

А.О. Сухов, Л.Н. Лядова

Национальный исследовательский университет  
«Высшая школа экономики»  
(Пермский филиал)

ASuhov@hse.ru, LLyadova@hse.ru

### **АЛГОРИТМЫ ФУНКЦИОНИРОВАНИЯ И ОПЕРАЦИИ НАД МОДЕЛЯМИ В СИСТЕМЕ METALANGUAGE<sup>3</sup>**

Система MetaLanguage – это языковой инструментарий, предназначенный для создания визуальных динамически настраиваемых предметно-ориентированных языков моделирования [2, 3, 7]. Процесс создания и трансформации моделей предметной области с использованием системы MetaLanguage схематично представлен на рис. 1.

Как видно из рисунка, для создания нового предметно-ориентированного языка следует задать основные конструкции языка: определить сущности и отношения между ними, задать ограничения, налагаемые на сущности и отношения метамодели. При этом конкретный синтаксис языка задается в процессе описания абстрактного синтаксиса.

После построения метамодели (модели языка моделирования) разработчик получает в распоряжение готовый расширяемый настраиваемый визуальный язык моделирования. Используя полученный DSL, пользователь может создавать модели, содержащие конкретные объекты предметной области и связи между ними.

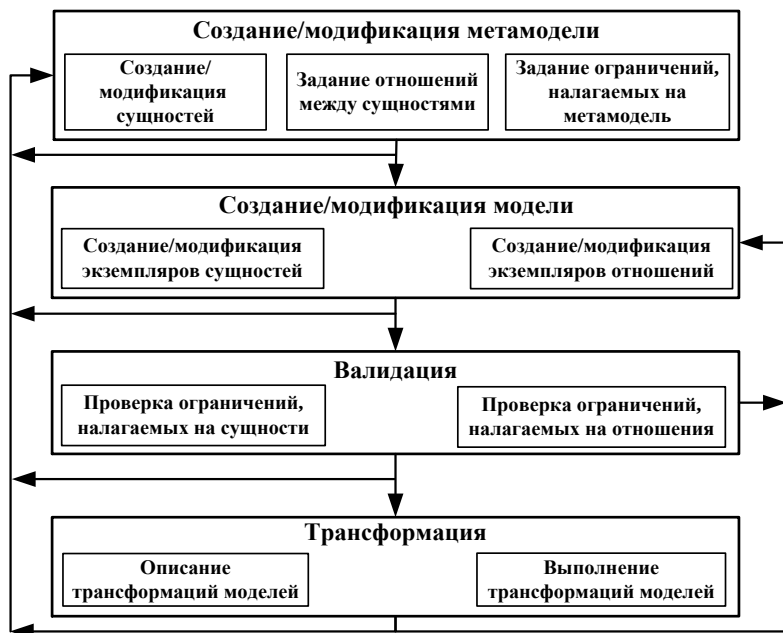
Если в процессе построения модели оказалось, что в описании языка были допущены неточности, то пользователь может на любом этапе вернуться к его описанию и внести необходимые изменения без повторной генерации кода и перезапуска системы, при этом система автоматически выполнит все необходимые каскадные изменения в соответствующих моделях.

---

<sup>3</sup> Работа выполнена при поддержке РФФИ (проект № 12-07-00763-а)

© Сухов А.О., Лядова Л.Н., 2013

После построения модели необходимо проверить, удовлетворяет ли она ограничениям метамодели, с помощью которой описана эта модель. Для этого следует выполнить валидацию созданной модели. Если какие-либо из ограничений не выполняются, то пользователь будет об этом проинформирован. Если ошибок в созданной модели не обнаружено, пользователь может приступить к описанию трансформаций модели, т.е. выполнить ее преобразование в эквивалентную модель, описанную в иной текстовой/графической нотации [1, 6].



**Рис. 1. Процесс создания/модификации моделей предметной области с использованием системы MetaLanguage**

Рассмотрев процесс создания моделей предметной области с использованием системы MetaLanguage, перейдем к описанию алгоритмов, реализующих основные этапы этого процесса.

Для формального описания процесса создания визуальных предметно-ориентированных языков была построена математическая модель, в которой даны формальные определения метамодели и модели, описаны операции создания графа модели и его интерпретации [4, 5].

## Построение графа метамодели

*Создание метамодели* – это построение такого графа метамодели  $GMM = (V, E)$ , для которого в начальный момент времени  $V = \emptyset, E = \emptyset$ , а в процессе добавления сущностей языка моделирования, отношений между ними и ограничений, налагаемых на сущности и отношения, множества  $V$  и  $E$  пополняются новыми элементами в соответствии с правилами создания сущностей, отношений и ограничений. Проследим, как меняется граф  $GMM$  при создании различных конструкций метаязыка системы MetaLanguage.

### Создание сущности

Определим операцию создания сущности. Пусть в граф метамодели  $GMM' = (V', E')$  необходимо добавить сущность, которой соответствует вершина  $ent'$ . Введём следующие обозначения:

- $EAttr'$  – множество вершин графа метамодели, соответствующих атрибутам создаваемой сущности;
- $ERest'$  – множество вершин графа метамодели, соответствующих ограничениям, налагаемым на создаваемую сущность;
- $eea'$  – дуга графа метамодели, соединяющая вершину  $ent'$  с множеством вершин  $EAttr'$ ;
- $eer'$  – дуга графа метамодели, соединяющая вершину  $ent'$  с множеством вершин  $ERest'$ .

*Создание сущности* – это операция над графом метамодели  $GMM'$ , результатом выполнения которой является такой граф метамодели  $GMM = (V, E)$ , что

$$V = V' \cup \{ent'\} \cup EAttr' \cup ERest', \quad E = E' \cup \{eea'\} \cup \{eer'\}.$$

В соответствии с алгоритмом 1 при создании сущности в множество вершин графа метамодели необходимо включить вершину  $ent'$ , определяющую создаваемую сущность, множество вершин, соответствующих атрибутам, ограничениям создаваемой сущности. Помимо этих вершин, необходимо создать дуги, соединяющие вершину  $ent'$  с каждым из множеств  $EAttr', ERest'$ .

При создании сущности в граф метамодели будут добавлены  $|EAttr'| + |ERest'| + 1$  вершина и две дуги, т.е. сложность алгоритма добавления новой сущности в метамодель линейна и равна  $O(|EAttr'| + |ERest'|)$ .

---

**Алгоритм 1. Добавление новой сущности в метамодель**

---

```
Добавить_новую_вершину(  $ent', V'$  );  
foreach (  $eattr'_i \in EAttr'$  )  
    Добавить_новую_вершину(  $eattr'_i, V'$  );  
Добавить_новую_дугу(  $eea', E'$  );  
foreach (  $erest'_i \in ERest'$  )  
    Добавить_новую_вершину(  $erest'_i, V'$  );  
Добавить_новую_дугу(  $eer', E'$  );
```

---

### **Создание отношения**

После описания сущностей языка моделирования, необходимо задать отношения между ними. Поскольку метамодель может содержать отношения трех типов (наследование, ассоциация и агрегация), рассмотрим создание каждого из типов отношений отдельно.

**Создание отношения наследования.** Смысл отношения данного типа заключается в наследовании атрибутов, ограничений, отношений сущности-родителя. Наследование атрибутов и ограничений реализовано посредством их копирования. Это обосновано тем, что вершины, соответствующие этим характеристикам сущности, сгруппированы, поэтому при добавлении собственных атрибутов и ограничений дочерней сущности не требуется создание нового множества вершин и дополнительной дуги. Наследование отношений, в которых участвует родительская сущность, реализуется созданием дуг графа метамодели от вершин, соответствующих дочерним сущностям, к вершинам, соответствующим наследуемым отношениям. Таким образом, при создании отношения наследования в множество атрибутов, ограничений сущности-потомка необходимо добавить вершины, соответствующие атрибутам, ограничениям сущности-родителя, а также провести дуги к вершинам, которые соответствуют отношениям, наследуемым у сущности-родителя. Поскольку при построении модели экземпляр отношения наследования создать нельзя, то собственных атрибутов и ограничений данный тип отношения не имеет.

Пусть в граф метамодели  $GMM' = (V', E')$  необходимо добавить отношение наследования, которому соответствует вершина  $rel'$ . Отношению наследования поставим в соответствие вершину графа и две дуги. Первая дуга направлена от вершины, соответствующей дочерней сущности, к вершине графа, соответствующей отношению наследования. Вторая дуга направлена от вершины, соответствующей

отношению наследования, к вершине, соответствующей родительской сущности.

Введём следующие обозначения:

- $ent_i$  – вершина графа метамодели, соответствующая родительской сущности;
- $ent_{i-1}$  – вершина графа метамодели, соответствующая дочерней сущности;
- $EAttr'_i$  – множество вершин графа метамодели, соответствующих атрибутам родительской сущности;
- $ERest'_i$  – множество вершин графа метамодели, соответствующих ограничениям, налагаемым на родительскую сущность;
- $EAttr'_{i-1}$  – множество вершин графа метамодели, соответствующих атрибутам дочерней сущности;
- $ERest'_{i-1}$  – множество вершин графа метамодели, соответствующих ограничениям дочерней сущности;
- $eea'_{i-1}$  – дуга метамодели, соединяющая вершину  $ent_{i-1}$  с множеством вершин, соответствующих атрибутам дочерней сущности;
- $eer'_{i-1}$  – дуга графа метамодели, соединяющая вершину  $ent_{i-1}$  с множеством вершин, соответствующих ограничениям, налагаемым на дочернюю сущность;
- $Rel'_i$  – множество дуг графа метамодели, соединяющих вершину  $ent_{i-1}$  с вершинами графа метамодели, соответствующими отношениям, в которых участвует родительская сущность.

Создание отношения наследования – это операция над графом метамодели  $GMM' = (V', E')$ , результатом выполнения которой является такой граф метамодели  $GMM = (V, E)$ , что

$$V = V' \cup \{rel'\} \cup RAttr'_i \cup RRest'_i,$$

$$E = E' \cup \{eea'_{i-1}\} \cup \{eer'_{i-1}\} \cup Rel'_i \cup \{(ent_{i-1}, rel')\} \cup \{(rel', ent_i)\}.$$

При создании отношения наследования в метамодель будут добавлены  $|RAttr'_i| + |RRest'_i| + 1$  вершина и  $|Rel'_i| + 4$  дуги. Сложность алгоритма создания отношения наследования (см. алгоритм 2) равна  $O(|RAttr'_i| + |RRest'_i| + |Rel'_i|)$ .

---

**Алгоритм 2. Создание отношения наследования**

---

```
Добавить_новую_вершину (  $rel', V'$  ) ;  
foreach (  $eattr'_{i_j} \in EAttr'_i$  )  
    Добавить_новую_вершину (  $eattr'_{i_j}, EAttr'_{i-1}$  ) ;  
foreach (  $erest'_{i_j} \in ERest'_i$  )  
    Добавить_новую_вершину (  $erest'_{i_j}, ERest'_{i-1}$  ) ;  
if (  $eea'_{i-1} \notin E'$  )  
    Добавить_новую_дугу (  $eea'_{i-1}, E'$  ) ;  
if (  $eer'_{i-1} \notin E'$  )  
    Добавить_новую_дугу (  $eer'_{i-1}, E'$  ) ;  
foreach (  $rel'_{i_j} \in Rel'_i$  )  
    Добавить_новую_дугу (  $rel'_{i_j}, E'$  ) ;  
Добавить_новую_дугу (  $(ent_{i-1}, rel'), E'$  ) ;  
Добавить_новую_дугу (  $(rel', ent_i), E'$  ) ;
```

---

Рассмотрим пример. Опишем фрагмент графа  $GMM$ , соответствующий метамодели ERD-диаграмм, изображенной на рис. 2.

Метамодель содержит абстрактную сущность «Абстрактный класс», которая имеет атрибуты «Имя», «Описание». Данная сущность выступает в качестве родительской для сущности «Сущность», которая не имеет собственных атрибутов (см. рис. 3).

Теперь в метамодели создадим отношение наследования между сущностями «Абстрактный класс» и «Сущность» (см. рис. 4).

Как видно из рис. 4, помимо дуг пути «Сущность» – «Наследование» – «Абстрактный класс» (все дуги пути имеют метку « $inh$ »), в графе метамодели было произведено копирование наследуемых атрибутов  $EAttr_{i-1} = \{\langle \text{Имя} \rangle, \langle \text{Описание} \rangle\}$  и проведена дуга от сущности «Сущность» к множеству этих атрибутов.

**Создание отношения ассоциации.** Пусть в граф метамодели  $GMM' = (V', E')$  необходимо добавить отношение ассоциации, которому соответствует вершина  $rel'$ . Пусть  $ent_{i-1}$ ,  $ent_{i+1}$  – вершины, соответствующие соединяемым сущностям.

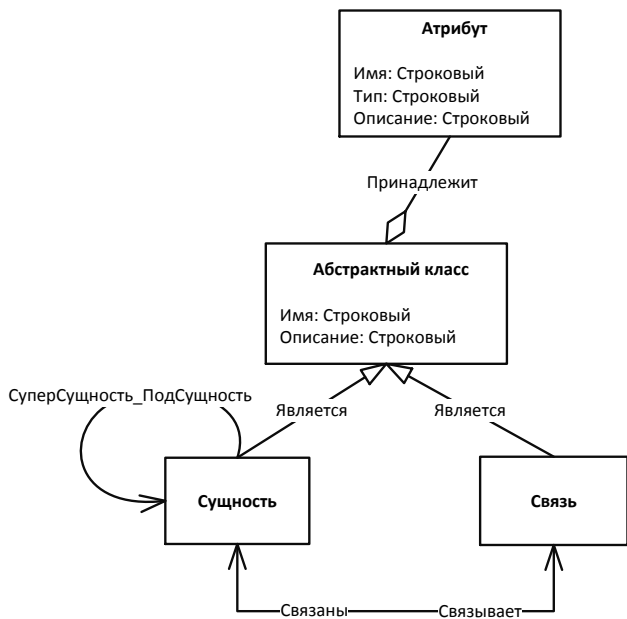


Рис. 2. Фрагмент метамодели диаграмм «Сущность-Связь»

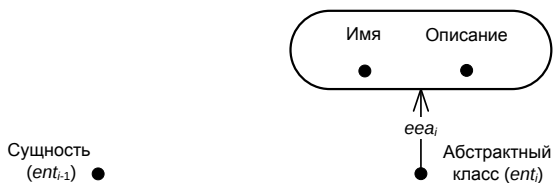


Рис. 3. Создание отношения наследования в графе метамодели.  
Этап I

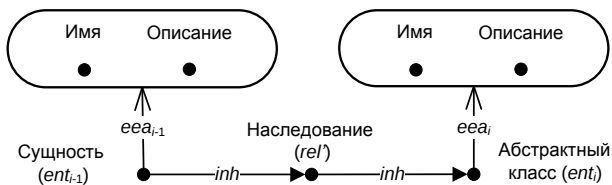


Рис. 4. Создание отношения наследования в графе метамодели.  
Этап II

Поскольку отношение «Ассоциация» может являться как однонаправленным, так и двунаправленным, то при создании этого отношения в граф метамодели для однонаправленной ассоциации будут добавлены дуги лишь прямого пути  $(ent_{i+1}, rel', ent_{i-1})$ , а для двунаправленной ассоциации, помимо дуг прямого пути  $(ent_{i+1}, rel', ent_{i-1})$ , будут добавлены дуги обратного пути  $(ent_{i-1}, rel', ent_{i+1})$ .

Кроме того, в граф метамодели необходимо добавить атрибуты и ограничения, налагаемые на создаваемое отношение, а также дуги, соединяющие вершину, соответствующую отношению ассоциации, с множествами вершин, соответствующих атрибутам и ограничениям.

Введём следующие обозначения:

- $RAttr'$  – множество вершин графа метамодели, соответствующих атрибутам создаваемого отношения;
- $RRest'$  – множество вершин графа метамодели, соответствующих ограничениям, налагаемым на создаваемое отношение ассоциации;
- $era'$  – дуга графа метамодели, соединяющая создаваемое отношение с множеством вершин, соответствующих его атрибутам;
- $err'$  – дуга графа метамодели, соединяющая создаваемое отношение с множеством вершин, соответствующих ограничениям, налагаемым на него.

*Создание отношения ассоциации* – это операция над графом метамодели  $GMM'$ , результатом выполнения которой является такой граф метамодели  $GMM = (V, E)$ , что  $V = V' \cup \{rel'\} \cup RAttr' \cup RRest'$ , а множество дуг определяется следующим образом:

- $E = E' \cup \{era'\} \cup \{err'\} \cup \{(ent_{i+1}, rel')\} \cup \{(rel', ent_{i-1})\}$ , если ассоциация однонаправлена;
- $E = E' \cup \{era'\} \cup \{err'\} \cup \{(ent_{i+1}, rel')\} \cup \{(rel', ent_{i-1})\} \cup \{(ent_{i-1}, rel')\} \cup \{(rel', ent_{i+1})\}$ , если ассоциация двунаправлена.

При создании отношения ассоциации в метамодель будут добавлены  $|RAttr'| + |RRest'| + 1$  вершина, четыре дуги в случае однонаправленной ассоциации и шесть дуг в случае двунаправленной (см. алгоритм 3). Сложность алгоритма создания отношения ассоциации равна  $O(|RAttr'| + |RRest'|)$ .

---

**Алгоритм 3. Создание двунаправленного  
отношения ассоциации**

---

```

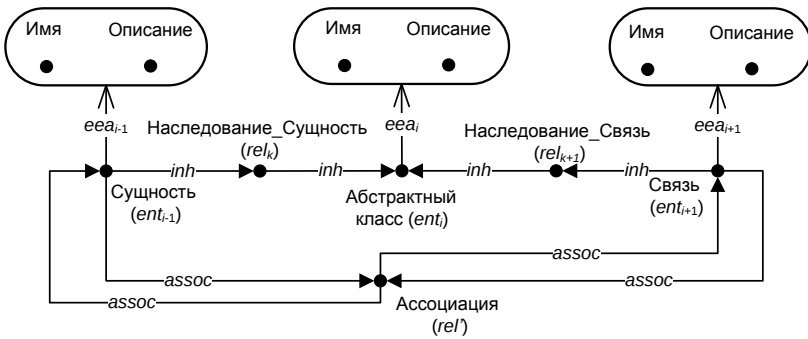
Добавить_новую_вершину(  $rel', V'$  );
foreach (  $rattr'_i \in RAttr'$  )
    Добавить_новую_вершину(  $rattr'_i, V'$  );
Добавить_новую_дугу(  $era', E'$  );
foreach (  $rrest'_i \in RRest'$  )
    Добавить_новую_вершину(  $rrest'_i, V'$  );

Добавить_новую_дугу(  $err', E'$  );
Добавить_новую_дугу(  $(ent_{i+1}, rel'), E'$  );
Добавить_новую_дугу(  $(rel', ent_{i-1}), E'$  );
Добавить_новую_дугу(  $(ent_{i-1}, rel'), E'$  );
Добавить_новую_дугу(  $(rel', ent_{i+1}), E'$  );

```

---

Расширим граф метамодели диаграмм «Сущность-Связь» сущностью «Связь», являющейся дочерней для сущности «Абстрактный класс». Соединим в графе метамодели сущности «Сущность» и «Связь» двунаправленной ассоциацией (см. рис. 5).



**Рис. 5. Создание отношения ассоциации в графе метамодели**

Из рисунка видно, что в графе метамодели  $GMM$  отношению «Ассоциация» соответствует четыре дуги:  $(ent_{i-1}, rel')$ ,  $(rel', ent_{i+1})$ ,  $(ent_{i+1}, rel')$ ,  $(rel', ent_{i-1})$ , все они имеют метку «assoc».

**Создание отношения агрегации.** Определим операцию создания отношения агрегации. Пусть в граф метамодели  $GMM' = (V', E')$  необходимо добавить отношение агрегации, которому соответствует вершина  $rel'$ .

*Агрегация*, в отличие от ассоциации, моделирует отношение «часть-целое» и изображается при построении метамодели в виде односторонней дуги, поэтому при ее создании в граф метамодели необходимо добавить лишь дуги прямого пути, проведенного от вершины, соответствующей сущности «часть», к вершине, соответствующей данному отношению, а далее – к вершине, соответствующей сущности «целое».

Введём следующие обозначения:

- $ent_i$  – вершина графа метамодели, соответствующая сущности «целое»;
- $ent_{i+2}$  – вершин графа метамодели, соответствующая сущности «часть»;
- $RAttr'$  – множество вершин графа метамодели, соответствующих атрибутам создаваемого отношения агрегации;
- $RRest'$  – множество вершин графа метамодели, соответствующих ограничениям, налагаемым на создаваемое отношение;
- $era'$  – дуга графа метамодели, соединяющая создаваемое отношение с множеством вершин, соответствующих его атрибутам;
- $err'$  – дуга графа метамодели, соединяющая создаваемое отношение с множеством вершин, соответствующих ограничениям, налагаемым на него.

*Создание отношения агрегации* – это операция над графом метамодели  $GMM'$ , результатом выполнения которой является такой граф метамодели  $GMM = (V, E)$ , что

$$V = V' \cup \{rel'\} \cup RAttr' \cup RRest',$$

$$E = E' \cup \{era'\} \cup \{err'\} \cup \{(ent_{i+2}, rel')\} \cup \{(rel', ent_i)\}.$$

В соответствии с алгоритмом 4, при создании отношения агрегации в метамодель будут добавлены  $|RAttr'| + |RRest'| + 1$  вершина и четыре дуги. Сложность алгоритма добавления в метамодель отношения агрегации равна  $O(|RAttr'| + |RRest'|)$ .

---

**Алгоритм 4. Создание отношения агрегации**

---

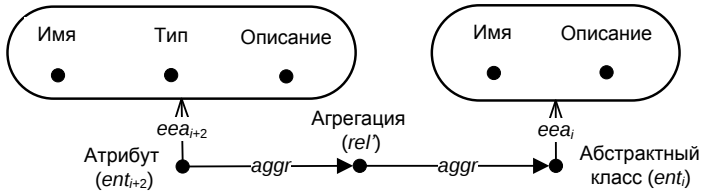
```

Добавить_новую_вершину(  $rel', V'$  );
foreach (  $rattr'_i \in RAttr'$  )
    Добавить_новую_вершину(  $rattr'_i, V'$  );
Добавить_новую_дугу(  $era', E'$  );
foreach (  $rrest'_i \in RRest'$  )
    Добавить_новую_вершину(  $rrest'_i, V'$  );
Добавить_новую_дугу(  $err', E'$  );
Добавить_новую_дугу(  $(ent_{i+2}, rel'), E'$  );
Добавить_новую_дугу(  $(rel', ent_i), E'$  );

```

---

Рассмотрим пример. Пусть в метамодель ERD-диаграмм необходимо добавить сущность «Атрибут» с атрибутами «Имя», «Тип», «Описание». Причем сущность «Атрибут» связана агрегацией с сущностью «Абстрактный класс» (см. рис. 6).



**Рис. 6. Создание отношения агрегации в графе метамодели**

Как видно из рис. 6, при создании агрегации между сущностями «Атрибут» и «Абстрактный класс» в граф метамодели были добавлены дуги пути «Атрибут» – «Агрегация» – «Абстрактный класс» (на рисунке все дуги пути имеют метку «aggr»).

**Создание ограничений**

Пусть в графе метамодели  $GMM' = (V', E')$  необходимо задать ограничение  $rest'$ , налагаемое на сущность (отношение)  $ent'$ . Обозначим дугу метамодели, соединяющую вершину  $ent'$  с множеством вершин, соответствующих ограничениям, налагаемым на  $ent'$ , через  $eer'$ .

*Создание ограничения, налагаемого на сущность (отношение)  $ent'$*  – это операция над графом метамодели  $GMM'$ , результатом вы-

полнения которой является такой граф метамодели  $GMM = (V, E)$ , что

$$V = V' \cup \{rest'\}, E = \begin{cases} E' \cup \{eer'\}, & \text{если } eer' \notin E'; \\ E', & \text{в противном случае} \end{cases}.$$

Таким образом, при добавлении ограничения, налагаемого на сущность (отношение), в граф метамодели необходимо включить вершину  $rest'$ , соответствующую данному ограничению, а также дугу, соединяющую сущность (отношение)  $ent'$  с множеством вершин, соответствующих ограничениям, на нее (него) налагаемым, если эта дуга ранее не была создана.

### Построение графа модели

При *создании модели* необходимо определить основные понятия предметной области, которые являются экземплярами сущностей метамодели, а также связи между понятиями, в качестве которых выступают экземпляры отношений метамодели.

*Создание модели* – это построение такого графа модели  $GM = (VI, EI)$ , для которого в начальный момент времени  $VI = \emptyset, EI = \emptyset$ , а в процессе добавления новых объектов предметной области и связей между ними множества  $VI, EI$  пополняются новыми элементами в соответствии с правилами создания экземпляров сущностей и отношений.

Рассмотрим основные операции создания модели и то, как меняется граф модели при выполнении этих операций.

#### *Создание экземпляра сущности*

Определим операцию создания экземпляра сущности. Пусть в граф модели  $GM' = (VI', EI')$  необходимо добавить экземпляр сущности, которому соответствует вершина  $entI'$ . Введём следующие обозначения:

- $ent'$  – вершина графа метамодели, соответствующая сущности, экземпляр которой добавляется в модель;
- $EAttrI'$  – множество вершин графа модели, соответствующих значениям атрибутов создаваемого экземпляра сущности;
- $eeal'$  – дуга графа модели, соединяющая вершину  $entI'$  с множеством вершин  $EAttrI'$ ;
- $t'$  – дуга, соединяющая вершину графа модели  $entI'$  с верши-

ной графа метамодели  $ent'$ .

Дуга  $t'$  позволяет поддерживать метамодели и созданные на их основе модели в согласованном состоянии. При внесении изменений в метамодель системы MetaLanguage выполняет все необходимые изменения в соответствующих моделях. Кроме того, наличие данной связи позволяет выполнять интерпретацию моделей, определять типы их элементов, выполнять различные операции над этими моделями.

*Создание экземпляра сущности* – это операция над графом модели  $GM'$ , результатом выполнения которой является такой граф модели  $GM = (VI, EI)$ , что

$$VI = VI' \cup \{entI'\} \cup EAttrI', \quad EI = EI' \cup \{eeal'\} \cup \{t'\}.$$

В соответствии с алгоритмом 5, при создании экземпляра сущности в множество вершин модели необходимо включить вершину  $entI'$ , соответствующую создаваемому экземпляру сущности, а также множество вершин, соответствующих значениям его атрибутов. Помимо этого, необходимо создать дугу, соединяющую вершину  $entI'$  с множеством вершин  $EAttrI'$ , и дугу  $t'$ , соединяющую создаваемый экземпляр сущности с сущностью, на основе которой создаётся данный экземпляр.

---

**Алгоритм 5. Создание экземпляра сущности**

---

```

Добавить_новую_вершину( $entI', VI'$ ) ;
foreach ( $eattrI'_i \in EAttrI'$ )
    Добавить_новую_вершину( $eattrI'_i, VI'$ ) ;
Добавить_новую_дугу( $eeal', EI'$ ) ;
Добавить_новую_дугу( $t', EI'$ ) ;

```

---

При создании экземпляра сущности в граф модели будут добавлены  $|EAttrI'| + 1$  вершина и две дуги, т.е. сложность алгоритма добавления нового экземпляра сущности в модель линейна и равна  $O(|EAttrI'|)$ .

**Создание экземпляра отношения**

В соответствии с предложенным подходом в моделях могут быть созданы только экземпляры отношений ассоциации и агрегации. Поскольку отношение агрегации – это частный случай отношения ассоциации, а экземпляр отношения ассоциации всегда является однонаправленным, то достаточно рассмотреть алгоритм создания экземпляра

отношения только одного типа. Проследим, как изменится граф модели при создании экземпляра отношения ассоциации (агрегации).

Определим операцию создания экземпляра отношения ассоциации (агрегации). Пусть в граф модели  $GM' = (VI', EI')$  необходимо добавить экземпляр отношения ассоциации (агрегации), которому соответствует вершина  $relI'$ . Пусть  $entI'_i$ ,  $entI'_{i+1}$  – вершины, соответствующие соединяемым экземплярам сущностей, а  $t'$  – дуга, соединяющая вершину графа модели  $relI'$  с вершиной графа метамодели, соответствующей отношению, экземпляр которого создается. Обозначим через  $RAttrI'$  множество вершин графа модели, соответствующих значениям атрибутов создаваемого экземпляра отношения, а через  $eraI'$  – дугу графа модели, соединяющую вершину  $relI'$  с множеством вершин  $RAttrI'$ .

*Создание экземпляра отношения ассоциации (агрегации)* – это операция над графом модели  $GM'$ , результатом выполнения которой является такой граф модели  $GM = (VI, EI)$ , что

$$VI = VI' \cup \{relI'\} \cup RAttrI',$$

$$EI = EI' \cup \{eraI'\} \cup \{(entI'_{i+1}, relI')\} \cup \{(relI', entI'_i)\} \cup \{t'\}.$$

Таким образом, при создании экземпляра отношения ассоциации (агрегации) в граф модели необходимо добавить вершину, соответствующую создаваемому экземпляру отношения, множество вершин-значений атрибутов экземпляра отношения, дугу  $eraI'$ , дугу  $t'$ , соединяющую вершину графа модели, соответствующую создаваемому экземпляру отношения, и вершину графа метамодели, соответствующую отношению, на основе которого создается этот экземпляр, а также дуги пути  $(entI'_i, relI', entI'_{i+1})$ .

При создании экземпляра отношения ассоциации (агрегации) в граф модели будут добавлены  $|RAttrI'| + 1$  вершина и четыре дуги (см. алгоритм 6). Сложность алгоритма создания экземпляра отношения ассоциации (агрегации) равна  $O(|RAttrI'|)$ .

Рассмотрим пример. На рис. 7 изображен граф, представляющий собой объединение графа метамодели ERD-диаграмм и графа модели, созданной с помощью этой метамодели. Построение такого графа возможно в силу того, что для описания графов моделей различных уровней иерархии используется единый формализм – ориентированные псевдо-метаграфы. Для наглядности на рисунке не изображены вершины, соответствующие атрибутам сущностей.

---

**Алгоритм 6. Создание экземпляра отношения  
ассоциации (агрегации)**

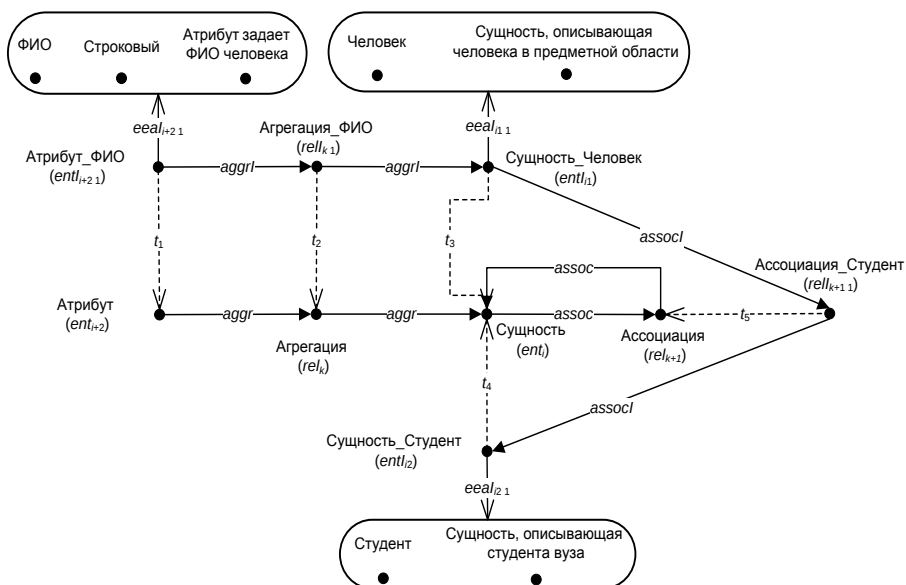
---

```

Добавить_новую_вершину(  $relI', VI'$  );
foreach (  $attrI'_i \in RAttrI'$  )
    Добавить_новую_вершину(  $attrI'_i, VI'$  );
Добавить_новую_дугу(  $eraI', EI'$  );
Добавить_новую_дугу( (  $entI'_{i+1}, relI'$  ),  $EI'$  );
Добавить_новую_дугу( (  $relI', entI'_i$  ),  $EI'$  );
Добавить_новую_дугу(  $t', EI'$  );

```

---



**Рис. 7. Граф модели, построенный на основе графа  
метамодели ERD-диаграмм**

На рис. 7 представлены следующие вершины, соответствующие экземплярам сущностей:

- $entI_{i,1}$  – вершина графа модели «Сущность\_Человек», соответствующая экземпляру сущности «Сущность»;
- $entI_{i,2}$  – вершина графа модели «Сущность\_Студент», соответ-

ствующая экземпляру сущности «Сущность», которая является подсущностью для суперсущности «Сущность\_Человек»;

- $entI_{i+2_1}$  – вершина графа модели «Атрибут\_ФИО», соответствующая экземпляру сущности «Атрибут».

Каждая из этих вершин соединена дугой с вершинами, соответствующими значениям ее атрибутов.

Модель содержит экземпляры двух отношений, в графе  $GM$  им соответствуют вершины:

- $relI_{k_1}$  – экземпляр отношения «Агрегация»;
- $relI_{k+1_1}$  – экземпляр отношения «Ассоциация».

Путь  $(entI_{i_1}, relI_{k+1_1}, entI_{i_2})$  определяет экземпляр отношения ассоциации. Вершина  $entI_{i+2_1}$  соединена дугой с вершиной  $relI_{k_1}$ , которая, в свою очередь, смежна вершине  $entI_{i_1}$ , что отвечает правилам создания экземпляра отношения агрегации в модели.

Дуги  $t_1, t_2, t_3, t_4, t_5$  соединяют экземпляры сущностей и отношений с сущностями и отношениями, на основе которых они созданы. Эти дуги представлены на рисунке пунктирными линиями.

### **Удаление элементов модели и метамодели**

Еще одной операцией над моделью и метамоделью является удаление их элементов, причем при исключении сущностей и отношений необходимо каскадно удалить все их экземпляры.

#### **Удаление элементов модели**

Поскольку алгоритмы удаления элементов модели используются при выполнении алгоритмов удаления элементов метамодели, то рассмотрим их в первую очередь. Проследим, что происходит с графом модели при удалении экземпляров сущностей и отношений.

**Удаление экземпляра отношения.** Определим операцию удаления экземпляра отношения. Пусть из графа модели  $GM' = (VT', EI')$  необходимо удалить экземпляр отношения, которому соответствует вершина  $relI'$ . Для этого необходимо удалить:

- вершину графа модели  $relI'$ , соответствующую удаляемому экземпляру отношения;
- множество вершин графа модели, соответствующих атрибутам удаляемого экземпляра отношения, –  $RAttrI'$ ;

- дугу графа модели, соединяющую вершину  $relI'$  с множеством вершин  $RAttrI'$ , –  $eraI'$ ;
- дугу  $t'$ , соединяющую удаляемый экземпляр отношения с отношением, на основе которого он был создан;
- множество дуг графа модели, инцидентных вершине  $relI'$ , –  $ERI'$ .

Удаление экземпляра отношения – это операция над графом модели  $GM' = (VI', EI')$ , результатом выполнения которой является такой граф модели  $GM = (VI, EI)$ , что

$$VI = VI' \setminus (\{relI'\} \cup RAttrI'), \quad EI = EI' \setminus (\{eraI'\} \cup ERI' \cup \{t'\}).$$

В соответствии с алгоритмом 7, при выполнении операции удаления экземпляра отношения из графа модели будут удалены  $|RAttrI'| + 1$  вершина и  $|ERI'| + 2$  дуги, т.е. сложность алгоритма линейна и равна  $O(|RAttrI'| + |ERI'|)$ .

---

**Алгоритм 7. Удаление экземпляра отношения**

---

```

Удалить_дугу( $eraI', EI'$ );
foreach ( $rattrib_i \in RAttrI'$ )
    Удалить_вершину( $rattrib_i, VI'$ );
foreach ( $erI_i \in ERI'$ )
    Удалить_дугу( $erI_i, EI'$ );
Удалить_дугу( $t', EI'$ );
Удалить_вершину( $relI', VI'$ );

```

---

**Удаление экземпляра сущности.** Определим операцию удаления экземпляра сущности. Пусть из графа модели  $GM' = (VI', EI')$  необходимо удалить экземпляр сущности, которому соответствует вершина  $entI'$ . Для этого необходимо удалить:

- вершину графа модели  $entI'$ , соответствующую удаляемому экземпляру сущности;
- множество вершин графа модели, соответствующих значениям атрибутов удаляемого экземпляра сущности, –  $EAttrI'$ ;
- множество экземпляров отношений, в которых участвует удаляемый экземпляр сущности, –  $RelI'$ , удаление экземпляров отношений осуществляется с помощью алгоритма 7;
- дугу графа модели, соединяющую вершину  $entI'$  с множе-

- дугу  $t'$ , соединяющую удаляемый экземпляр сущности с сущностью, на основе которой он был создан.

Удаление экземпляра сущности – это операция над графом модели  $GM' = (VI', EI')$ , результатом выполнения которой является такой граф модели  $GM = (VI, EI)$ , для которого после удаления множества экземпляров отношений  $RelI'$  выполняется

$$VI = VI' \setminus (\{entI'\} \cup EAttrI'), \quad EI = EI' \setminus (\{eeal'\} \cup \{t'\}).$$

В соответствии с алгоритмом 8, при удалении экземпляра сущности из графа модели будут удалены  $|EAttrI'| + 1$  вершина, две дуги, а также  $|RelI'|$  экземпляров отношений, т.е. сложность алгоритма удаления экземпляра сущности равна  $O(|EAttrI'| + N)$ , где  $N$  – сложность удаления  $|RelI'|$  экземпляров отношений.

---

**Алгоритм 8. Удаление экземпляра сущности**

---

```

Удалить_дугу ( eeal', EI' );
foreach ( eattrI'_i ∈ EAttrI' )
    Удалить_вершину ( eattrI'_i, VI' );
foreach ( relI'_i ∈ RelI' )
    Удалить_экземпляр_отношения ( relI'_i );
Удалить_дугу ( t', EI' );
Удалить_вершину ( entI', VI' );

```

---

Рассмотрим пример. Удалим из графа, представленного на рис. 7, экземпляр «Сущность\_Студент» сущности «Сущность».

Проследим, как изменялся граф модели при удалении экземпляра «Сущность\_Студент». На первом этапе из графа модели был удален экземпляр отношения, в котором участвует удаляемый экземпляр сущности, – «Ассоциация\_Студент» (см. рис. 8). При его удалении из графа модели были исключены:

- дуги пути  $(entI_{i_1}, relI_{k+1_1}, entI_{i_2})$ ;
- дуга  $t_5$ ;
- вершина  $relI_{k+1_1}$ .

На втором этапе из графа модели были удалены:

- дуга  $eeal_{i_2_1}$ , соединяющая вершину  $entI_{i_2}$  с множеством атри-

- бутов удаляемого экземпляра сущности;
- множество вершин  $EAttrI_{i_2}$ , соответствующих значениям атрибутов экземпляра «Сущность\_Студент»;
- дуга  $t_4$ ;
- вершина  $entI_{i_2}$ , соответствующая удаляемому экземпляру сущности.

Результатом выполнения операции является граф, представленный на рис. 9.

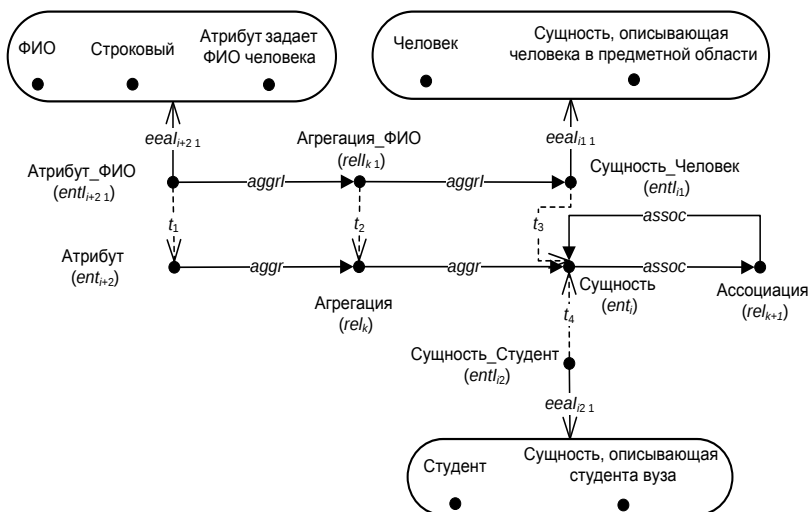


Рис. 8. Удаление экземпляра сущности «Сущность\_Студент». Этап I

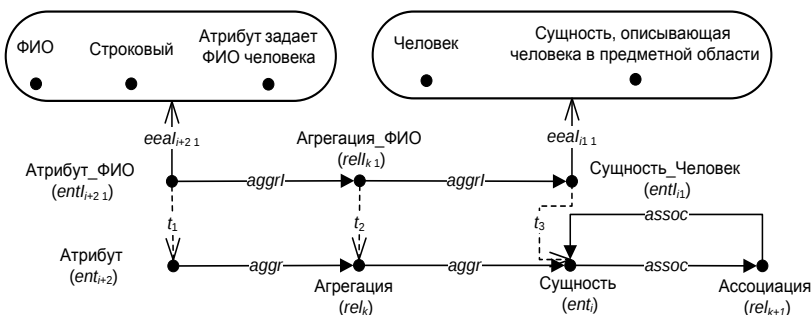


Рис. 9. Удаление экземпляра сущности «Сущность\_Студент». Этап II

### **Удаление элементов метамодели**

При рассмотрении алгоритмов удаления элементов метамодели (сущностей, отношений, ограничений) воспользуемся описанными ранее алгоритмами удаления элементов модели. Метамодель может содержать три типа отношений: наследования, ассоциации, агрегации. Поскольку агрегация – это частный случай отношения ассоциации, то семантики этих отношений схожи, отношение наследования имеет семантику, отличную от других типов отношений. Рассмотрим операцию удаления отношений метамодели отдельно для отношений ассоциации (агрегации) и отношений наследования.

**Удаление отношений «Ассоциация» и «Агрегация».** Определим операцию удаления отношения ассоциации (агрегации). Пусть из графа метамодели  $GM' = (VI', EI')$  необходимо удалить отношение ассоциации (агрегации), которому соответствует вершина  $rel'$ . Обозначим через  $k$  количество моделей, содержащих экземпляры удаляемого отношения.

Для выполнения операции удаления отношения «Ассоциация» («Агрегация») из графа метамодели и графов моделей необходимо удалить:

- вершину графа метамодели, соответствующую удаляемому отношению;
- множество вершин графа метамодели, соответствующих атрибутам удаляемого отношения, –  $RAttr'$ ;
- множество вершин графа метамодели, соответствующих ограничениям, налагаемым на удаляемое отношение, –  $RRest'$ ;
- дугу графа метамодели, соединяющую вершину  $rel'$  с множеством вершин  $RAttr'$ , –  $era'$ ;
- дугу графа метамодели, соединяющую вершину  $rel'$  с множеством вершин  $RRest'$ , –  $err'$ ;
- множество экземпляров удаляемого отношения из всех моделей, созданных на основе изменяемой метамодели, –  $\{Rel'_i\}, i = \overline{1, k}$ ; удаление экземпляров отношения осуществляется с помощью алгоритма 7;
- множество дуг графа метамодели, инцидентных вершине  $rel'$ , –  $ER'$ .

Удаление отношения ассоциации (агрегации) – это операция над графом метамодели  $GM' = (VI', EI')$  и множеством графов моделей  $GM'_i = (VI'_i, EI'_i), i = \overline{1, k}$ , результатом выполнения которой является

такой граф метамодели  $GMM = (V, E)$ , что

$$V = V' \setminus (\{rel'\} \cup RAttr' \cup RRest'), E = E' \setminus (\{era'\} \cup \{err'\} \cup ER'),$$

и графы моделей  $GM'_i = (VI'_i, EI'_i), i = \overline{1, k}$ , из которых удалены все элементы множеств  $\{RelI'_i\}, i = \overline{1, k}$ , т.е. все экземпляры удаляемого отношения.

В соответствии с алгоритмом 9, при удалении отношения ассоциации (агрегации) из графа метамодели будут удалены  $|RAttr'| + |RRest'| + 1$  вершина и  $|ER'| + 2$  дуги, а из графа модели –  $\sum_{i=1}^k |RelI'_i|$  экземпляров отношения, т.е. сложность данного алгоритма равна  $O(|RAttr'| + |RRest'| + |ER'| + \sum_{i=1}^k |RelI'_i|)$ .

---

**Алгоритм 9. Удаление отношения  
ассоциации (агрегации) из метамодели**

---

```

foreach ( $RelI'_i \subseteq \{RelI'_i\}, i = \overline{1, k}$ )
  foreach ( $relI'_{ij} \in RelI'_i$ )
    Удалить_экземпляр_отношения( $relI'_{ij}, GM'_i$ );
  Удалить_дугу( $era', E'$ );

  foreach ( $ratr'_i \in RAttr'$ )
    Удалить_вершину( $ratr'_i, V'$ );
  Удалить_дугу( $err', E'$ );

  foreach ( $rrest'_i \in RRest'$ )
    Удалить_вершину( $rrest'_i, V'$ );

  foreach ( $er'_i \in ER'$ )
    Удалить_дугу( $er'_i, E'$ );
  Удалить_вершину( $rel', V'$ );

```

---

**Удаление отношения «Наследование».** Определим операцию удаления отношения наследования. Пусть из графа метамодели  $GM' = (V', EI')$  необходимо удалить отношение наследования, которому соответствует вершина  $rel'$ . Пусть отношение наследования соединяет родительскую сущность, которой соответствует вершина  $ent_i$  с дочерней сущностью, которой соответствует вершина  $ent_{i+1}$ .

При удалении отношения наследования из графа метамодели и из графов моделей будут удалены:

- вершина графа метамодели, соответствующая удаляемому отношению;
- элементы множества  $EAttr'_{i+1}$ , которые являются наследованными атрибутами, –  $EAttrInh$ ;
- элементы множества  $ERest'_{i+1}$ , которые соответствуют ограничениям, наследованным от родительской сущности, –  $ERestInh$ ;
- множество вершин графов моделей, соответствующих значениям атрибутов экземпляров сущности  $ent_{i+1}$ , которые она унаследовала от родительской сущности, –  $\{EAttrI'_j\}$ ,  $j = \overline{1, k}$ ;
- дуга графа метамодели  $eea'_{i+1}$ , соединяющая вершину  $ent_{i+1}$  с множеством  $EAttr'_{i+1}$ , если оно пусто;
- дуга графа метамодели  $eer'_{i+1}$ , соединяющая вершину  $ent_{i+1}$  с множеством  $ERest'_{i+1}$ , если оно пусто;
- множество дуг графа метамодели, инцидентных вершине  $rel'$ , –  $ER'$ ;
- множество дуг графов моделей  $EEAI'$ , соединяющих экземпляры дочерней сущности с семейством множеств  $\{EAttrI'_j\}$ ,  $j = \overline{1, k}$ , если эти множества пусты.

Удаление отношения наследования – это операция над графом метамодели  $GM' = (VI', EI')$ , множеством графов моделей  $GM'_j = (VI'_j, EI'_j)$ ,  $j = \overline{1, k}$ , результатом выполнения которой является такой граф метамодели  $GMM = (V, E)$ , что

$$V = V' \setminus (\{rel'\} \cup EAttrInh \cup ERestInhI),$$

$$E = E' \setminus (\{eea'_{i+1}\} \cup \{eer'_{i+1}\} \cup ER'),$$

и графы моделей  $GM'_j = (VI'_j, EI'_j)$ ,  $j = \overline{1, k}$ , из которых удалены:

- вершины семейства множеств  $\{EAttrI'_j\}$ ,  $j = \overline{1, k}$ ;
- дуги множества  $EEAI'$ .

В соответствии с алгоритмом 10, при выполнении операции удаления отношения наследования из графа метамодели будут удалены  $|EAttrInh| + |ERestInh| + 1$  вершина и  $|ER'| + 2$  дуги, из графов моделей

–  $\sum_{j=1}^k (|EAttrI'_j|)$  вершин и  $|EEAI'|$  дуг, т.е. сложность алгоритма удаления отношения наследования равна

$$O\left(|EAttrInh| + |ERestInh| + |ER'| + \sum_{j=1}^k (|EAttrI'_j|) + |EEAI'|\right).$$

---

**Алгоритм 10. Удаление отношения наследования  
из метамодели**

---

```

foreach (  $EAttrI'_j \subseteq \{EAttrI'_{j_1}\}, j = \overline{1, k}$  )
{
    foreach (  $eattrI'_{j_1} \in EAttrI'_j$  )
        Удалить_вершину(  $eattrI'_{j_1}, VI'_j$  );
    if (  $EAttrI'_j == \emptyset$  )
        Удалить_дугу(  $eeaI'_j, EI'_j$  );
}
foreach (  $eattr'_{j_{i+1}} \in EAttrInh$  )
    Удалить_вершину(  $eattr'_{j_{i+1}}, EAttr'_{i+1}$  );
foreach (  $erest'_{j_{i+1}} \in ERestInh$  )
    Удалить_вершину(  $erest'_{j_{i+1}}, ERest'_{i+1}$  );
if (  $EAttr'_{i+1} == \emptyset$  )
    Удалить_дугу(  $eea'_{i+1}, E'$  );
if (  $ERest'_{i+1} == \emptyset$  )
    Удалить_дугу(  $eer'_{i+1}, E'$  );
foreach (  $er'_i \in ER'$  )
    Удалить_дугу(  $er'_i, E'$  );
Удалить_вершину(  $rel', V'$  );

```

---

Рассмотрим пример. Удалим из графа метамодели, представленного на рис. 5, отношение «Наследование» между сущностями «Сущность» и «Абстрактный класс» (см. рис. 10).

При выполнении операции из графа метамодели были удалены:

- дуга  $eea_{i-1}$ , соединяющая вершину, соответствующую дочерней сущности, с множеством ее атрибутов;
- множество атрибутов сущности «Сущность» –  $EAttr_{i-1}$ ;
- дуги пути  $(ent_{i-1}, rel_k, ent_i)$ ;
- вершина  $rel_k$ , соответствующая удаляемому отношению.

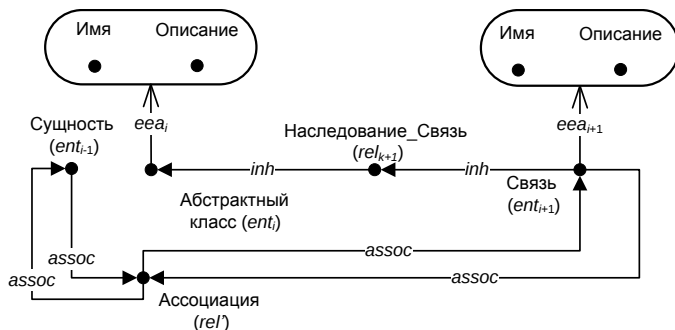


Рис. 10. Удаление отношения «Наследование»

**Удаление сущности метамодели.** Определим операцию удаления сущности. Пусть из графа метамодели  $GMM' = (V', E')$  необходимо удалить сущность, которой соответствует вершина  $ent'$ . Обозначим через  $k$  количество моделей, содержащих экземпляры удаляемой сущности. Для выполнения операции удаления сущности из графа метамодели и графов моделей необходимо удалить:

- вершину графа метамодели, соответствующую удаляемой сущности;
- множество вершин графа метамодели, соответствующих атрибутам удаляемой сущности, –  $EAttr'$ ;
- множество вершин графа метамодели, соответствующих ограничениям, налагаемым на удаляемую сущность, –  $ERest'$ ;
- множество отношений метамодели, в которых участвует удаляемая сущность, –  $Rel'$ ;
- множество экземпляров удаляемой сущности из всех моделей, созданных на основе изменяемой метамодели, –  $\{EntI_i'\}, i = \overline{1, k}$ , удаление экземпляров сущности осуществляется с помощью алгоритма 8;
- дугу графа метамодели, соединяющую вершину  $ent'$  с множеством вершин  $EAttr'$ , –  $eea'$ ;
- дугу графа метамодели, соединяющую вершину  $ent'$  с множеством вершин  $ERest'$ , –  $eer'$ .

**Удаление сущности** – это операция над графом метамодели  $GMM' = (V', E')$  и множеством графов моделей  $GM'_i = (VI'_i, EI'_i), i = \overline{1, k}$ , результатом выполнения которой является

такой граф метамодели  $GMM = (V, E)$ , для которого после удаления множества отношений  $Rel'$  выполняется:

$$V = V' \setminus (\{ent'\} \cup EAttr' \cup ERest'), \quad E = E' \setminus (\{eea'\} \cup \{eer'\}),$$

и графы моделей  $GM'_i = (VI'_i, EI'_i)$ ,  $i = \overline{1, k}$ , из которых удалены все элементы семейства множеств  $\{EntI'_i\}$ ,  $i = \overline{1, k}$ , т.е. все экземпляры удаляемой сущности.

В соответствии с алгоритмом 11 при удалении сущности из графа метамодели будут удалены  $|EAttr'| + |ERest'| + 1$  вершина, две дуги,  $|Rel'|$  отношений, а из графов моделей  $\sum_{i=1}^k |EntI'_i|$  экземпляров сущности. Таким образом, сложность алгоритма удаления сущности равна  $O(|EAttr'| + |ERest'| + M + N)$ , где  $M$  – сложность удаления  $|Rel'|$  отношений, а  $N$  – сложность удаления  $\sum_{i=1}^k |EntI'_i|$  экземпляров сущностей.

---

**Алгоритм 11. Удаление сущности из метамодели**

---

```

foreach ( $EntI'_i \subseteq \{EntI'_i\}, i = \overline{1, k}$ )
foreach ( $entI'_{ij} \in EntI'_i$ )
    Удалить_экземпляр_сущности( $entI'_{ij}, GM'_i$ );
    Удалить_дугу( $eea', E'$ );

foreach ( $eatr'_i \in EAttr'$ )
    Удалить_вершину( $eatr'_i, V'$ );
    Удалить_дугу( $eer', E'$ );

foreach ( $rel'_i \in Rel'$ )
    Удалить_отношение( $rel'_i$ );
    Удалить_вершину( $ent', V'$ );

```

---

Рассмотрим пример. Предположим, что из графа, представленного на рис. 7, необходимо удалить сущность «Сущность», которой соответствует вершина  $ent_i$ . На первом этапе из графа модели были удалены все экземпляры сущности «Сущность», в результате был получен граф, изображенный на рис. 11 (в граф добавлены скрытые ранее множества  $EAttr_i$ ,  $EAttr_{i+2}$  и дуги  $eea_i$ ,  $eea_{i+2}$ ).

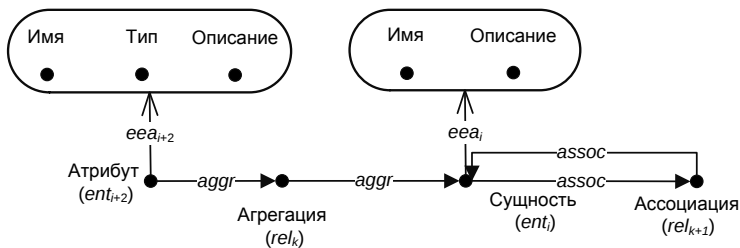


Рис. 11. Удаление сущности «Сущность». Этап I

На втором этапе из метамодели были удалены отношения, в которых участвует сущность «Сущность»: отношения «Агрегация» и «Ассоциация». При этом из графа метамодели были исключены (см. рис. 12):

- дуги путей  $(ent_{i+2}, rel_k, ent_i)$ ,  $(ent_i, rel_{k+1}, ent_i)$ ;
- вершины  $rel_k$ ,  $rel_{k+1}$ .

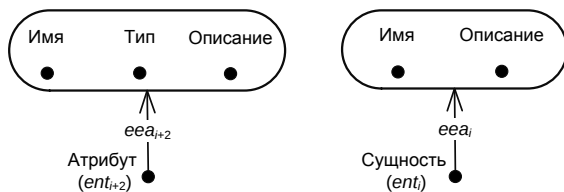


Рис. 12. Удаление сущности «Сущность». Этап II

На третьем этапе из графа метамодели были удалены:

- дуга  $ееa_i$ , соединяющую вершину  $ent_i$  с множеством вершин  $EAttr_i$ ;
- множество вершин, соответствующих атрибутам удаляемой сущности, –  $EAttr_i$ ;
- вершина  $ent_i$ , соответствующая удаляемой сущности.

Результатом выполнения операций является граф, представленный на рис. 13.

Поскольку сущность «Атрибут» связана агрегацией с сущностью «Абстрактный класс», то при удалении сущности «Сущность» она не была удалена.

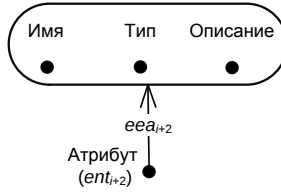


Рис. 13. Удаление сущности «Сущность». Этап III

**Удаление ограничений.** Пусть в графе метамодели  $GMM' = (V', E')$  необходимо удалить ограничение  $rest'$ , налагаемое на сущность (отношение)  $ent'$ . Введём следующие обозначения:

- $ERest'$  – множество вершин, соответствующих ограничениям, налагаемым на  $ent'$ ;
- $eer'$  – дуга, соединяющая вершину  $ent'$  с множеством  $ERest'$ .

Удаление ограничения, налагаемого на сущность (отношение)  $ent'$  – это операция над графом метамодели  $GMM' = (V', E')$ , результатом выполнения которой является такой граф метамодели  $GMM = (V, E)$ , что

$$V = V' \setminus \{rest'\}, E = \begin{cases} E' \setminus \{eer'\}, & \text{если } ERest' = \{rest'\}; \\ E', & \text{в противном случае.} \end{cases}$$

При удалении ограничения, налагаемого на сущность (отношение), из графа метамодели необходимо удалить вершину  $rest'$ , соответствующую данному ограничению, а также дугу, соединяющую вершину  $ent'$  с множеством ограничений, налагаемых на изменяемую сущность (отношение), если это множество стало пустым.

### Проверка ограничений

Проверка ограничений, налагаемых на метамодель, реализуется посредством функции над графом модели  $GM = (VI, EI)$ , которая возвращает значение  $\varepsilon$  такое, что

$$\varepsilon = \begin{cases} true, & \text{если присутствует нарушение ограничения;} \\ false, & \text{при отсутствии нарушений.} \end{cases}$$

Некоторые виды ограничений проверяются при создании сущностей и отношений, другие – непосредственно после создания модели. Рассмотрим алгоритмы проверки каждого из видов ограничений более подробно.

## **Проверка ограничений, налагаемых на сущности**

Уникальность имени экземпляра сущности проверяется непосредственно при ее создании. В случае, если задано значение «уникальна в рамках метамодели», то будут просмотрены все экземпляры сущности во всех моделях, построенных на основе данной метамодели (см. алгоритм 12). Если сущность «уникальна в рамках модели», то достаточно просмотреть все экземпляры сущности только текущей модели.

Сложность алгоритма проверки уникальности имени экземпляра сущности равна  $O\left(\sum_{k=1}^{|M'|} \sum_{i=1}^{|Ent_k|} |EntI_{k_i}|\right)$ .

---

### **Алгоритм 12. Проверка уникальности имени экземпляра сущности**

---

```

switch ( entI.EUnique )
{
    case "уникальна в рамках метамодели":
        foreach ( m'_k ∈ M' )
            foreach ( ent_{k_i} ∈ m'_k )
                foreach ( entI_{k_{i_1}} ∈ ent_{k_i} )
                    if ( entI.EName == entI_{k_{i_1}}.EName )
                    {
                        Выдать_сообщение_об_ошибке();
                        ε = true;
                    }
                    break;
    case "уникальна в рамках модели":
        foreach ( ent_{k_i} ∈ m'_k )
            foreach ( entI_{k_{i_1}} ∈ ent_{k_i} )
                if ( entI.EName == entI_{k_{i_1}}.EName )
                {
                    Выдать_сообщение_об_ошибке();
                    ε = true;
                }
                break;
}

```

---

Ограничения на количество экземпляров сущности в модели, как и ограничения на уникальность имени экземпляра, проверяются непосредственно при создании экземпляра сущности. В случае, если число

уже созданных экземпляров сущности определенного типа равно максимально допустимому числу экземпляров, то новый экземпляр создан не будет.

Проверка ограничений на значения атрибутов экземпляров сущностей происходит во время валидации модели. При этом ограничения, налагаемые на каждую сущность метамодели, будут проверены для всех ее экземпляров (см. алгоритм 13).

---

**Алгоритм 13. Проверка ограничений, налагаемых на значения атрибутов экземпляров сущностей**

---

```

foreach (  $ent_i \in Ent$  )
  foreach (  $erest_{ij} \in ERest_i$  )
    foreach (  $entI_{ik} \in EntI_i$  )
      if (  $Ограничение\_не\_выполняется(erest_{ij}, entI_{ik})$  )
      {
        Выдать_сообщение_об_ошибке();
         $\varepsilon = true$ ;
      }

```

---

Сложность алгоритма проверки ограничений, налагаемых на значения атрибутов экземпляров сущностей, является полиномиальной и равна  $O\left(\sum_{i=1}^{|Ent|} |ERest_i| \cdot |EntI_i|\right)$ .

**Проверка ограничений, налагаемых на отношения**

Ограничения на уникальность имени экземпляра отношения проверяются непосредственно при его создании. Алгоритм проверки ограничений этого вида аналогичен алгоритму проверки ограничений на уникальность имени экземпляра сущности (см. алгоритм 6) за исключением следующего: вместо обхода по экземплярам сущностей производится обход по экземплярам отношений.

Ограничения на типы соединяемых экземпляров сущностей проверяются при создании экземпляра отношения. Как только пользователь проводит связь между двумя экземплярами сущностей, система определяет их тип и проверяет, существует ли в метамодели отношение между сущностями этого типа. Если такое отношение существует, система позволит создать связь, в противном случае будет выдано сообщение об ошибке.

При проверке ограничений на кратность отношений система для

каждого из соединяемых экземпляров сущностей подсчитывает количество экземпляров сущностей данного типа в модели, уже участвующих в этом отношении, и в случае, если это число будет равно кратности создаваемого экземпляра отношения, то будет выдано сообщение об ошибке, и новый экземпляр отношения создан не будет.

Проверка ограничений, налагаемых на значения атрибутов соединяемых экземпляров сущностей, проводится на этапе валидации в соответствии с алгоритмом 14.

---

**Алгоритм 14. Проверка ограничений, налагаемых на значения атрибутов соединяемых экземпляров сущностей**

---

```

foreach ( $rel_{k_i} \in m'_k$ )
    foreach ( $relI_{k_{i,j}} \in rel_{k_i}$ )
    {
        if (Ограничение_не_выполняется( $relI_{k_{i,j}}$ ,  $rel_{k_i}$ .источник))
        {
            Выдать_сообщение_об_ошибке();
             $\varepsilon = \text{true}$ ;
        }
        if (Ограничение_не_выполняется( $relI_{k_{i,j}}$ ,  $rel_{k_i}$ .приемник))
        {
            Выдать_сообщение_об_ошибке();
             $\varepsilon = \text{true}$ ;
        }
    }

```

---

Сложность алгоритма проверки ограничений, налагаемых на значения атрибутов соединяемых экземпляров сущностей, равна

$$O\left(\sum_{i=1}^{|Rel_k|} |RelI_{k_i}|\right).$$

### Библиографический список

1. Лядова Л.Н., Серый А.П., Сухов А.О. Подходы к описанию вертикальных и горизонтальных трансформаций метамodelей // Математика программных систем: межвуз. сб. науч. ст. Пермь: Изд-во Перм. гос. нац. исслед. ун-та, 2012. Вып. 9. С. 33-49.
2. Сухов А.О. Инструментальные средства создания визуальных предметно-ориентированных языков моделирования // Фундаментальные исследования. 2013. № 4 (ч. 4). С. 848-852.

3. Сухов *А.О.* Среда разработки визуальных предметно-ориентированных языков моделирования // Математика программных систем: межвуз. сб. науч. ст. Пермь: Изд-во Перм. гос. ун-та, 2008. Вып. 5. С. 84-94.
4. Сухов *А.О.* Теоретические основы разработки DSL-инструментария с использованием графовых грамматик // Информатизация и связь. 2011. № 3. С. 35-37.
5. Сухов *А.О.* Формальное описание метаязыка системы MetaLanguage // Труды всерос. научно-практической конференции «Современные проблемы математики и ее прикладные аспекты». Пермь: Изд-во Перм. гос. ун-та, 2010. С. 154-159.
6. Sukhov *A.O.*, Lyadova *L.N.* Horizontal Transformations of Visual Models in MetaLanguage System // Proc. of the 7th Spring/Summer Young Researchers' Colloquium on Software Engineering. Moscow: ISP RAS, 2013. P. 31-40.
7. Sukhov *A.O.*, Lyadova *L.N.* MetaLanguage: a Tool for Creating Visual Domain-Specific Modeling Languages // Proc. of the 6th Spring/Summer Young Researchers' Colloquium on Software Engineering. Moscow: ISP RAS, 2012. P. 42-53.