

Генерация кода в режиме “метамоделирования на лету” в системе QReal

Студент: Тихонова Мария, гр. 444

Научный руководитель: Литвинов Ю.В.,
ст.преп. кафедры СП

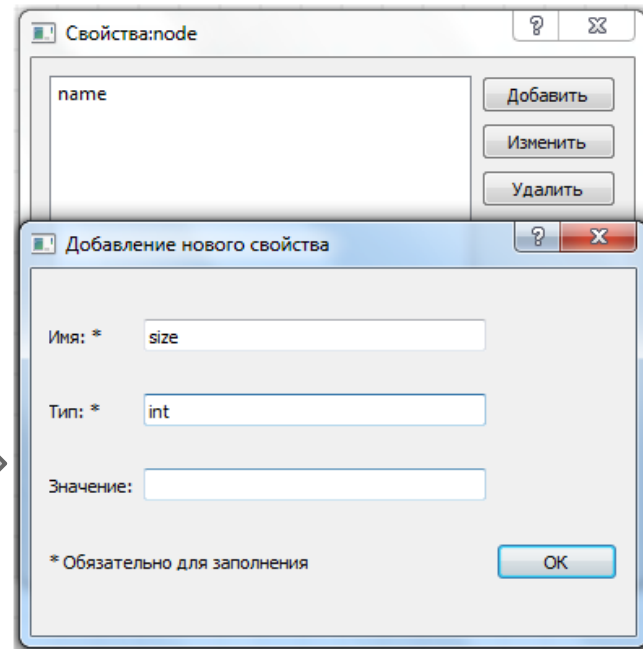
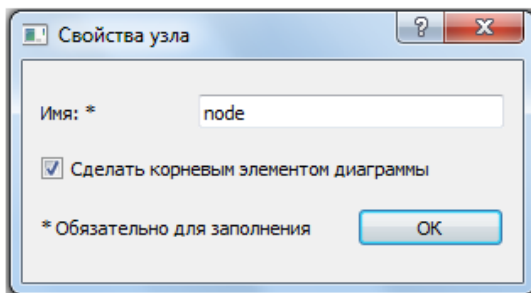
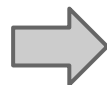
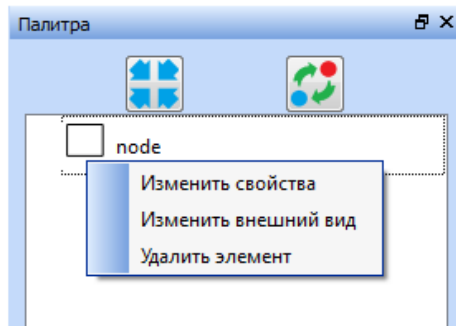
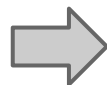
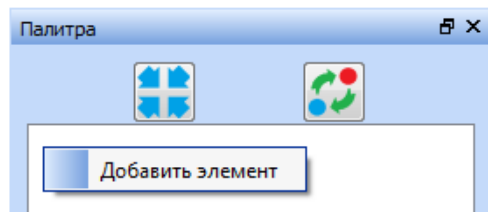
Рецензент: Подкопаев А.В.,
программист ООО “ИнтеллиДжей Лабс”

Визуальное моделирование, DSM

- визуальное моделирование
 - ускорение процесса разработки
 - диаграммы на языках общего назначения очень сложны
- предметно-ориентированные языки, DSM
- DSM-платформы (MetaEdit+, Microsoft VM SDK, QReal)
- предметно-ориентированные языки задаются с помощью метаязыка
- генерация кода – одна из важных задач DSM-платформы

“Метамоделирование на лету”

- МОЖНО ВНОСИТЬ ИЗМЕНЕНИЯ В ЯЗЫК прямо в процессе работы над ним
- уровень метамодели скрыт от пользователя



Постановка задачи

Цель:

- разработать механизм задания правил генерации в режиме “метамоделирования на лету”

Задачи:

- проанализировать средства задания правил генерации в существующих системах
- реализовать средство задания правил генерации в режиме “метамоделирования на лету”
- провести апробацию метода

Обзор существующих средств

- MetaEdit+
- Microsoft Visualization and Modeling SDK
- Eclipse (Actifsource)
- средства в QReal
 - QReal:Geny
 - визуальное средство

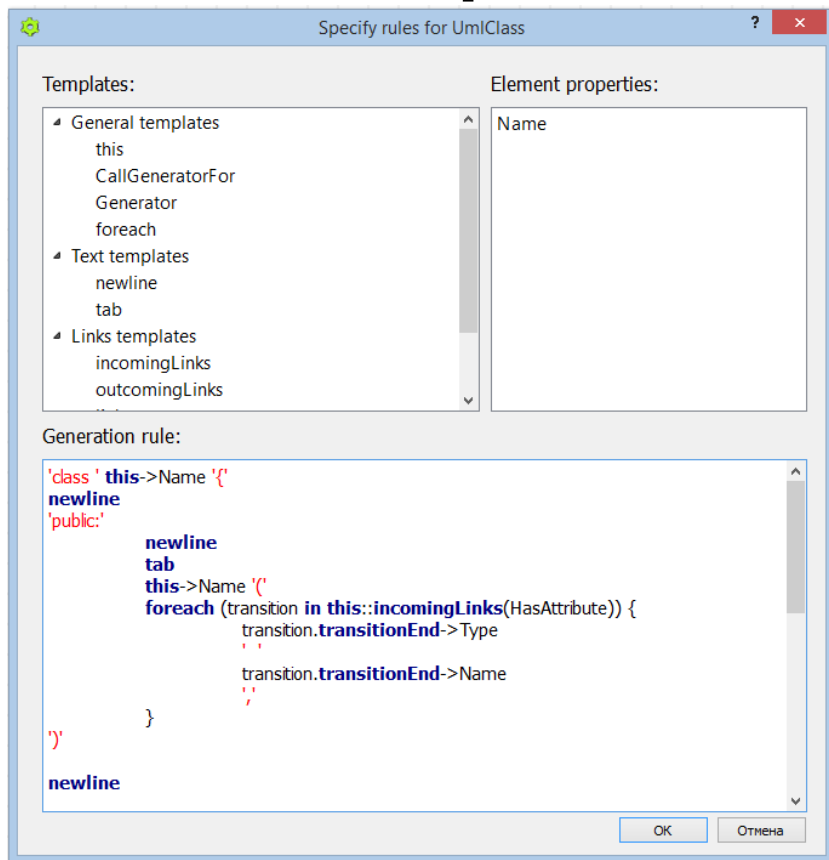
Требования

- текстовый язык
- правило задаётся для элемента языка
- средства для обхода модели
- возможность генерировать код на любом языке
- простота

Язык для задания правил генерации

- управляющие конструкции
 - foreach (element in List) { }
 - incomingLinks, outgoingLinks, links
 - transitionStart, transitionEnd
 - callGeneratorFor(element, generatorName)
 - generateToFile(fileName) { }
 - Generator generatorName { }
 - element->property
 - this
 - if () { } else { }
 - newline, tab
- текст, попадающий в итоговый код без изменений, заключается в кавычки “

Реализация

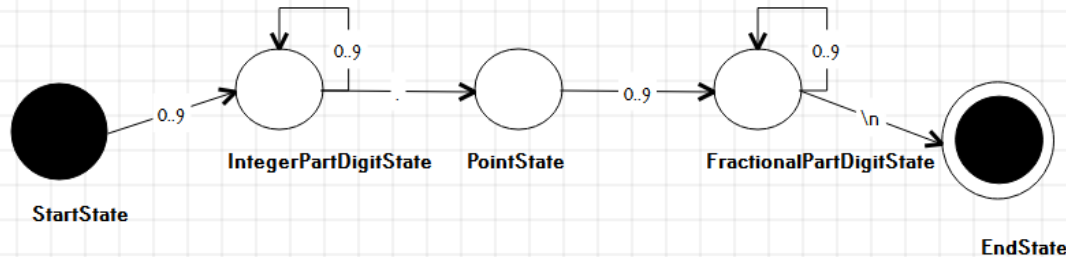


- подключаемый модуль к QReal
- правила хранятся в метамодели
- лексический, синтаксический анализаторы и генератор

Пример (генерация перечисления)

```
'enum State {  
    foreach (currentState in State) {  
        currentState->name  
        ','  
        newline  
    }  
    foreach (currentState in StartState) {  
        currentState->name  
    }  
    ','  
    newline  
    foreach (currentState in EndState) {  
        currentState->name  
    }  
    newline  
}'
```

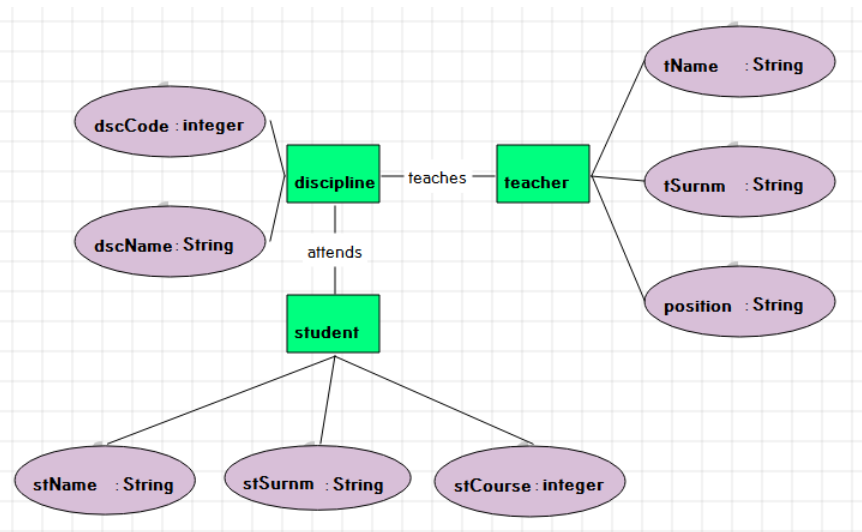
```
enum State {  
    IntegerPartDigitState  
    , PointState  
    , FractionalPartDigitState  
    , StartState  
    , EndState  
};
```



Пример (ER-диаграмма)

- правило для сущности:

```
'CREATE TABLE ' this->Name '('  
  newline  
  foreach (attributeLink in this::outcomingLinks(HasAttribute)) {  
    callGeneratorFor(attributeLink.transitionEnd)  
  }  
)'
```



- правило для свойства:

```
tab  
this->Name ''  
this->Type ''  
'NOT NULL,'  
newline
```

Сгенерированный код:

```
CREATE TABLE discipline(  
  dscName String NOT NULL,  
)  
CREATE TABLE teacher(  
  tSurnm String NOT NULL,  
  position String NOT NULL,  
)  
CREATE TABLE student(  
  stSurnm String NOT NULL,  
  stName String NOT NULL,  
  stCourse integer NOT NULL,  
)
```

Результаты

- разработан язык для задания правил генерации в режиме “метамоделирования на лету”
- спроектировано и реализовано средство для задания правил генерации в системе QReal
- проведена апробация средства для задания правил генерации
- результаты представлены на конференции "Современные технологии в теории и практике программирования"