

О ПОДХОДЕ К РАЗРАБОТКЕ ПРОФЕССИОНАЛЬНО-ОРИЕНТИРОВАННЫХ СИСТЕМ НА ОСНОВЕ DSM-ПЛАТФОРМ

А.О. Сухов, Л.Н. Лядова

Введение

Развитие информационных технологий (ИТ), расширение их возможностей привело к тому, что средства ИТ находят применение в новых областях. Возросшие потребности в создании и внедрении новых информационных систем (ИС) для различных предметных областей, автоматизирующих всё новые виды деятельности, уже не удовлетворяются при создании ИС с использованием традиционных средств разработки. Ещё одна проблема – настройка существующих систем на меняющиеся условия эксплуатации, потребности пользователей. Адаптируемость становится одним из ключевых свойств ИС, гарантирующих эффективность вложений в создание и внедрение ИС, определяющих их живучесть. Существующие промышленные технологии создания ИС не позволяют оперативно решать эти задачи. Всё это обусловило необходимость создания новых технологий, позволяющих оперативно осуществлять адаптацию, конфигурирование ИС.

Применение формальных моделей на всех стадиях проекта позволяет снизить вероятность возникновения ошибок в реализации системы; 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.

Моделирование с использованием предметно-ориентированных языков

При реализации *модельно-ориентированного подхода* к разработке ИС (или *Model-Centric Software*

Development – MCSDD) модели становятся центральным звеном на всех этапах создания ИС. Максимальная гибкость системы достигается при использовании различных технологий, основанных на применении моделей, не только в процессе разработки ИС, но и в ходе эксплуатации: система функционирует в режиме интерпретации построенных моделей, определяющих её характеристики, управляющих её поведением; настройка ИС осуществляется через изменение моделей. Таким образом, оперативность создания и настройки ИС определяется возможностями средств создания и модификации моделей.

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 Language*) или *предметно-ориентированных языков моделирования* (DSML – *Domain Specific Modeling Language*), позволяющих разработчикам и пользователям работать с формальными моделями в привычных для них терминах. Созданный DSL играет роль *метамодели* в иерархии создаваемых моделей.

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 для решения различных задач. Здесь речь идёт уже об использовании *нескольких предметно-ориентированных*

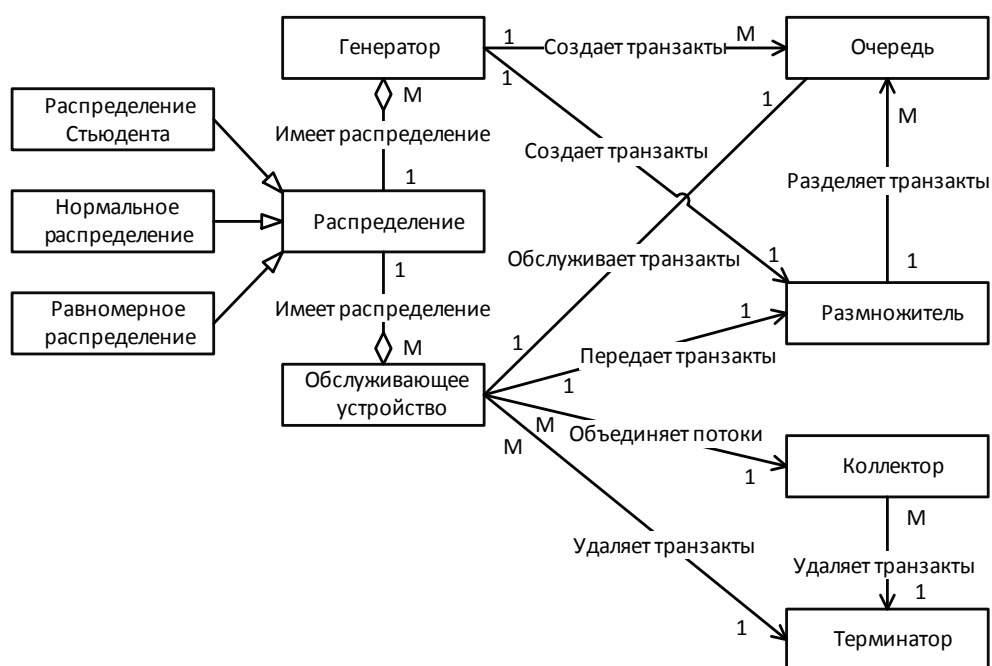


Рис. 1. Метамодел «общецелевого» языка описания моделирования СМО, выполненная средствами языка UML

языков, предназначенных для профессионально-ориентированной разработки и настройки системы.

Выбор «базового» языка для разработки DSL во многом определяет характеристики DSM-платформы.

Во многих случаях для построения моделей применяют язык UML, основанный на объектно-ориентированной парадигме, и подчёркивают значимость стандартизованного языка моделирования в качестве центрального звена процесса разработки систем (пример – MDA [4]).

Построение концептуальной модели предметной области с использованием UML, как правило, выполняется в нотации диаграмм классов, описывающих основные понятия предметной области и связи между ними (рис. 1).

На основе построенной модели может быть выполнена генерация объектно-ориентированного кода путём прямого отображения элементов диаграмм классов (визуальной модели, созданной в терминах объектно-ориентированной парадигмы) в описание классов (в код

на целевом языке программирования). Однако такое решение вступает в противоречие с идеей абстрагирования от конкретной парадигмы реализации системы при построении модели, что в некоторых случаях приводит к усложнению модели при попытке описать специфику предметной области с использованием унифицированного языка моделирования. Кроме того, использование языков моделирования, оперирующих конструкциями той или иной парадигмы программирования, а не терминами предметной области, усложняет понимание модели экспертами.

На основе метамодели общецелевого языка описания моделей СМО попытаемся средствами UML построить языки проектирования компьютерных сетей и моделирования работы автозаправочной станции. Для упрощения рассмотрим лишь фрагмент метамодели, представленной на рис. 1: классы «Обслуживающее устройство», «Очередь» и двустороннюю ассоциацию «Обслуживает транзакты».

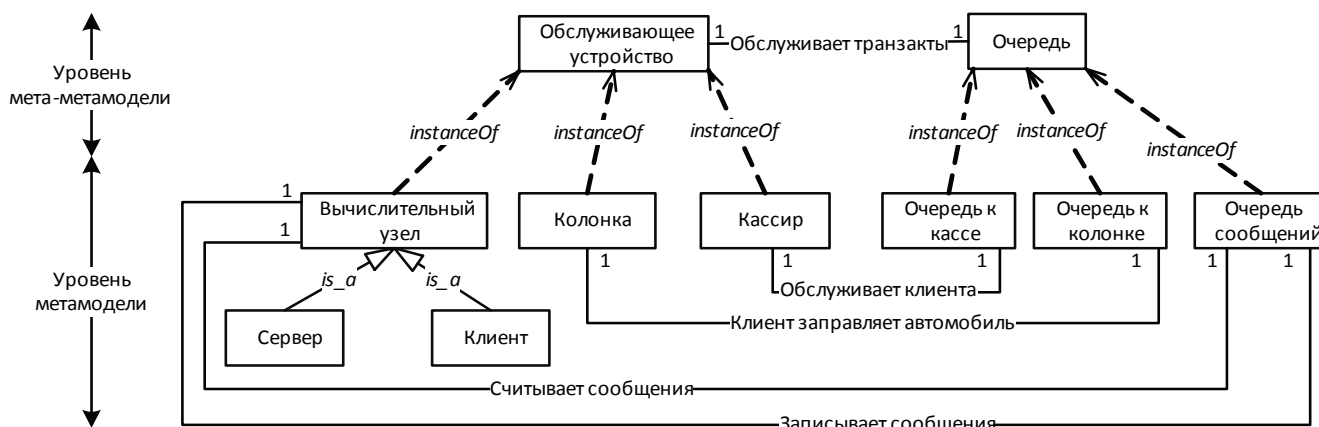


Рис. 2. Связь элементов метамодели «общецелевого» и предметно-ориентированных языков описания СМО

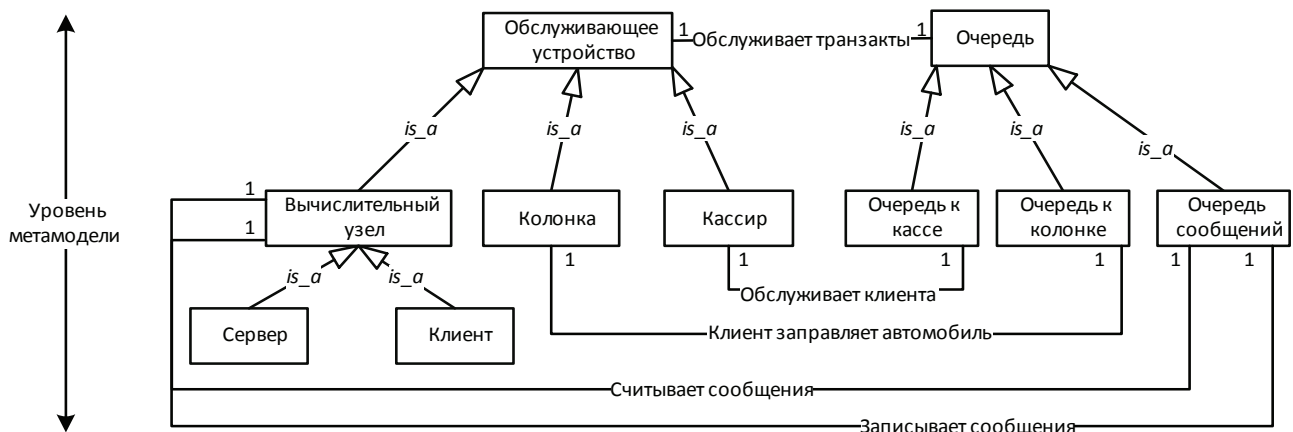


Рис. 3. Представление многоуровневой модели предметной области в виде иерархии классов

В языке проектирования компьютерных сетей этим конструкциям будут соответствовать следующие элементы: «Очередь сообщений», «Вычислительный узел» («Сервер», «Клиент»), «Считывает сообщения», «Записывает сообщения», а в языке моделирования работы автозаправочной станции: «Колонка», «Кассир», «Очередь к кассе», «Очередь к колонке», «Обслуживает клиента», «Клиент заправляет автомобиль». Ассоциации «Считывает сообщения», «Записывает сообщения», «Обслуживает клиента», «Клиент заправляет автомобиль» являются экземплярами отношения «Обслуживает транзакты» (рис. 2).

Как видно из данного примера, конструкции, которыми оперирует каждый из DSL, различны и не могут быть использованы одновременно в обеих предметных областях, поэтому для построения такого многоуровневого языка необходимо описать многоуровневую модель предметной области.

Одним из подходов к описанию многоуровневой модели стандартными средствами UML является построение иерархии классов перечисленных конструкций языка (рис. 3). Получившаяся в результате одноуровневая метамодель не отражает многоуровневый характер языка моделирования, т.к. в соответствии с ней конструкция языка «Очередь к кассе» является потомком конструкции

«Очередь», хотя между этими конструкциями должно быть установлено отношение инстанцирования. Такое решение приведёт к тому, что каждый из разрабатываемых DSL будет включать термины обеих предметных областей и конструкции общецелевого языка моделирования.

Другим вариантом описания многоуровневого языка моделирования является использование расширений UML: метаклассов, стереотипов, именованных значений и пр., которые позволяют добавить дополнительные уровни в описания метамodelей, что является искусственным расширением создаваемой метамодели (рис. 4). Такие расширения возможностей языка UML делают создаваемую модель ещё более запутанной и сложной для понимания пользователями – непрофессиональными программистами.

В настоящее время развивается подход к разработке информационных систем, основанный на парадигме программирования, ориентированного на языки (*Language-Oriented Programming* – LOP [5]), *мультимоделированию* (DSMM – *Domain-Specific Multimodeling* [3]). Разработка системы ведётся с использованием нескольких предметно-ориентированных языков. Этот подход существенно повышает производительность при разработке ИС, обеспечивает преемственность при переходе от одного эта-

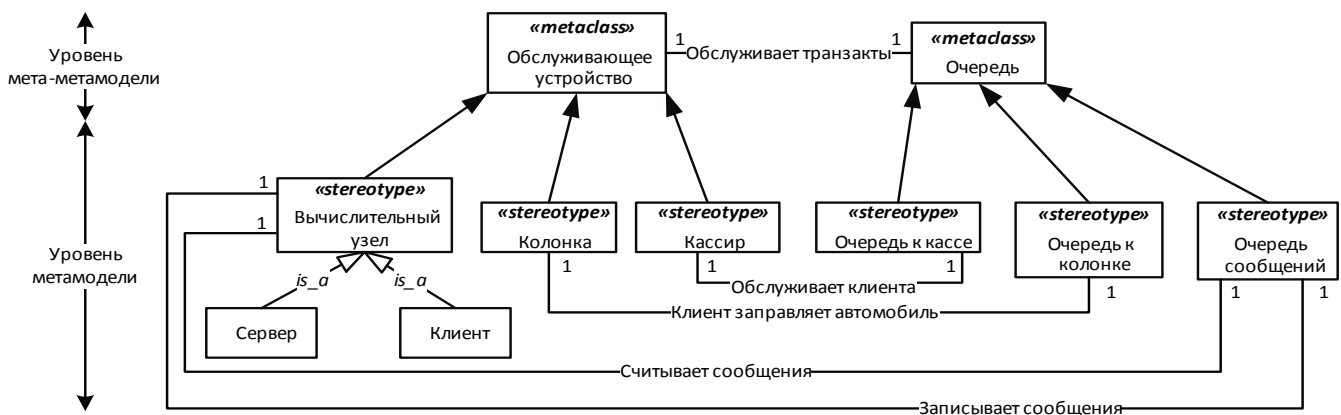


Рис. 4. Использование расширений языка UML для построения многоуровневой модели предметной области

па разработки к другому, облегчает её сопровождение, позволяет привлекать к работе экспертов в различных 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*. Определить правила, в соответствии с которыми выполняются преобразования моделей, предлагается также с помощью специальных языков – *языков трансформации моделей* (*Model Transformation Language – MTL*).

Привсех достоинств DSL у них есть один большой недостаток – сложность разработки: для каждой предметной области, а в некоторых случаях, даже для каждой задачи приходится создавать свой предметно-ориентированный язык. Кроме того, для работы с DSL необходимо создавать полнофункциональные DSL-редакторы. Для решения проблем предметно-ориентированного моделирования активно развивается новый тип программного обеспечения – *языковые инструментарии* (*language workbenches*). Языковой инструментарий, или *DSM-платформа*, – это инструментальное программное обеспечение, предназначенное для поддержки разработки и сопровождения предметно-ориентированных языков. Использование при создании DSL языкового инструментария значительно упрощает процесс создания языков.

Анализ существующих языковых инструментариев и подходов к созданию DSL позволил выявить ряд имеющихся в них *ограничений* [2]:

- отсутствие возможности изменения описания метаязыка, что позволило бы оперативно настраивать средства описания моделей на изменения предметных областей;
- в большинстве случаев внесение изменений в описание DSL возможно лишь после повторной генерации кода его редактора, что снижает производительность работы пользователей;

- отсутствие удобного интерфейса пользователя, который позволил бы пользователям-непрограммистам разрабатывать и настраивать DSL;
- наличие «избыточной» для конкретных пользователей функциональности, не используемой ими при создании DSL и моделей, что затрудняет выбор и изучение нужных средств пользователями;
- в большинстве рассмотренных систем отсутствуют средства описания и выполнения горизонтальных трансформаций моделей, что позволило бы объединить создаваемые модели, генерировать на основе построенных моделей код и документацию по задаваемым пользователями шаблонам, переводить модели на другие языки с целью экспорта во внешние системы.

При разработке системы MetaLanguage выполняется попытка снять некоторые ограничения существующих DSM-платформ [1].

Заключение

При реализации DSM-платформы MetaLanguage создан визуальный метаязык для описания DSL, который позволяет не только производить многоуровневое и мультиязыковое моделирование, но и выполнять динамическое изменение описаний моделей различных уровней иерархии без повторной генерации кода DSL-редактора, выполнять трансформацию описанных моделей в иные текстовые/графические нотации. Для создания моделей различных уровней иерархии используется единый пользовательский интерфейс, позволяющий работать как программистам-профессионалам, так и специалистам-экспертам в предметных областях.

Работа выполнена при поддержке РФФИ (проект № 12-07-00763-а).

Список литературы:

1. **Сухов А.О.** Инструментальные средства создания визуальных предметно-ориентированных языков моделирования // *Фундаментальные исследования*. – 2013. – № 4 (ч. 4). – С. 848-852.
2. **Сухов А.О.** Сравнение систем разработки визуальных предметно-ориентированных языков // *Математика программных систем: межвуз. сб. науч. ст.* – 2012. – Вып. 9. – С. 84-111.
3. **Hessellund A.** Domain-specific Multimodeling: PhD thesis. IT-University of Copenhagen, Software Development Group, 2009. 190 p.
4. **Soley R.** Model Driven Architecture / White Paper of the Object Management Group, 2000. [Электронный ресурс]. URL: <http://www.omg.org/~soley/mda.html> (дата обращения: 17.02.2013).
5. **Ward M.P.** Language Oriented Programming / Computer Science Department, Science Labs, South Rd, Durham, 1994. [Электронный ресурс]. URL: <http://www.cse.dmu.ac.uk/~mward/martin/papers/middle-out-t.pdf> (дата обращения: 11.02.2013).