

Метаязык построения визуальных языков моделирования

Л.Н. Лядова, А.О. Сухов

Пермский государственный университет

LNLyadova@gmail.com, Sukhov_PSU@mail.ru

Введение

С увеличением числа требований к адаптируемым информационным системам (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.

Чтобы решить эти проблемы необходимо предоставить пользователям возможность изменять описание языков создания ИС и использовать в процессе разработки языки моделирования, созданные специально для предметной области, в которой 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 [1, 2], при проектировании пользовательского интерфейса [3] 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 разрабатываемая на кафедре Математического обеспечения вычислительных систем Пермского государственного университета 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 метаязыка, позволяющего описывать языки моделирования, используемые в процессе создания ИС. Благодаря этому появляется возможность:

- настройки языков на меняющиеся условия эксплуатации и потребности пользователя;
- работы в привычных для пользователя терминах предметной области;
- трансформации моделей из одного языка моделирования в другой;

- повторного использования DSL в схожих проектах;
- интеграции нескольких языков в одной системе.

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

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

1. Сущность

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

- именем, однозначно идентифицирующим сущность в рамках метамодели;
- 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.

2. 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.

2.1. 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:М.

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 используется для простоты модификации метамодели и наглядности.

2.2. 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.

2.3. Агрегация

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

Основным отличием агрегации от 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 ориентирован, его кратность – 1:М, а концы не могут принадлежать одной сущности.

3. 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.

3.1. 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.

Ограничение на количество экземпляров сущности в модели задается указанием количества экземпляров при создании сущности. Так экземпляры абстрактных сущностей, имеющих значение свойства «количество» равно 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.

3.2. 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.

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

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

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

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.

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.

При создании системы MetaLanguage были использованы следующие технологии и программные средства: программная платформа – .NET Framework; компонентная технология – .NET; среда разработки – MS Visual Studio 2005, язык программирования – C#; метод доступа к данным – ADO.NET; СУБД – MS SQL Server.

СПИСОК ЛИТЕРАТУРЫ

1. *Correal D., Casallas R. Using Domain Specific Languages for Software Process Modeling: [Электронный документ]* (<http://www.dsmforum.org/events/DSM07/papers/correal.pdf>).
2. *Roser S., Lautenbacher F. Generation of Workflow Code from DSMs: [Электронный документ]* (<http://www.dsmforum.org/events/DSM07/papers/roses.pdf>).
3. *Safa L. The Making Of User-Interface Designer A Proprietary DSM Tool: [Электронный документ]* (<http://www.dsmforum.org/events/DSM07/papers/safa.pdf>).