

ВИЗУАЛЬНЫЕ ЯЗЫКИ И ЯЗЫКОВЫЕ ИНСТРУМЕНТАРИИ: МЕТОДЫ И СРЕДСТВА РЕАЛИЗАЦИИ*

Л.Н. Лядова¹, А.О. Сухов²

Рассматриваются методы реализации визуальных языков и DSM-платформ. Приведены описания различных подходов к реализации предметно-ориентированных систем и языковых инструментариев, основанных на метамоделировании, предназначенных для создания предметно-зависимых языков. Рассматриваются вопросы интеграции DSL-инструментариев и CASE-средств, возможности динамической адаптации созданных с их помощью информационных систем. Описывается архитектура и реализация языкового инструментария MetaLanguage, обеспечивающего возможность создания предметно-ориентированных языков, допускающих динамические изменения и возможность экспорта во внешние системы.

Введение

Основными проблемами инженерии программного обеспечения (ПО) становятся ускорение изменений, необходимость быстрой адаптации к новым условиям, обеспечение, с одной стороны, высокой степени гибкости, а с другой – функциональной надежности разрабатываемых информационных систем (ИС).

Современные корпоративные информационные системы создаются с использованием мощных инструментальных средств, *предметно-ориентированного ПО промежуточного уровня*, доступного не только программистам, но и другим специалистам, участвующим в разработке и эксплуатации ИС. Эти средства должны обеспечивать возможность настройки создаваемых систем на быстро меняющиеся потребности различных категорий пользователей и условия эксплуатации. Кроме того, так как крупные ИС носят интегрированный характер, охватывают все

* Работа выполнена при поддержке гранта РФФИ 10-01-00794

¹ 614070, г. Пермь, ул. Студенческая, 38, Пермский филиал ГУ-ВШЭ, lnl@nm.ru

² 614990, г. Пермь, ул. Букирева, 15, ПГУ, sukhov_psu@mail.ru

сферы деятельности предприятий, возникает необходимость привлекать к работе специалистов в различных предметных областях. Таким образом, инструментальные средства должны обеспечивать возможность работы не только профессионалов в области информационных технологий (ИТ), но и других категорий пользователей с использованием привычной для них терминологии, формальных моделей, используемых для описания данных и бизнес-процессов. Эти возможности обеспечиваются при использовании CASE-средств, основанных на технологии DSM (Domain Specific Modeling), и языковых инструментариев, позволяющих для каждой предметной области, каждого вида деятельности, поддерживаемого ИС, разрабатывать предметно-ориентированные языки (DSL, Domain Specific Languages).

1. Метамоделирование и DSL

Адаптируемость – одно из ключевых свойств ИС. Максимальные возможности адаптации обеспечиваются при использовании *технологии DSM с интерпретацией моделей*. При этом программное обеспечение ИС выступает в качестве интерпретатора, а «поведение» системы определяется созданными при ее разработке моделями, описывающими все стороны деятельности предприятия, автоматизируемые этой системой.

При создании ИС нельзя ограничиваться созданием только одной модели: если система сложная, то учет всех ее характеристик в одной модели приведет к чрезвычайной ее громоздкости. Таким образом, приходим к необходимости разработки *нескольких моделей*, которые позволяют описать разные стороны функционирования системы. Кроме того, различные модели могут описывать систему с разной степенью абстракции. Может потребоваться преобразование моделей, их приведение к единому представлению, унифицированному для всех областей, «понятному» для исполнителя, в качестве которого выступает программное обеспечение ИС.

Модель системы представляет собой абстрактное описание на некотором формальном языке характеристик системы, важных с точки зрения цели моделирования, ее поведения. Технология DSM обеспечивает *моделирование в терминах предметной области*. Для решения каждой задачи может применяться свой язык моделирования, в котором используются исключительно понятия и отношения из соответствующей предметной области ИС. При этом в системе создается несколько уровней моделей: данные, хранящиеся в базе данных (БД) системы, представляют собой *модель состояния предметной области* ИС; их описание, обеспечивающее интерпретацию данных или генерацию кода для работы с ними, – 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, – здесь используется уже *мета-мета-модель*. Таким образом, при использовании современных CASE-систем строятся метамодели, которые определяют предметно-зависимые языки моделирования. Для создания этих языков также необходимы соответствующие инструментальные средства («мета-CASE-средства») – *языковой инструментарий*, называемый также *DSL-инструментарием*. Мета-CASE- и CASE-средства могут быть объединены в одной CASE-системе, интегрирующей «традиционные» CASE-средства и DSL-инструментарии, с помощью которых создаются метамодели (рис. 1).

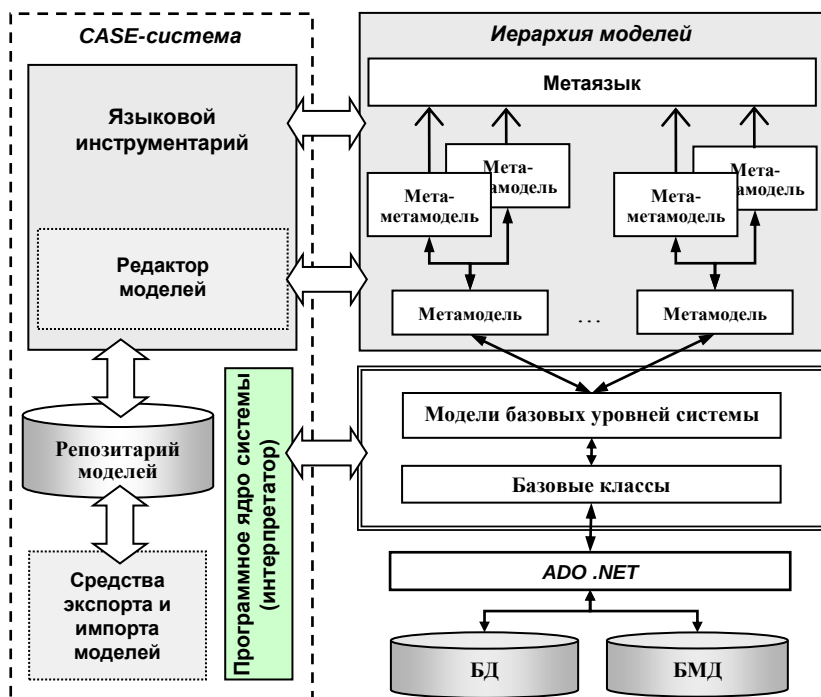


Рис. 1. Упрощенная структура информационной системы, основанной на DSM-платформе с интерпретацией (мета)моделей

При работе в режиме интерпретации моделей реализация нескольких уровней моделирования в ИС приводит к значительной потере производительности, но позволяет построить систему с уникальными возможностями динамической адаптации. Для того чтобы данный подход

оказался применимым на практике, необходимо чтобы мета-метамодель была максимально выразительной, что обеспечивает высокую степень гибкости ИС, возможности ее настройки на потребности пользователей.

Фактически создание системы сводится к разработке предметно-ориентированных языков (DSL) – мета-метамodelей ИС. Использование предметно-зависимых языков 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.

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

В последнее время на стандарт языка моделирования претендует UML, 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 ведется 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 определенной предметной области.

Язык, используемый для создания других языков, будем называть *метаязыком*.

Процесс создания модели может быть *итеративным*: создав некоторый язык, мы можем использовать его как метаязык для создания другого языка, который, в свою очередь, также может быть использован как метаязык, и т.д.

2. Языковые инструментарии

Существует большое число средств для разработки графических редакторов DSL с возможностью определения собственных графических нотаций. Такими средствами являются MetaEdit+, MS DSL Tools, Eclipse

GMF, State Machine Designer, REAL-IT, UFO-toolkit и пр. Ключевым аспектом DSM-подхода является создание инструментальных средств (*DSM-платформ*), поддерживающих тот вариант использования визуального моделирования, который оптимален для конкретной области (здесь DSM-платформа – инструментальное средство разработки визуальных DSL).

Система UFO-toolkit [Маторин, 2004] позволяет проводить системно-объектный анализ, используя концептуальные классификационные модели. Инструментарий UFO-toolkit предназначен для описания бизнес-процессов помощью UFO-элементов. Он позволяет представлять структуру и функции моделируемых систем; обеспечивает учет семантики предметной области. Данная система, в отличие от всех остальных, позволяет проводить имитационное моделирование на основе созданных моделей. Однако существенным недостатком этой системы является то, что используемые в системе нотации не являются общепризнанным стандартом, поэтому использование созданных моделей в других системах невозможно.

Технология REAL-IT [Иванов, 2004] основана на применении языка UML. Разработка ИС сводится к моделированию БД и пользовательского интерфейса с помощью CASE-пакета. По этим моделям автоматически генерируется приложение. Возможность генерации обеспечивается стандартизацией интерфейса пользователя и отсутствием нетривиальной логики обработки данных. В противном случае сгенерированный код приходится дополнять кодом, написанным «вручную».

Системы REAL-IT и UFO-toolkit при создании ИС позволяют использовать только встроенные языки моделирования. Это значительно ограничивает настраиваемость этих систем.

Благодаря использованию подхода, основанного на интерпретации метамodelей, а не на генерации, технология MetaEdit+ [Mazanek, 2008], [Tolvanen, 2008], позволяет вносить изменения в описание DSL во время работы системы. Система позволяет работать, как с языками, так и с метаязыками единообразно, используя один и тот же инструментарий. К недостаткам MetaEdit+ можно отнести то, что эта DSM-платформа для экспорта моделей использует свой собственный формат файлов (MXT), а это сказывается на открытости технологии.

Технологии DSL Tools и GMF помимо визуального редактора метамodelей предоставляют в распоряжение пользователя развитые среды программирования MS Visual Studio и Eclipse соответственно. Благодаря этому существует возможность «ручной» доработки кода на языках высокого уровня, но это может привести к несогласованности диаграмм и исходного кода. Средства State Machine Designer, по существу, являются

лишь надстройкой над DSL Tools, устраняющей ряд ее недостатков. Однако при этом State Machine Designer позволяет создавать DSL лишь с помощью диаграмм деятельности UML, что значительно ограничивает круг решаемых задач. Язык «ручной» доработки сгенерированного кода остался достаточно сложным для того, чтобы появилась возможность работы с ним пользователей-непрограммистов.

В отличие от других технологий в MPS реализован текстовый способ определения DSL, основанный на принципе вложенных ячеек. Это не очень удобно, поскольку текст не обладает достаточной выразительностью.

DSM-платформа GMF является самой мощной из рассмотренных. Однако ее использование затруднено отсутствием документации, громоздкостью и сложностью, а также частым выходом новых версий. Фактически, GMF находится еще в стадии интенсивного развития.

Нередки случаи, когда DSL становятся частью других приложений. Например, специально созданный язык для описания бизнес-процессов может использоваться в системе документооборота. Поэтому еще одной важной характеристикой DSM-платформ является их *отчуждаемость* от среды разработки. Технологии DSL Tools, GMF, MPS сильно связаны с платформами разработки – MS Visual Studio, Eclipse, IntelliJ-IDEA соответственно, поэтому созданные с помощью данных инструментариев языки не могут быть использованы в сторонних приложениях.

Ни одна из рассмотренных выше технологий не предоставляет возможности создавать как визуальные, так и текстовые DSL. Кроме того, ни одна из технологий, за исключением MetaEdit+, не позволяет создавать динамически настраиваемые языки.

Устранение отмеченных недостатков – цель разработки метаязыка и DSL-инструментария MetaLanguage [Лядова, 2008].

3. Реализация MetaLanguage

Одним из подходов к созданию динамически изменяемых языков является хранение описания языка в базе метаданных (*репозитории*) системы, доступной в процессе ее функционирования.

Благодаря этому появляются возможности

- динамической настройки языка на меняющиеся условия эксплуатации и потребности пользователя;
- работы в привычных для пользователя терминах предметной области;
- повторного использования DSL в схожих проектах;
- интеграции нескольких языков в одной системе.

Для создания языка необходимо построить его модель, а для этого

также нужен некоторый язык – *метаязык*, на базе которого разработчик сможет создавать новые языки либо вносить изменения в уже существующие [Лядова, 2009]. Если описание самого метаязыка будет также представлено в виде метаданных, то появится возможность изменения и этого описания [Сухов, 2008]. Кроме того, процесс создания метамодели станет итеративным: определив метамодель качества метаязыка, разработчик может использовать эту метамодель для построения других метамodelей, и этот процесс может быть бесконечным.

В разрабатываемый метаязык необходимо интегрировать как текстовое, так и графическое представление: *графическое представление* удобно использовать при описании моделей предметной области, а *текстовое* – при создании БД (при определении ограничений, описании шаблонов отчетов и т.д.). *Базовыми элементами*, создаваемого метаязыка являются сущность, отношения, ограничения.

По аналогии с традиционными формальными языками *визуальные языки* можно определить как *совокупность наборов графических символов и формализованных правил для построения визуальных моделей*. Графовые грамматики представляют традиционный подход к *формальному описанию синтаксиса* визуальных языков. Наиболее подходящей моделью *внутреннего представления* разрабатываемых языков в системе в ходе ее функционирования, являются *мультиграфы*, несмотря на простоту их топологии. Все операции, выполняемые при создании метамodelей, можно описать в терминах *ориентированных помеченных мультиграфов*.

Для работы с объектами метаязыка спроектирована *среда разработки*, которая включает графический редактор, браузер объектов, репозиторий и валидатор [Сухов, 2009]. Среда включает в себя главное меню и панель инструментов с функциями создания/изменения/удаления объектов и всего проекта, настройки графических параметров: цветов, шрифтов, толщины линий и пр.

Процесс создания DSL начинается с *построения его метамодели* (задаются основные конструкции языка, отношения между ними, ограничения, накладываемые на сущности и отношения метамодели). После построения метамодели разработчик получает в распоряжение готовый расширяемый настраиваемый визуальный язык моделирования. Используя полученный DSL, пользователь может *создавать модели*, содержащие объекты, описывающие конкретные сущности предметной области и связи между ними. После построения модели необходимо проверить, удовлетворяет ли она ограничениям, – *выполнить валидацию* модели.

Заключение

В работе представлен проект DSM-платформы, которая обеспечивает все этапы создания ИС от разработки предметно-ориентированных языков моделирования до создания на их основе моделей, которые могут использоваться в конкретной реализации ИС либо могут быть экспортированы в другую систему.

Модификация метамodelей может производиться на любом этапе создания DSL. При этом после внесения изменений в метамodelь система автоматически внесет все необходимые изменения в модели, созданные с помощью этой метамodelи.

Описанный выше подход реализован в виде библиотеки классов. Основная задача библиотеки заключается в обеспечении работы с репозитарием системы.

Список литературы

- [Иванов, 2004] Иванов А.Н. Технологическое решение REAL-IT: создание информационных систем на основе визуального моделирования // Сб. «Системное программирование» / Под ред. проф. А.Н. Терехова и Д.Ю. Булычева. – СПб: Изд. СПбГУ, 2004. С. 89-100.
- [Лядова, 2009] Лядова Л.Н., Сухов А.О. Метаязык построения визуальных языков моделирования // Технологии Microsoft в теории и практике программирования / Материалы конф. Нижегородский ун-т. Нижний Новгород, 2009. С. 267-273.
- [Лядова, 2008] Лядова Л.Н., Сухов А.О. Языковой инструментарий системы MetaLanguage // Математика программных систем: Межвуз. сб. научн. тр. / Перм. ун-т. Пермь, 2008. С. 40-51.
- [Маторин, 2004] Маторин В.С., Маторин С.И., Попов А.С. CASE-инструментарий UFO-toolkit. Автоматизация построения UFO-моделей // Проблемы программирования. 2004. № 2-3. С. 144-149.
- [Сухов, 2008] Сухов А.О. Предметно-ориентированный язык в адаптируемых информационных системах // Технологии Microsoft в теории и практике программирования / Материалы конф. Новосибирск, 2008. С. 25-26.
- [Сухов, 2009] 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.
- [Mazanek, 2008] Mazanek S. Visual Languages. MetaEdit+ : [Электронный документ] (<http://visual-languages.blogspot.com/2007/11/metaedit.html>). Проверено 21.05.2008.
- [Tolvanen, 2008] Tolvanen J.-P., Rossi M. MetaEdit+: defining and using domain-specific modeling languages and code generators : [Электронный документ] (<http://portal.acm.org/citation.cfm?id=949365>). Проверено 21.05.2008.