Лядова Людмила Николаевна
доцент, кандидат физико-математических наук,
доцент кафедры информационных технологий в бизнесе,
Национальный исследовательский университет «Высшая школа экономики»,
г. Пермь
E-mail: LLyadova@hse.ru

Суворов Николай Михайлович
студент бакалавриата, образовательная программа «Программная инженерия»,
Национальный исследовательский университет «Высшая школа экономики»,
г. Пермь
E-mail: SuvorovNM@gmail.com

Василюк Василий Алексеевич
студент бакалавриата, образовательная программа «Программная инженерия»,
Национальный исследовательский университет «Высшая школа экономики»,
г. Пермь
E-mail: vasvasal@gmail.com

АРХИТЕКТУРА DSM-ПЛАТФОРМЫ, ОСНОВАННОЙ НА ЗНАНИЯХ

Lyadova Lyudmila Nikolaevna
Docent, PhD in Computer Science,
Associate Professor at Information Technology in Business Department
National Research University Higher School of Economics, Perm City
E-mail: LLyadova@hse.ru

Suvorov Nikolai Mikhailovich
Student of the bachelor educational program "Software Engineering"
National Research University Higher School of Economics, Perm City
E-mail: SuvorovNM@gmail.com

Vasilyuk Vasiliy Alekseyevich
Student of the bachelor educational program "Software Engineering"
National Research University Higher School of Economics, Perm City
E-mail: vasvasal@gmail.com

THE ARCHITECTURE OF THE KNOWLEDGE-BASED DSM-PLATFORM

Аннотация: Цель – разработка архитектуры DSM-платформы, основные функции которой базируются на использовании многоаспектной онтологии. Языковые инструментарии, автоматизирующие разработку редакторов моделей, генерацию кода и документации, снижают трудоёмкость создания и использования новых предметно-ориентированных языков (DSL), но разработка DSL требует не только

глубоких знаний в области моделирования, но и понимания особенностей предметных областей. В архитектуру включаются средства, автоматизирующие построение метамоделей DSL на основе существующих языков и онтологий предметных областей, для моделирования которых создаются языки. Разработка DSL основывается на выполнении проекции онтологии предметной области на онтологию языков моделирования, результатом которой является метамодель нового языка. Онтология является также основой реализации семантических трансформаций моделей.

Abstract: The goal is developing the architecture of the DSM platform, the main functions of which are based on the use of multifaceted ontology. Language toolkits automating the model editors development, code generation and documentation reduce the difficulty of creating and using new domain specific languages (DSL), but DSL development requires not only deep knowledge in the field of modeling, but also an understanding of the domain areas features. The architecture includes tools that automate the designing DSL metamodels on the basis of existing languages and domain ontologies intended for the modeling of areas languages are created for. DSL development is based on performing a mapping of the domain ontology into the ontology of modeling languages, the result of which is a metamodel of the new language. Ontology is also the basis for the implementation of semantic model transformations.

Ключевые слова: предметно-ориентированное моделирование, многоуровневое моделирование, языки моделирования, многоаспектная онтология, языковой инструментарий, автоматизация разработки DSL.

Keywords: domain specific modeling, multi-level modeling, modeling languages, multi-faceted ontology, language toolkits, DSL development automation.

Введение

В современном динамично меняющемся мире остро встаёт проблема оперативной разработки новых информационных систем (ИС), соответствующих потребностям пользователей, отражающих всю специфику их деятельности, а также актуализации существующих систем, их настройки на меняющиеся условия.

Одним из подходов к решению этой задачи является создание систем, работающих в режиме интерпретации моделей предметных областей, описывающих все аспекты автоматизируемой деятельности. В рамках этого подхода реализованы различные технологии, которые были использованы при создании ИС различного назначения [5, 6]. Однако «слабым звеном» в таких системах остаётся реализация сложной бизнес-логики, автоматизации не отдельных операций, а бизнес-процессов или технологических, производственных процессов в целом – требуется реализация доступных пользователям средств их формального описания.

Наиболее перспективным подходом к решению этой задачи является применение *специализированных языков* (текстовых или визуальных), которые были бы доступными для использования не профессиональными разработчиками, а пользователями, являющимися специалистами в конкретных предметных областях. Универсальные языки программирования и даже диаграммные языки не удовлетворяют этим требованиям.

Языково-ориентированный подход предложен в 90-е годы как новая парадигма программирования [15]. Специализированные языки, системы программирования, предназначенные для решения определённых задач в определённой предметной области, разрабатывали и ранее. Однако создание таких языков связано с необходимостью разработки не только языка, но и соответствующих средств разработки: редактора, компилятора или интерпретатора. Парадигма языково-ориентированной разработки предназначена для решения этой проблемы.

Развитие средств визуального моделирования, повсеместное применение диаграммных языков моделирования (диаграмм UML и пр.) при создании ИС с помощью CASE-инструментариев значительно упростило процесс разработки. *Предметно-ориентированное моделирование* (*Domain Specific Modelling*, DSM) становится шагом в направлении привлечения пользователей к процессу разработки, делая для них доступными средства моделирования, позволяющие строить модели на языках, отражающих особенности конкретной предметной области и задач пользователей.

Задачи создания визуальных языков, настроенных на применение в определённой области, решаются с помощью *языковых инструментариев*, DSM-платформ. Они включают либо средства генерации программного обеспечения в соответствии с построенными моделями, либо средства их интерпретации. В настоящее время существует множество *DSM-платформ*, которые часто выступают в качестве «meta-CASE-инструментариев», позволяющих создавать как языки моделирования для определённых предметных областей – предметно-ориентированные языки (Domain Specific Language, DSL), так и редакторы для этих языков, генераторы кода и документации (MetaEdit+ и пр.).

Языковые инструментарии, снижают трудоёмкость создания и использования новых предметно-ориентированных языков, но разработка DSL требует не только глубоких знаний в области моделирования, но и понимания особенностей предметных областей. Таким образом, при создании DSL должны взаимодействовать два специалиста: эксперт в предметной области и ИТ-специалист, владеющий знаниями в области формальных языков и грамматик, навыками разработки метамоделей новых языков.

Задача разработки языков усложняется при создании ИС для сложных, масштабных бизнес-систем, крупных предприятий, где необходимо разработать не одну модель, а множество моделей, отражающих различные аспекты моделируемой системы, и, следовательно, создать несколько языков моделирования, с помощью которых должны быть созданы эти модели [8, 9]. Кроме того, при переходе от решения одной задачи к другой необходимо согласовывать соответствующие модели, выполнять их трансформации.

Решение проблемы – в создании средств, которые позволили бы максимально упростить процесс создания DSL, а также определения правил трансформации моделей при переходе от решения одной задачи, для которой уже разработана модель, к другой, которые бы обеспечили преемственность в решении этих задач.

Перспективным подходом представляется решение на основе применения *онтологий*, которые, с одной стороны, можно применять для анализа документов, нормативно-справочной информации для получения знаний о предметной области, а с другой – для автоматизации разработки языков и трансформаций.

Цель данного проекта – разработка архитектуры DSM-платформы, основные функции которой базируются на использовании многоаспектной онтологии. В архитектуру DSM-платформы включаются средства, автоматизирующие построение метамоделей DSL на основе существующих языков и онтологий предметных областей, для моделирования которых создаются языки. Разработка DSL основывается на выполнении проекции онтологии предметной области на онтологию языков моделирования, результатом которой является метамодель нового языка. Онтология является также основой реализации семантических трансформаций моделей.

Языково-ориентированный подход

В рамках традиционного подхода к разработке программ выполняется три этапа решения задачи с помощью компьютера: анализ задачи и формирование «концептуальной модели» решения (алгоритма); выбор языка программирования общего назначения, на котором будет реализован алгоритм решения задачи; собственно программирование (кодирование), в ходе которого осуществляется отображение «концептуальной модели» на выбранный язык. Этот подход применим при решении относительно простых задач, при реализации «численных» алгоритмов, но он не подходит при решении сложных задач автоматизации деятельности крупных бизнес-систем, где осуществляются различные виды деятельности, параллельно развиваются технологические, производственные процессы и процессы управления.

Решение каждой задачи автоматизации в сложной предметной области требует реализации собственных подходов, своих языковых средств, обладающих достаточной выразительностью для отражения специфики условий решения.

Языково-ориентированный подход (ЯОП) предполагает после выполнения этапа анализа выбор специального DSL, подходящего для решения этой предметной области и задачи, а при его отсутствии – создание такого DSL. При этом третий этап – программирования – предполагает, что решение является относительно прямым отображением концептуальной модели на DSL.

Суть ЯОП [15] в том, что разработчик получает возможность работать в терминах концепций и понятий проблемы, задачи, которую он решает, вместо того чтобы переводить разработанную концепцию в нотации языков программирования общего назначения (классы, методы, циклы и т.п.). При определённых условиях и подготовке потенциальный пользователь становится разработчиком.

Сложность решения задачи «сдвинута» на шаг создания DSL, но применение подходящих инструментов – языковых инструментариев – существенно облегчает этот шаг. В настоящее время существует множество универсальных платформ (DSM-платформ), предназначенных для создания языков (Meta Programming System, MetaEdit+, QReal и пр.).

Многомодельный подход и предметно-ориентированное моделирование с использованием онтологического подхода

Было показано, что, с одной стороны, использование DSL позволяет разработчикам абстрагироваться от особенностей универсальных языков (нотаций), сосредоточиться на решаемых задачах, разработке концептуальных моделей, а с другой – абстрактные конструкции универсальных языков заменяет специальными средствами DSL, заточенными на решение задачи именно в этой области. Это также упрощает и сопровождение программного обеспечения.

Методы и средства разработки отдельных DSL хорошо изучены, но сложности возникают при создании реальных корпоративных приложений, где требуется *несколько DSL*, предназначенных для решения различных задач, что требует особой поддержки. В [9] предлагается подход к разработке ИС с использованием DSL, где предлагается решение «перекрытия» задач, решаемых с помощью нескольких DSL. Этот метод основан на установлении соответствия между моделями, созданными на разных языках, что даёт возможность *трансформаций* построенных моделей.

Рассматривается три типа связей. Простейшим способом подключения различных DSL является использование *явно типизированных ссылок*, «поставляемых» с базовой технологией моделирования, *связывающих метамодели языков*. Однако использование типизированных привязок приводит к появлению «монолитных» моделей, что затрудняет редактирование, приводит к сильным зависимостям между языками, ограничивает возможность повторного использования созданных DSL. Применение нескольких *небольших*, *связанных моделей* («*мягких*» *программных ссылок*) может помочь повысить производительность и оставить языки независимыми. Использование программных ссылок требует специальной платформы для выполнения, которая знает используемые языки и может соответствующим образом обрабатывать неявные модельные соединения в ходе эксплуатации. В генеративных сценариях, где DSL предназначены для многократного использования и компоновки в различных контекстах, необходимы расширенные ссылки, такие как *семантические соединения* [8]. Для их реализации предлагается использовать *онтологический подход*: метамодель DSL связывается с соответствующими понятиями из онтологии языка, созданной для DSL. Понятия онтологии языка наследуются от концептов онтологии обрамления (единой онтологии моделирования программного обеспечения, USMO). Связи «*maps to*» указывают, что на уровне метамодели определены соответствующие правила отображения, которые запускают создание связей между моделями, созданными с помощью DSL, описанных в онтологии. Таким образом, DSM-платформы с репозиториями

моделей расширяются базами знаний, которые «кодирует» различные соединения моделей на уровне языка моделирования.

Таким образом, применение онтологий позволяет автоматизировать решение сложных задач при разработке ИС. Но сложность «смещается» на выполнение другой задачи – задачи разработки онтологии, создания базы знаний.

Управление изменением метамоделей в MDE

Разработка на основе моделей (Model-driven development, MDD) поднимает значение моделей до основных артефактов разработки. Этот подход успешно применяется в различных сложных областях. При этом используются как универсальные языки моделирования, нотации (UML и т.п.), так и DSL. Традиционным при разработке новых языков является подход, основанный на грамматиках. Однако реализуется и другой подход, в основе которого лежит понятие метамодели.

Между языками, основанными на *метамоделях*, и языками, основанными на *грамматиках*, существует различие: грамматики позволяют задать интегрированное определение конкретного и абстрактного синтаксиса, что облегчает обработку моделей, но затрудняет повторное использование созданных на их основе языковых инструментариев (таких как редакторы, интерпретаторы, отладчики и т.д.). Метамодели описывают только абстрактный синтаксис (структуру) языка и их конкретный синтаксис обычно определяется через разработанные для них редакторы [7].

Разработка программного обеспечения на основе моделей (Model-Driven Engineering, MDE) предполагает систематическое использование моделей при создании программного обеспечения. Модель должна быть задана с помощью четко определенного языка моделирования с точным синтаксисом и семантикой. Синтаксис языка определяется *метамоделью*. Как правило, метамодели являются фиксированными, но существует несколько сценариев использования, которые требуют *настройки* существующих метамоделей. Базовые языки, предназначенные для решения определённых задач, могут быть изменены для настройки на специфику конкретной области, в которой эти задачи решаются. Метамодели языков должны быть преобразованы согласно определенным правилам, которые задаются неформально, что затрудняет автоматизацию преобразований.

В работе [10] для решения этой проблемы предлагается механизм, позволяющий задавать правила настройки, расширения для метамоделей, а также инструмент, позволяющий изменять метамодели согласно этим правилам. Инструмент основан на структуре моделирования Eclipse. Он был реализован в виде плагина Eclipse. Рассматриваются различные типы настройки метамодели: детализация; расширение (добавление новых элементов); упрощение (устранение деталей); сокращение (удаление отдельных элементов) и пр. Эти правила описываются с помощью языка, метамодель которого (метамодель кастомизации) представлена в работе.

Эти правила могут быть описаны в онтологии (базе знаний) DSM-платформы, что позволит не только задавать правила семантических

трансформаций моделей, но и правила создания новых языков на основе метамоделей существующих (базовых) языков, выступающих в роли «шаблонов» («паттернов»).

Онтологический подход к разработке языков моделирования

Авторами [2] предлагается всесторонний анализ и систематизация визуальных языков моделирования. Отмечается, что визуальные языки хорошо дополняют технологии онтологического моделирования, делая содержимое онтологий наглядным для человека. В статье предлагается подход для выбора визуального языка в зависимости от типа представляемых знаний (аспектов предметной области), для описания которых он должен использоваться. В основу предлагаемого подхода положена *семантическая классификация визуальных языков и типов знаний*. Одним из наиболее значимых результатов применения предложенного подхода являются практические рекомендации по выбору подходящей визуальной нотации при моделировании бизнес-приложений на примере решения задач автоматизации органов государственного управления.

В работе определяется взаимосвязь между типом знаний, которые нужно представить, и визуальными языками моделирования: требуется взгляд с разных точек зрения: необходимо представить: ЧТО-, КАК-, ЗАЧЕМ-, ПОЧЕМУ-, КТО-, ГДЕ- и КОГДА-знания. Чтобы дать формализованное описание ответа на каждый вопрос, применяются соответствующие языки (нотации). Таким образом можно установить и описать в онтологии как задачи, так и языки, применяемые для решения этих задач.

Архитектура DSM-платформы, основанной на многоаспектной онтологии

При реализации языкового инструментария MetaLanguage [12, 13] были разработаны средства создания иерархий моделей, а также алгоритмы горизонтальной и вертикальной трансформации моделей, созданных и использованием DSL, снижающие трудоёмкость их выполнения.

В [14] предлагается использовать новую математическую модель, которая должна снизить сложность реализации алгоритмов, упростить процесс разработки редакторов моделей.

Однако разработанные визуальные средства и алгоритмы трансформаций не обладают достаточной степенью гибкости, выразительности. Кроме того, процесс разработки самого DSL, его метамодели требует подготовки.

Решить эти проблемы предлагается за счёт включения в состав DSM-платформы базы знаний – много аспектной онтологии, которая должна обеспечить как автоматизацию построения DSL, так и выполнения семантических трансформаций моделей.

Упрощённая структура DSM-платформы показана на рис. 1.

Рис. 1. Структура DSM-платформы

Ядро инструментария – *многоаспектная онтология*, где *онтология предметной* (*проблемной*) *области* содержит описания понятий предметной области и связей между ними; *онтология задач* – описание задач, для решения которых разрабатываются различные языки моделирования; *онтология языков моделирования* – это описание различных языков моделирования, связей между ними (на уровне метамоделей, описывающих синтаксис языков), которые могут быть использованы при выполнении трансформаций.

В онтологии языков моделирования содержатся как описания «базовых» языков, их метамоделей, так и всех языков, созданных на их основе в соответствии с заданными правилами, перечисленными выше [10]. Многоаспектная онтология включает также описание связей между этими онтологиями, правила их «соединения» при реализации конкретного проекта, разработке ИС. Эта онтология задаёт семантику трансформации моделей при переходе от решения одной задачи к другой с использованием различных языков моделирования. Кроме того, определяется классификация языков с точки зрения их назначения [2]. Средства генерации метамоделей новых DSL – это средства настройки на предметную область и решаемые задачи. Новые языки создаются с помощью операции отображения (проекции) понятий предметной области на элементы «базовых» языков, описанных в онтологии языков моделирования [8]. Правила трансформации, заданные для базовых языков, «работают» для всех DSL, созданных на их основе [3, 4]. Упрощённая схема связи фрагментов многоаспектной онтологии показана на рис. 2.

Кроме того, многоаспектная онтология – средство для семантического анализа текстов (программной документации, нормативно-справочной информации, справочников и классификаторов), которые могут использоваться как основа для построения онтологии предметной области, которая становится, таким образом, «саморасширяемой», а также база для генерации документов, автоматического документирования проектов [1, 11].

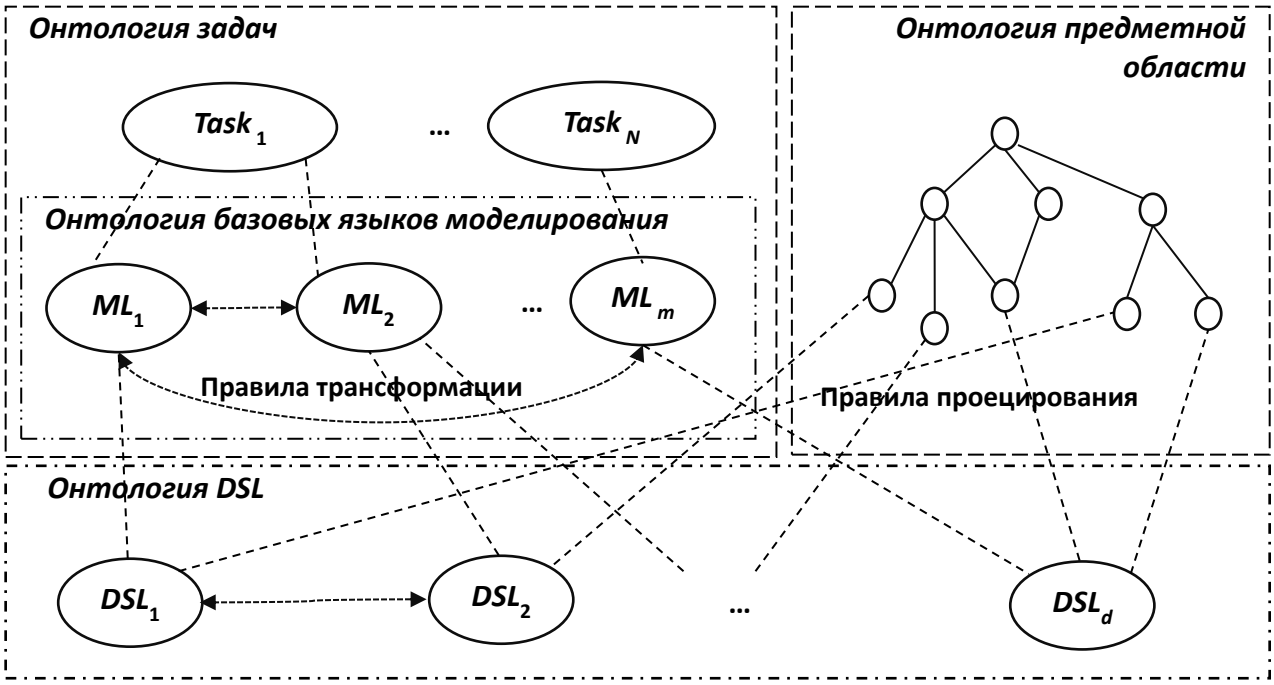

Рис. 2. Структура многоаспектной онтологии

Заключение

Предлагаемая архитектура DSM-платформы и разработанные для её реализации модели и алгоритмы дают возможность создания гибких средств разработки моделей при выполнении сложных проектов, требующих применения множества языков моделирования. На основе базовых языков моделирования, в качестве которых могут выступать диаграммы UML и др., можно создать DSL, предназначенные для решений задач конкретных проектов, задать связи между ними для выполнения семантических трансформаций при решении задач разных этапов. Созданные DSL, в свою очередь, могут служить основой для создания новых языков и т.д. Такая архитектура обеспечивает максимальную возможность переиспользования созданных (мета)моделей.

Сложность разработки онтологии компенсируется специальными средствами, основанными на интервьюировании экспертов, а также средствами автоматизации извлечения знаний из различных источников (документов, стандартов, нормативно-справочной информации, классификаторов и т.п.).

Список литературы

1. Брезгина Э.А., Ланин В.В. Автоматизированное формирование требований к информационным системам на основе анализа проектной документации // Технологии разработки информационных систем: сборник статей международной научно-практической конференции, ТРИС-2015. – Таганрог: Издательство ЮФУ, 2015. – С. 19-27.

2. Кудрявцев Д.В., Гаврилова Т.А. Систематизация визуальных языков моделирования // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012: труды конференции. Т. 2. – Белгород: Изд-во БГТУ, 2012. – С. 177-184.

3. Лядова Л.Н., Сахипова М.С., Сухов А.О. Семантическая трансформация моделей на основе онтологий // Информатизация и связь. – 2019. – № 5. С. 85-88.

4. Миков А.И., Лядова Л.Н. Методы создания DSL-инструментария и моделирования предметных областей на основе многоуровневых онтологий и графовых грамматик: отчет по проекту РФФИ № 10-01-00794, 2010.

5. Рогозов Ю.И., Свиридов А.С., Кучеров С.А. Структурно-независимые базы данных для разработки конфигурируемых пользователем информационных систем // Информатизация и связь. – 2013. – № 2. С. 20-22.

6. Рыжков С.А., Лядова Л.Н. CASE-технология METAS // Математика программных систем: межвузовский сб. науч. трудов / Перм. ун-т – Пермь, 2003. – С. 4-18.

7. Butting A., Jansen N., Rumpe B., Wortmann A. Translating Grammars to Accurate Metamodels // International Conference on Software Language Engineering (SLE'18). – ACM, 2018. – P. 174-186.

8. Campos E., Kulesza U., Freire M., Aranha E. A Generative Development Method with Multiple Domain-Specific Languages // Product-Focused Software Process Improvement. PROFES 2014. Lecture Notes in Computer Science. – 2014. – Vol 8892. Springer, Cham.

9. Hessellund A., Lochmann H. An Integrated View on Modeling with Multiple Domain-Specific Languages // IASTED on ICSE. – 2009. – P. 1-10.

10. Jácome S., De Lara J. Controlling Meta-Model Extensibility in Model-Driven Engineering // IEEE Access. – 2018. – Vol. 6. – P. 19923-19939.

11. Lanin V., Sokolov G. Using multidimensional ontology of electronic document for solving semantic indexing problem // Proceedings of the 8th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2014). – P. 166-169.

12. Sukhov A., Lyadova L. An Approach to Development of Visual Modeling Toolkits // Advances in Information Science and Applications. Proceedings of the 18th International Conference on Computers. Vol. 1-2. – Santorini Island: CSCC, 2014. – P. 61-66.

13. Sukhov A., Lyadova L. Visual Models Transformation in MetaLanguage System // Advances in Information Science and Applications: Proceedings of the 18th International Conference on Computers. Vol. 2. – Santorini Island: CSCC, 2014. – P. 460-467.

14. Suvorov N.M., Lyadova L.N. HP-Graph as a Basis of a DSM Platform Visual Model Editor // Proceedings of the Institute for System Programming of the RAS. – 2020. – Vol. 32. No. 2. – P. 149-160.

15. Ward M. Language Oriented Programming // Software – Concepts and Tools. – 1994. – № 15. – P. 147-161.