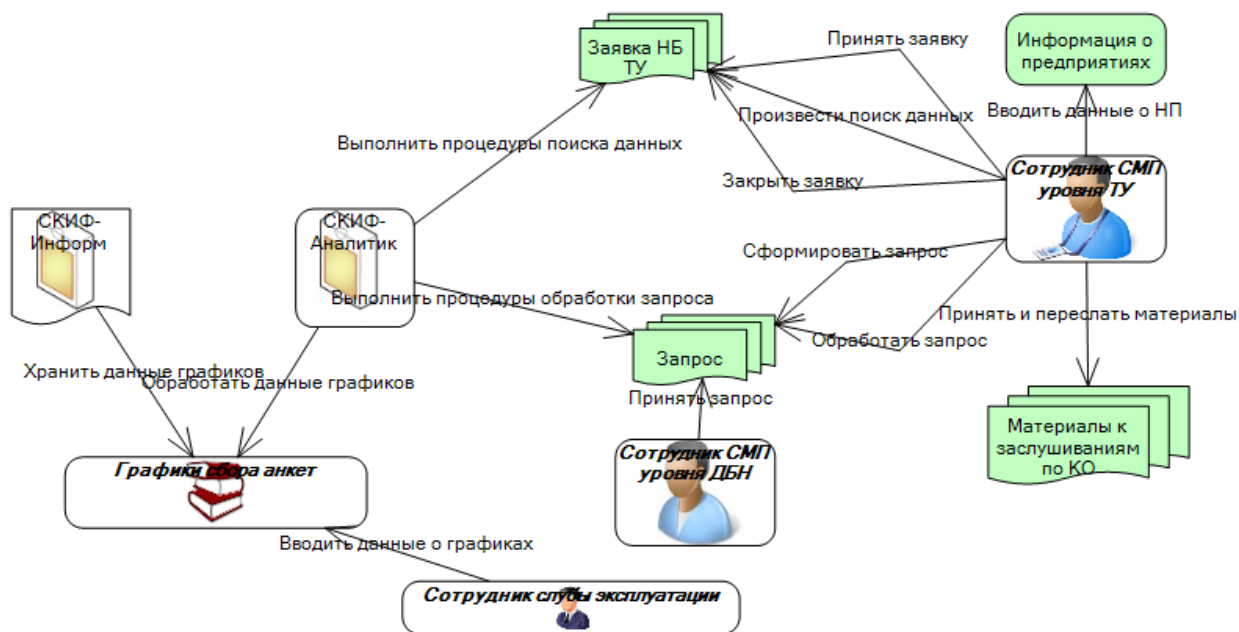


Рис. 4. Упрощённая модель обработки заявки, построенная на основе DSL второго уровня

По сравнению с ранее построенной моделью, построенной на основе общецелевой метамодели, данная модель, разработанная с использованием предметно-ориентированного языка второго уровня, более наглядна, отражает специфику работы конечных пользователей и функциональные возможности одной из подсистем, позволяет сотрудникам службы мониторинга работать в привычных терминах. Языки моделирования для описания других подсистем также могут быть созданы на основе общецелевого языка моделирования.

Приведённые примеры показывают возможности подхода, обеспечивающие создание моделей, настраиваемых на нужный уровень детализации и специфику работы подсистем и пользователей. При реализации исследовательского проекта были созданы модели процессов для двух пилотных подсистем комплексной информационно-аналитической системы.

Однако для решения задач проекта возникла необходимость *перевода построенных моделей на другие языки* для передачи результатов моделирования разработчикам ИС, в частности, потребовалось построить ER-диаграммы, диаграммы прецедентов и пр. Автоматизировать перевод построенных моделей на нужный язык можно с помощью средств трансформации моделей системы MetaLanguage.

Трансформация моделей включает два шага:

- *разработка правил трансформации* с использованием метамodelей (языков) исходной и целевой моделей;
- собственно *трансформация моделей* (перевод моделей с исходного языка моделирования на целевой язык в соответствии с описанными правилами).

Если правила трансформации заданы, то второй шаг выполняется автоматически в соответствии с этими правилами. Более того, описанные правила можно применять и для трансформации моделей, которые будут описаны с использованием тех же языков позднее, что существенно снижает трудоёмкость перевода. Набор правил может расширяться. В описания правил, как и в описания языков, можно вносить изменения).

Для перевода моделей, описанных на созданном DSL, в нотации UML (диаграммы деятельности) была использована метамодель, представленная на рис. 5.

Правила трансформации, разработанные в системе MetaLanguage для перевода построенных моделей на язык UML (для построения диаграммы деятельности), приведены в табл. 1.

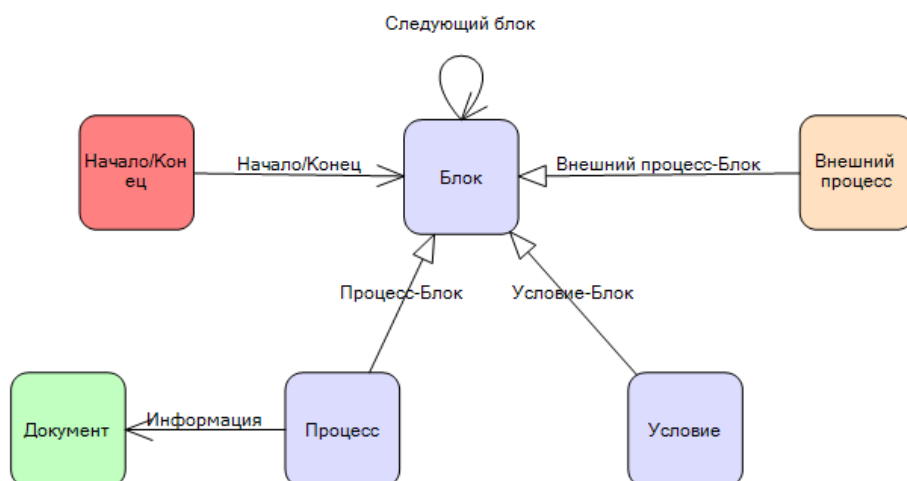
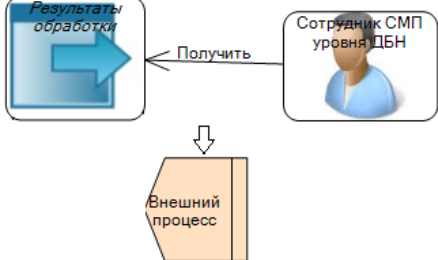



Рис. 5. Мета модель диаграммы деятельности в системе MetaLanguage

Таблица 1. Правила трансформации модели «Обработка заявки» в нотации UML (диаграмма деятельности)

Правило трансформации (левая и правая части)	Графическое отображение
Информация о предприятиях – Документ	
Обеспечивающая информация – Документ	
Результат обработки – Документ	
СКИФ-Информ – Блок	
СКИФ-Аналитик – Блок	
Сотрудник службы эксплуатации – Блок	
Сотрудник СМП уровня ТУ – Блок	
Сотрудник СМП уровня ДБН – Блок	

Правило трансформации (левая и правая части)	Графическое отображение
Обработать обеспечивающие данные – Процесс	
Обработать данные о НП – Процесс	
Предоставить результаты обработки – Процесс	
Использовать хранящиеся данные – Процесс	
Вводить обеспечивающие данные – Внешний процесс	
Вводить данные о НП – Внешний процесс	

Правило трансформации (левая и правая части)	Графическое отображение
Получить – Внешний процесс	
Получить 1 – Внешний процесс	

На рис. 6 представлен результат выполнения трансформации типа «модель-модель» для упрощённой модели «Обработка заявки».

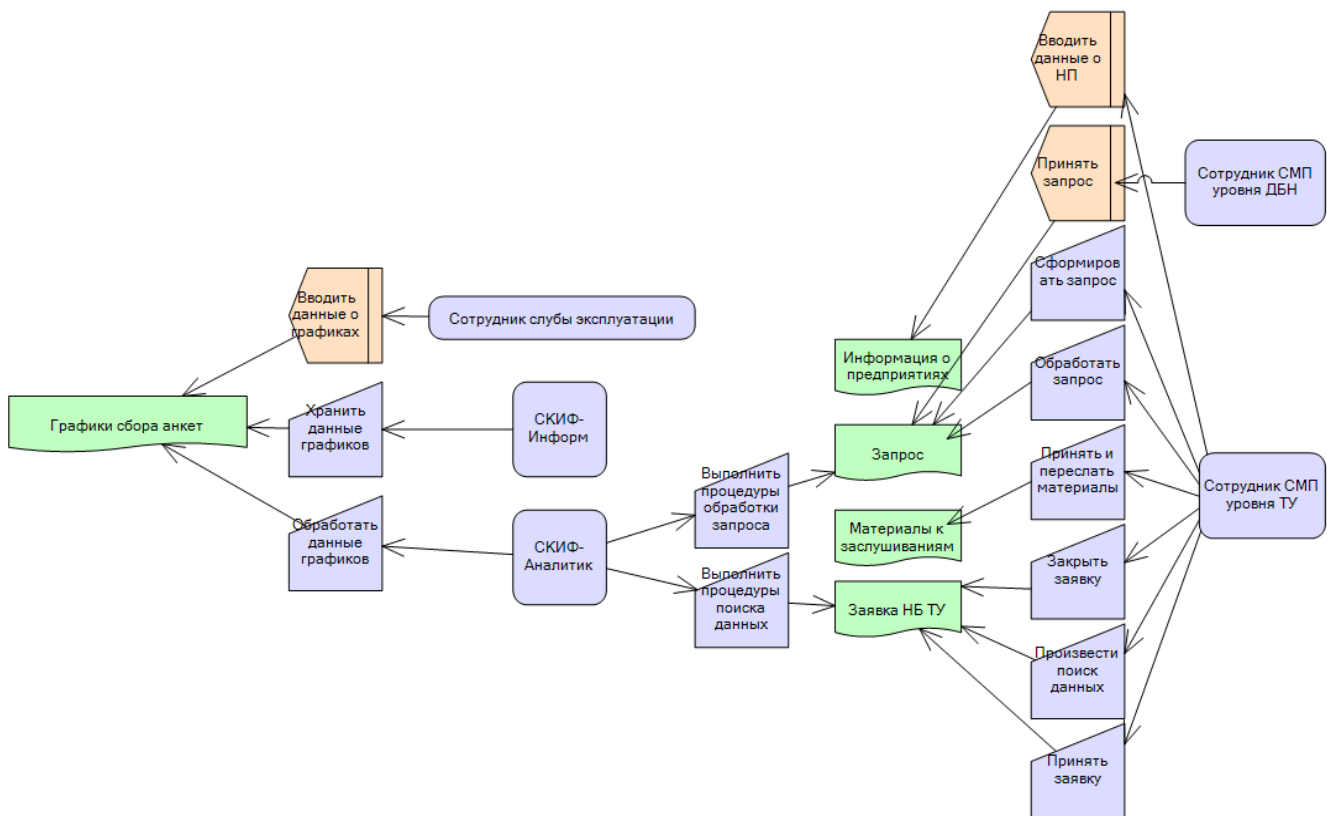



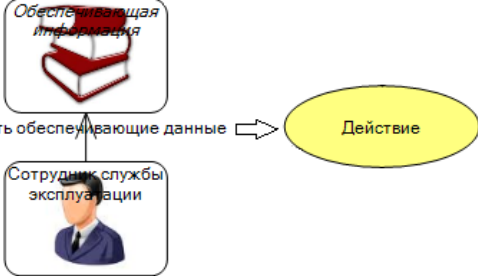
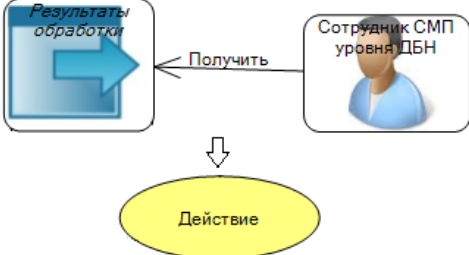
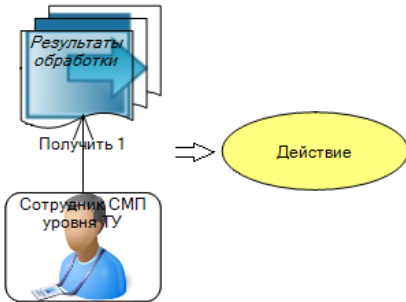
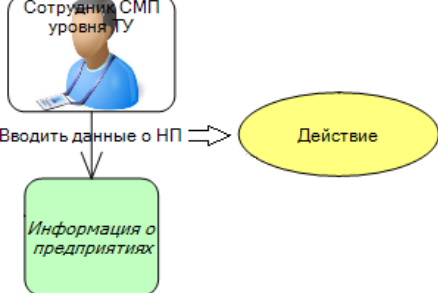


Рис. 6. Результат трансформации модели «Обработка заявки» – диаграмма деятельности

Помимо этого, модель обработки заявки была трансформирована в нотацию диаграммы вариантов. На рис. 7 представлена метамодель диаграммы Use-Case, а на рис. 8 – результат трансформации модели, выполненной в соответствии с правилами, показанными в табл. 2.

**Таблица 2. Правила трансформации модели «Обработка заявки»
в нотации диаграмм вариантов использования**

Правило трансформации (левая и правая части)	Графическое отображение
Сотрудник службы эксплуатации – Актор	
Сотрудник СМП уровня ДБН – Актор	
Сотрудник СМП уровня ТУ – Актор	
Вводить обеспечивающие данные – Действие	
Получить результаты обработки – Действие	
Получить результаты обработки – Действие	
Вводить данные о НП – Действие	

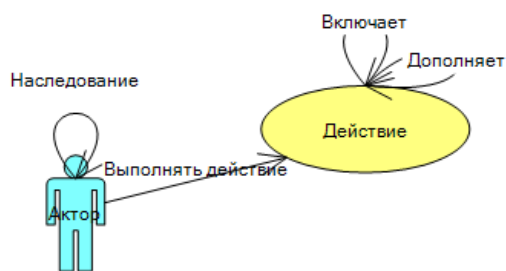


Рис. 7. Мета модель диаграммы вариантов использования

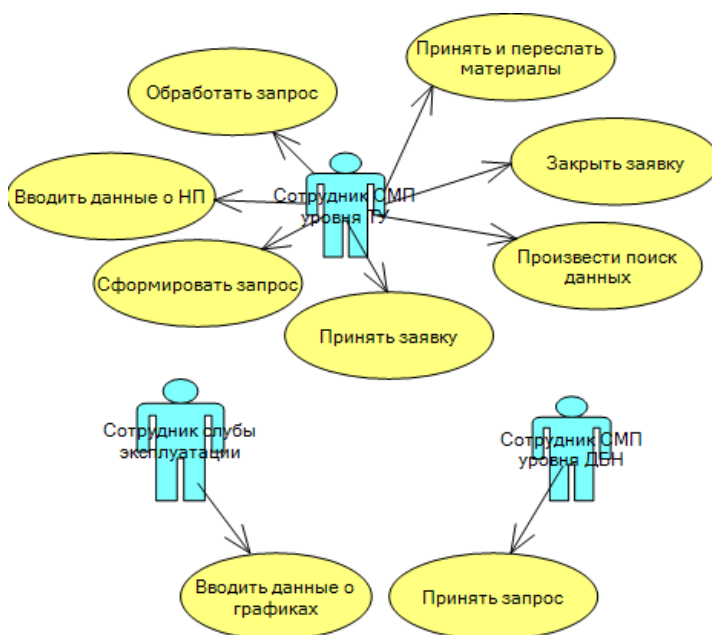


Рис. 8. Результат трансформации модели «Обработка заявки» – диаграмма Use-Case

Описание трансформаций «модель-модель» не требует от пользователя знания какого-либо специального языка программирования. Все операции реализуются в той же среде и на том же уровне требований к квалификации пользователей, что и разработка языков и моделей.

В ходе выполнения исследовательского проекта были также описаны правила и выполнены трансформации типа «модель-текст» для генерации кода на языке SQL. Полученный код может быть использован для создания базы данных интегрированной системы.

Заключение

Проведённые исследования показали, что современные DSM-платформы могут использоваться не только для разработки предметно-ориентированных языков и моделей, но и как средства интеграции различных информационных и аналитических систем. Возможности динамического изменения языков и моделей, их настройки в процессе разработки информационных и аналитических систем позволяют значительно снизить трудоёмкость решения задач моделирования за счёт переиспользования ранее созданных моделей, автоматизации их преобразований. Кроме того, средства моделирования при использовании DSM-платформ, предоставляющих развитые инструментари для разработки языков и моделей, не предъявляющие высоких требований к квалификации пользователей, становятся доступными для различных категорий пользователей.

Библиографический список

1. *Лядова Л.Н.* Многоуровневые модели и языки DSL как основа создания интеллектуальных CASE-систем / *Л.Н. Лядова* // Труды международных научно-технических конференций «Интеллектуальные системы» (AIS'08) и «Интеллектуальные САПР» (CAD-2008). Научное издание в 4-х томах. Т. 2. – М.: Физматлит, 2008. С. 37-41.
2. *Kelly S., Lyytinen K., Rossi M., Tolvanen J.P.* MetaEdit+ at the Age of 20 // In *Bubenko J., Krogstie J., Pastor O., Pernici B., Rolland C., Slyberg A.* (eds.) *Seminal Contributions to Information Systems Engineering*. Springer, 2013. P. 131-137.
3. MetaCase. MetaEdit+ Version 5.0 Workbench User's Guide. [<http://metacase.com/support/50/manuals/mwb/Mw.html>] (Проверено: 18.04.2014).
4. *Cook S., Jones G., Kent S., Wills A.C.* *Domain-Specific Development with Visual Studio DSL Tools*. Reading : Addison-Wesley, 2007. – 560 p.
5. Overview of Domain-Specific Language Tools [<http://msdn.microsoft.com/en-us/library/bb126327.aspx>] (Проверено: 18.04.2014).
6. *Сорокин А.В., Кознов Д.В.* Обзор Eclipse Modeling Project // Системное программирование. Вып. 5. 2010. С. 6-31.
7. *Gronback R.C.* *Eclipse Modeling Project: A Domain-Specific Language Toolkit*. Reading: Addison-Wesley, 2009. – 706 p.
8. *Терехов А.Н., Брыксин Т.А., Литвинов Ю.В.* QReal: платформа визуального предметно-ориентированного моделирования. // Программная инженерия, 2013, № 6, С. 11-19.
9. QReal:BP [http://qreal.ru/static.php?link=QRealBP_usage] (Проверено: 18.04.2014).
10. *Сухов А.О.* Методы трансформации визуальных моделей / Материалы III международной научно-технической конференции «Технологии разработки информационных систем ТРИС–2012». Таганрог: Изд-во Технол. инст. ЮФУ, 2012. – Т. 1. – С. 120-124.
11. *Замятина Е.Б., Лядова Л.Н., Сухов А.О.* О подходе к интеграции систем моделирования и информационных систем на основе DSM-платформы MetaLanguage / Материалы IV межд. научно-технической конференции «Технологии разработки информационных систем ТРИС-2013». Таганрог: Изд-во Технол. инст. ЮФУ, 2013. – Т. 1. – С. 61-70.
12. *Сухов А.О.* Инструментальные средства создания визуальных предметно-ориентированных языков моделирования / *А.О. Сухов* // Фундаментальные исследования. – 2013. – № 4 (ч. 4). – С. 848-852.
13. *Лядова Л.Н.* Подходы к описанию вертикальных и горизонтальных трансформаций метамodelей / *Л.Н. Лядова, А.П. Серый, А.О. Сухов* // Математика программных систем: межвуз. сб. науч. ст. – Пермь: Изд-во Перм. гос. нац. исслед. ун-та, 2012. – Вып. 9. – С. 33-49.
14. *Сухов А.О., Серый А.П.* Использование графовых грамматик для трансформации моделей / Материалы конференции «CSEDays 2012». Екатеринбург: Изд-во Урал. ун-та, 2012. – С. 48-55.
15. *Лубягина А.О.* О подходе к функциональной интеграции систем на основе технологии MDE / *А.О. Лубягина* // Известия ЮФУ. Технические науки. – 2014. – № 6(155). С. 159-163.