

Л.Н. Лядова, А.О. Сухов

Пермский государственный университет  
LNLyadova@mail.ru, sukhov\_psu@mail.ru

## **ЯЗЫКОВОЙ ИНСТРУМЕНТАРИЙ СИСТЕМЫ METALANGUAGE**

С увеличением числа требований к адаптируемым информационным систем (ИС) их разработка становится все более сложной. Для создания ИС разработчику приходится использовать множество языков: языки программирования общего назначения для описания функциональности приложения, XML для передачи данных между приложениями, SQL для создания запросов к базе данных (БД) и т.п.

Даже использование специальных программно-технических средств, называемых CASE-средствами, которые значительно упрощают процесс разработки ИС, не решает этой проблемы. Поскольку сами они обладают рядом недостатков.

Часто при работе с CASE-средствами появляется необходимость расширения существующего языка создания ИС. Однако ни одно CASE-средство, за редким исключением, не предоставляет возможности настройки языка моделирования.

Еще одним недостатком большинства современных CASE-средств и языков, используемых для создания ИС, является их универсализм: они оперируют не терминами предметной области, а конструкциями тех средств, с помощью которых создается система.

На практике часто возникает проблемы трансформации модели, т.е. преобразования модели с одного языка моделирования в другой. Эта проблема возникает, когда появляется необходимость представления одной и той же графической модели с помощью различных нотаций. Например, представление модели предметной области с помощью диаграмм «Сущность-Связь» и с помощью диаграмм классов UML. Проблема трансформации актуальна, когда с одной и той же моделью работают сразу несколько человек, причем каждый из них желает работать с наиболее удобным для него представлением модели.

Все эти недостатки CASE-средств приводят к тому, что в настоящее время все большее число разработчиков используют лишь часть предоставляемой ими функциональности, например, лишь возможность генерации схемы БД.

Чтобы решить эти проблемы необходимо предоставить пользователям возможность изменять описание языков создания ИС и работать в терминах предметной области. Это можно сделать за счет создания и использования в процессе разработки ИС языков моделирования, созданных специально для предметной области, в которых будет функционировать система. Такие языки называются *предметно-ориентированными языками* (DSL), а их использование – *предметно-ориентированным моделированием* (DSM). В последнее время DSM-подход получает все большую популярность. DSL применяются на различных этапах создания ИС: при описании модели предметной области и бизнес-процессов [2, 3], при проектировании пользовательского интерфейса [1, 4] и др.

Для построения языков моделирования необходимо создать средства их разработки, которые позволили бы в удобной форме определять конструкции языка, экспортировать и импортировать описание языков из других систем.

Примером такого инструментального средства является система MetaLanguage.

## Визуальный метаязык

В промышленных областях мы часто встречаемся с тем, что изучение объекта производится посредством создания его модели. В конце 60-х годов прошлого века идея создания моделей пришла и в компьютерные науки.

Под *моделью* в этой статье будем понимать описание объекта (процесса) на каком-либо формальном языке, составленное с целью изучения его свойств. А процесс создания моделей будем называть *моделированием*.

Поскольку визуальные языки являются формальными, то они строятся на основе некоторой модели. Назовем *метамоделью* – модель языка моделирования. Язык, используемый для создания других языков моделирования, будем называть *метаязыком*.

Одним из подходов к устранению недостатков языков моделирования традиционных CASE-средств является создание и использование в процессе разработки DSL метаязыка. Благодаря этому появляется возможность:

- 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;
- трансформации моделей с одного языка моделирования в другой.

Система MetaLanguage представляет собой среду для создания и использования визуальных динамически настраиваемых предметно-ориентированных языков моделирования.

Базовыми элементами, которые использует MetaLanguage для создания метамоделей являются: сущность, отношение, ограничение. Рассмотрим эти элементы подробнее.

### *Сущность*

*Сущность* – это какая-либо конструкция создаваемого языка моделирования. Сущности характеризуются

- именем, однозначно идентифицирующим сущность в рамках метамоделей;
- количеством экземпляров сущности, которое может быть создано при построении модели;
- набором атрибутов сущности;
- набором операций над сущностью;
- набором ограничений, накладываемых на сущность;
- флагом уникальности, определяющим пределы уникальности имен экземпляров сущности.

Рассмотрим некоторые из этих свойств более подробно.

*Количество экземпляров сущности* определяет, сколько экземпляров данной сущности может быть создано при построении модели. Количество экземпляров задается целым числом из интервала  $[0, \infty)$ . Если значение этой характеристики сущности равно нулю, то при построении модели данная сущность не будет указана в списке сущностей, предлагаемых для создания. Если значение характеристики равно бесконечности, то допускается создание произвольного числа экземпляров данной сущности.

*Атрибут* – это именованное свойство сущности, включающее описание множества значений, которые могут принимать экземпляры данной сущности.

Атрибут имеет

- имя, которое уникальным образом идентифицирует его в рамках данной сущности;
- тип, определяющий множество значений, которые может принимать атрибут, а также набор допустимых над атрибутом операций;

- значение по умолчанию, которое будет выбрано в качестве основного значения атрибута, если последнее не задано;
- описание, которое содержит некоторую дополнительную информацию об атрибуте.

Сущность может иметь любое число атрибутов, либо не иметь их вовсе.

*Операция* – это абстракция действий, которые можно выполнять над сущностью. В большинстве случаев применение операции приводит к тому, что сущность меняет свое состояние.

Операция включает в себя:

- имя, которое однозначно определяет операцию в рамках данной сущности;
- параметры операции;
- значения по умолчанию для параметров, которые в случае отсутствия основных значений будут использоваться при вызове операции;
- тип возвращаемого значения;
- описание, содержащее дополнительную информацию об операции.

Сущность может иметь произвольное число операций либо не иметь их вовсе.

### ***Отношение***

Конструкции визуальных языков в редких случаях существуют автономно, чаще всего они каким-либо образом взаимосвязаны друг с другом, поэтому при создании метамодели важно не только определить основные конструкции создаваемого языка, но и правильно задать связи между ними.

*Отношение* используется для обозначения физической или концептуальной связи между сущностями.

Любое отношение характеризуется

- именем, которое однозначно задает отношение в рамках данной модели;
- кратностью, которая определяет сколько экземпляров сущности может участвовать в отношении;
- флагом уникальности, определяющим пределы уникальности имен экземпляров отношений;
- ограничениями, накладываемыми на отношение.

В MetaLanguage выделяются следующие виды отношений: наследование, ассоциация, агрегация.

### **Наследование**

*Наследование* – это отношение между общим объектом (суперклассом, родителем) и конкретным объектом (подклассом, потомком).

Потомок наследует все атрибуты и операции родителя. Помимо родительских, он может иметь и свои собственные атрибуты и операции.

Объект может иметь лишь одного родителя и неограниченное число потомков, т.е. кратность такого отношения 1:M.

Как уже говорилось ранее, при создании сущности задается количество ее экземпляров, которое определяет сколько экземпляров сущности может быть создано в модели. Абстрактной называется сущность, которая не имеет экземпляров и в построении модели не участвует. Чаще всего абстрактные сущности играют роль родителей и используются для простоты модификации метамодели и наглядности.

На рис. 1 представлен фрагмент метамодели диаграмм «Сущность-Связь». Метамодель содержит сущности: «Абстрактный класс», «Атрибут», «Связь», «Сущность». Для сокращения диаграммы на рисунке не представлены допустимые над сущностями операции.

Атрибутами сущности «Абстрактный класс» являются «Имя», идентифицирующее экземпляр сущности и «Описание», содержащее дополнительную информацию об экземпляре сущности.

Сущность «Атрибут» имеет следующие атрибуты: «Имя», «Тип», «Ограничения», «Описание». Атрибут «Ограничения» определяет ограничения целостности, задаваемые на атрибут в ERD-модели.

Сущность «Абстрактный класс» является абстрактной, т.е. при создании модели экземпляры этой сущности создать нельзя. Сущность «Абстрактный класс» выступает в роли родителя для сущностей «Сущность» и «Связь». Обе дочерние сущности наследуют все атрибуты и операции родительской, собственных атрибутов и операций эти сущности не имеют.



Рис. 1. Фрагмент метамодели диаграмм «Сущность-Связь»

### Ассоциация

*Ассоциация* – это структурное отношение, определяющее, что сущности одного типа некоторым образом связаны с сущностями другого типа.

Ассоциация является двунаправленной и может быть проведена в любом направлении. Является допустимым случаем, когда оба конца ассоциации принадлежат одной сущности. Это означает, что некоторый экземпляр сущности может быть связан с другим экземпляром этой же сущности.

Помимо описанных ранее базовых характеристик отношений, существует дополнительная, применяемая лишь к ассоциации – роль. Сущности связанные ассоциацией играют в ней определенную роль. *Роль* – это имя, которое уникальным образом идентифицирует один из концов ассоциации.

В ассоциации может участвовать произвольное число экземпляров сущности как с одной, так и с другой стороны, т.е. в общем случае кратность этого отношения М:М.

По числу сущностей, участвующих в ассоциации, выделяют бинарные, тернарные и ассоциации более высокого порядка. Бинарная

ассоциация – это ассоциация, в которой участвуют две сущности. Тернарная ассоциация соединяет три сущности.

На рис. 1 представлены две ассоциации, которые изображены в виде неориентированных дуг. Первая из них соединяет сущности «Связь» и «Сущность», это говорит о том, что в ERD-моделях между экземплярами этих сущностей можно провести равноправную связь. Вторая ассоциация связывает сущность «Сущность» с самой собой, это позволяет в ERD-моделях любому экземпляру «Сущности» иметь родительскую «Сущность».

### **Агрегация**

*Агрегация* – это вид ассоциации, который моделирует неравноправное отношение типа «часть-целое».

Основным отличием агрегации от простой ассоциации является то, что последняя отражает отношение между равноправными сущностями, в то время как в агрегации одна из сущностей – главная, а другие – зависимые. Отличительными чертами агрегации также является и то, что этот вид отношения всегда является ориентированным, кратность этого отношения 1:М и его концы не могут принадлежать одной сущности.

В метамодели, представленной на рис. 1, между сущностями «Абстрактный класс» и «Атрибут» установлено отношение агрегации (на диаграмме этот вид отношений представлен дугой с наконечником в виде ромба). В результате чего в ERD-моделях экземпляры сущностей «Сущность» и «Связь» могут быть соединены агрегацией с экземплярами сущности «Атрибут».

При этом при удалении экземпляров сущностей «Сущность» и «Связь», автоматически будут удалены все принадлежащие им экземпляры сущности «Атрибут».

### **Ограничение**

На практике достаточно часто встречаются случаи, когда на сущности и связи между ними накладываются некоторые ограничения.

Если правила соединения диаграмм задают синтаксис визуального языка, то ограничения определяют его семантику. Часть ограничений задается через структуру метамодели, а часть через непосредственное их задание на некотором языке.

Все ограничения, накладываемые на метамодель можно разделить на две группы: ограничения, накладываемые на сущности, и ограничения, накладываемые на отношения.

Проведем детальное рассмотрение каждой из этих групп.

## Ограничения, накладываемые на сущности

Ограничения, накладываемые на сущности могут быть одного из следующих видов:

- ограничения на уникальность имени экземпляра сущности;
- ограничения на количество экземпляров сущности в модели;
- ограничения на значения атрибутов экземпляров сущности.

*Имя экземпляра сущности* может быть уникальным в рамках метамодели, в рамках модели либо не быть уникальным. Уникальность в рамках метамодели означает, что во всех моделях, созданных на основе этой метамодели имя экземпляра сущности встретиться всего лишь один раз.

Уникальность в рамках модели означает, что имя экземпляра сущности будет уникальным лишь в рамках той модели, которой эта сущность принадлежит. Примером такого ограничения может служить условие на уникальность имени сущности «Сущность» в рамках конкретной модели ERD-диаграммы.

*Ограничение на количество экземпляров сущности* в модели задается указанием количества экземпляров при создании сущности. Так экземпляры абстрактных сущностей, имеющих значение свойства «количество» равное нулю, не будут участвовать при построении модели. Если значение этого свойства равно единице, то в модели можно создать лишь единственный экземпляр сущности этого типа.

Наиболее важными с точки зрения задания семантики визуальных языков являются *ограничения, накладываемые на значения атрибутов сущности*. Такие ограничения задаются в виде троек

*<Имя атрибута><Знак><Значение>*.

Здесь в роли «значения» может выступать константа, значение атрибута какого-либо экземпляра сущности модели либо некоторая функция над значениями атрибутов экземпляров сущностей.

Ограничение этого типа может быть наложено на атрибут «Дата рождения» сущности «Человек» при построении метамодели «Генеалогическое дерево», т.к. дата не может превышать текущее время. Тогда такое ограничение должно иметь вид:

*Дата рождения <= Now()*.

## Ограничения, накладываемые на отношения

Все ограничения, накладываемые на отношения можно разделить на следующие группы:

- ограничения на уникальность имени экземпляра отношения;
- ограничения на типы соединяемых экземпляров сущностей;
- ограничение на кратность отношений;

- ограничения на значения атрибутов соединяемых экземпляров сущностей.

*Ограничения на уникальность имени экземпляра отношения* аналогично ограничениям на уникальность имени экземпляра сущности и могут принимать одно из значений: уникально в рамках метамодели, уникально в рамках модели, неуникально.

*Ограничения на типы соединяемых экземпляров сущности* определяются структурой метамодели. Эти ограничения задают правила соединения экземпляров сущностей различных типов. Так, например, на рис. 1 отсутствуют ассоциация, соединяющая сущность «Связь» саму с собой, это означает, что никакие два экземпляра сущности «Связь» соединить друг с другом невозможно.

*Ограничения на кратность отношений* задаются при создании последних. Так отношение наследования и агрегация допускает в общем случае лишь кратность 1:M, которая может быть уточнена изменением кратности лишь для второй сущности. Ассоциация допускает кратность M:M с возможностью ее уточнения.

Если в моделях ERD-диаграмм необходимо указать, что число экземпляров сущности «Атрибут», принадлежащих одному экземпляру сущности «Сущность» не должно превышать пяти, то при создании ассоциации между сущностями «Сущность» и «Атрибут» необходимо установить кратность «M:1..5».

*Ограничение на значения атрибутов соединяемых экземпляров сущностей* несут на себе наибольшую семантическую нагрузку.

Ограничение такого вида может быть задано на значения атрибута «Дата рождения» соединяемых экземпляров сущностей «Человек», поскольку даты рождения родителей не могут превышать даты рождения ребенка.

## **Процесс создания предметно-ориентированных языков с помощью системы MetaLanguage**

Описав основные конструкции метаязыка, перейдем к рассмотрению того, каким же образом в системе MetaLanguage создаются визуальные предметно-ориентированные языки моделирования (рис. 2).

Процесс описания DSL начинается с построения его метамодели. Для этого необходимо задать основные конструкции создаваемого языка, определить отношения между ними, задать 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, которые были на нее наложены. Для этого необ-



**Рис. 2. Процесс создания/модификации языка предметной области с помощью системы MetaLanguage**

ходимо выполнить валидацию созданной модели. Если какие-либо из ограничений не выполняются, то пользователь будет проинформирован об этом.

Используя генератор, пользователь может сохранить построенные метамодели и модели в виде XML-файлов, либо сгенерировать на их основе документацию к системе.

Заметим, что при внесении изменений в метамодель система автоматически внесет все необходимые изменения в модели, созданные с помощью этой метамодели.

## Заключение

В данной статье представлено описание языкового инструментария системы MetaLanguage. Данный метаязык позволяет создавать языки моделирования, которые могут быть настроены на меняющиеся условия эксплуатации и потребности пользователя без регенерации программного кода системы. При этом пользователь может самостоятельно определять собственный язык для создания ИС или изменять уже существующие языки. Кроме того, создаваемые языки предоставляют возможность работы в терминах предметной области. Аналитикам – непрофессиональным программистам это позволяет разрабатывать несложные программы, например, описывать бизнес-процессы. А профессиональные программисты благодаря этому могут сосредоточиться на творческом процессе разработки, а не на переводе высокоуровневых понятий предметной области в низкоуровневый программный код.

MetaLanguage позволяет интегрировать несколько предметно-ориентированных языков моделирования в рамках одной системы. Такой подход предоставляет возможность строить системы, работающие на стыке предметных областей.

Создаваемые DSL обладают высокой степенью согласованности с метаязыком. При внесении изменений в метаязык система автоматически произведет все необходимые изменения в языках, созданных на основе этого метаязыка. Таким образом, все языки и модели всегда находятся в актуальном состоянии.

## Библиографический список

1. *Лядова Л.Н., Сухов А.О.* Предметно-ориентированный язык построения выражений для компонента автоматической генерации запросов // Технологии Microsoft в теории и практике программирования / Материалы межвуз. конкурса-конференции. Политехнический ун-т. СПб, 2008. С. 164-165.
2. *Correal D., Casallas R.* Using Domain Specific Languages for Software Process Modeling: [Электронный документ] (<http://www.dsmforum.org/events/DSM07/papers/correal.pdf>). Проверено 21.05.2008.

3. *Roser S., Lautenbacher F.* Generation of Workflow Code from DSMs: [Электронный документ] (<http://www.dsmforum.org/events/DSM07/papers/rosen.pdf>). Проверено 21.05.2008.
4. *Safa L.* The Making Of User-Interface Designer A Proprietary DSM Tool: [Электронный документ] (<http://www.dsmforum.org/events/DSM07/papers/safa.pdf>). Проверено 21.05.2008.