Реализация механизма поддержки ограничений в проекте WMP

Д.А. Когутич

Научный руководитель:
к.т.н. Ю.В. Литвинов

Рецензент:
А.О. Перешеина

Введение

- Как обеспечить и описать все аспекты спецификации?
- UML-диаграммы
- Ограничения на состояние системы
- Как их лучше задавать?

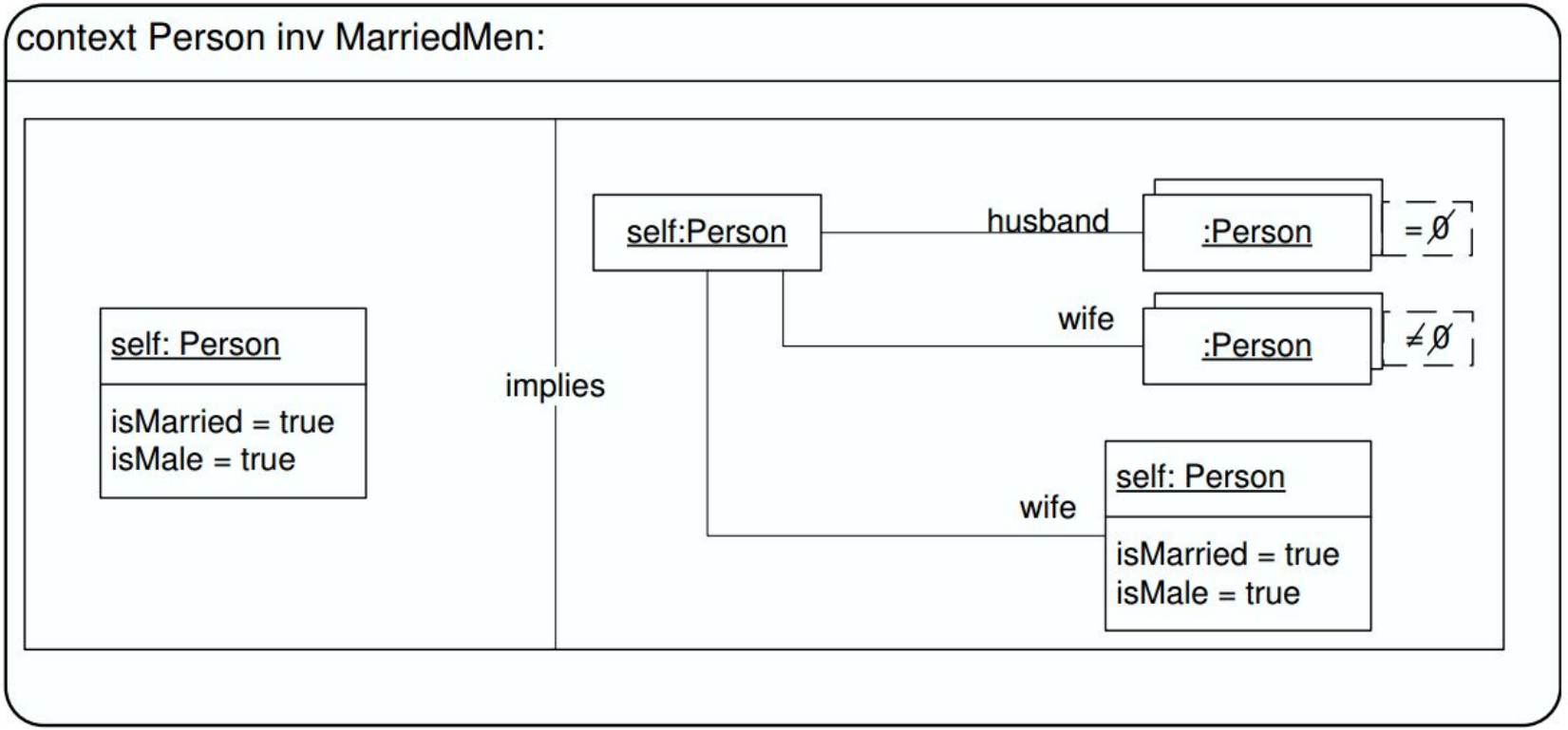
Object Constraint Language (OCL)

Примеры:


- ```
  context Company
  inv: self.numberOfEmployees > 50
  ```

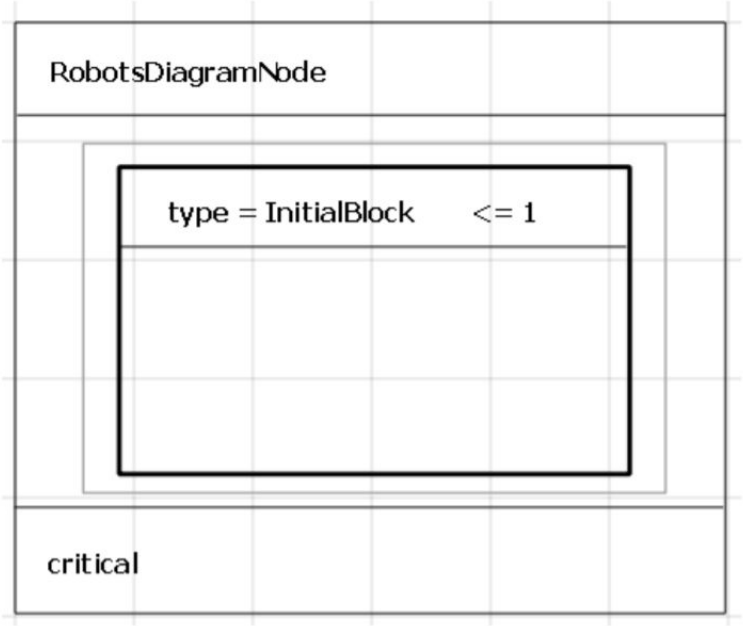
- ```
  context Person
  inv:
      let income : Integer = self.job.salary->sum() in
      if isUnemployed then
          income < 100
      else
          income >= 100
      endif
  ```

Альтернатива: VOCL



Опыт QReal

Курсовая работа студентки 345 группы Дерипаска Анны Олеговны :
Визуальный язык задания ограничений на модели в QReal , 2012



Постановка задачи

- Проанализировать существующие OCL-решения
- Реализовать синтаксический анализатор OCL-выражений
- Реализовать интерпретатор OCL-выражений
- Внедрить поддержку OCL в проект WMP

Обзор выбранного подмножества OCL

- The Essential OCL
  - Основная функциональность OCL
  - Не используется в чистом виде
- The Complete OCL
  - context, inv, def, ...

Обзор существующих реализаций OCL

- OCL.js
- Eclipse OCL
  - ANTLR-Tool

Решение проблемы с неоднозначностью грамматики

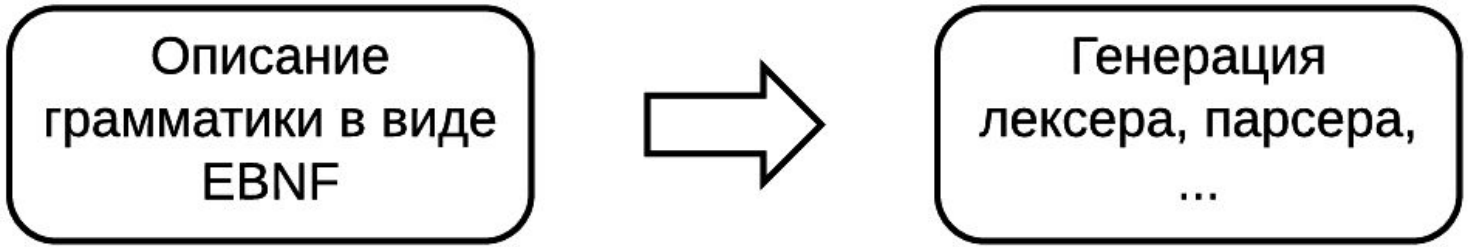
- Подход Eclipse OCL
- Расширение грамматики
- Собираем правила разрешения неоднозначностей в единое целое

Фрагмент грамматики

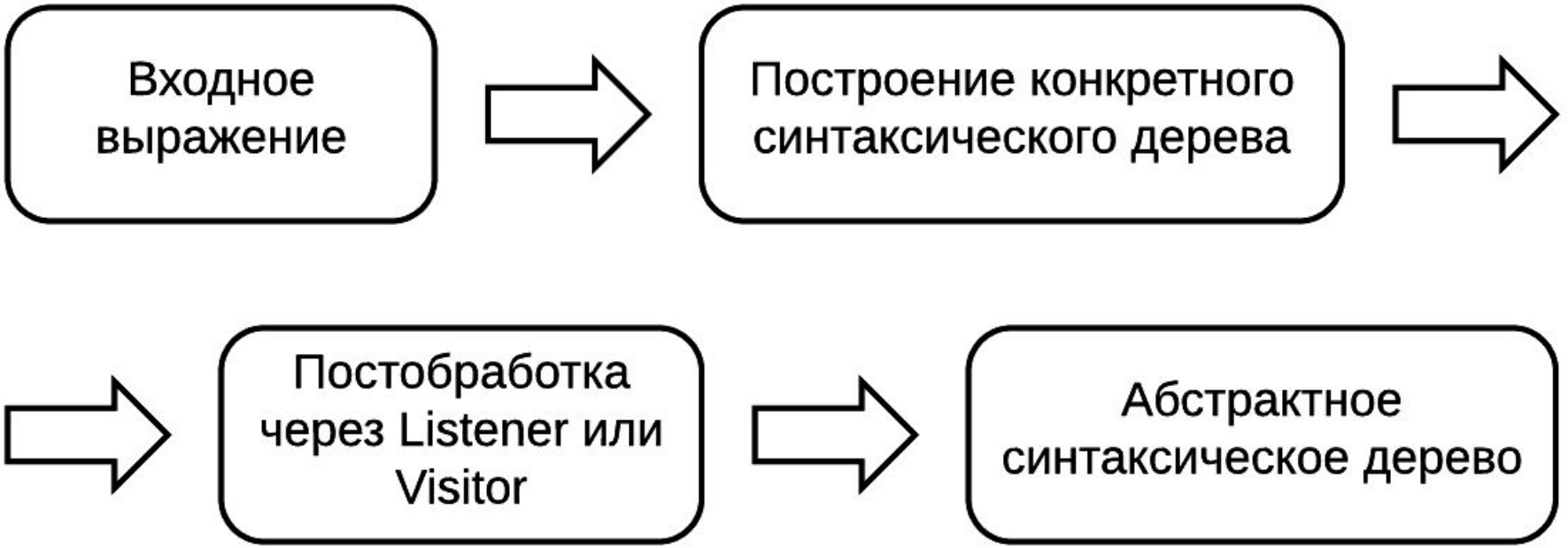
```
classifierContextDeclCS
:
      'context'
      (unrestrictedName ':')?
      nameExpCS
      (invCS | defCS)+
;

invCS
:
   'inv' unrestrictedName? ':' specificationCS
;
```

Использование ANTLR-Tool



Использование ANTLR-Tool



Интерпретация OCL

```
context A
    def: mul(a:var, b:var) = a*b
    inv: mul(self.x, self.y.z) = 10
    inv: self.elements->iterate(elem; acc:var = 1 | elem * acc) > 54
    inv: self.elements->size() * 7 = 28
    def: length = self.elements->size()
    inv: let x:var = 2 in self.elements->at(length) <> x

context B
    def: fact(n:var) = if n = 1 then n else n * fact(n - 1) endif
    inv: fact(5) = 120
    inv: Set{1,2,3,4}->size() = 4
```

Модель

```
let model = {
    <context_name1>: [...instances...],
    <context_name2>: [...instances...],
    ...
};
```

Результат интерпретации

```
{
    errors: {...},
    warnings: {...}
}
```

Поддержка OCL Standard Library

OclAny: =, <>, oclIsUndefined, oclIsInvalid, oclAsSet.

OclVoid: oclAsSet.

OclInvalid: =, <>, oclAsSet.

Операции с числами: +, –, *, /, <, <=, >, >=, abs, max, min, toString.

String: +, <, <=, >, >=, size, substring, toUpperCase, toLowerCase, indexOf.

Boolean: or, xor, and, not, implies, toString.

Collection: oclAsSet, asSet, asOrderedSet, asSequence, asBag, size, isEmpty, =, <>, includes.

Апробация в WMP

Тестовые ограничения:

- у каждой связи есть источник
- у каждой связи есть цель
- у каждой связи есть логический id
- значения свойства Guard принимает значение по умолчанию (’true’, ’false’, ’iteration’), либо пусто

Апробация в WMP

```
--Invariants for all links
context Link
inv hasSource:
 not self.jointObject.attributes.source.id.oclIsUndefined()
inv hasTarget:
 not self.jointObject.attributes.target.id.oclIsUndefined()
def: guardValue = self.changeableProperties.Guard.value
inv defaultGuardValue:
 Bag{'false', 'true', 'iteration', ''}->includes(guardValue)
inv hasLogicalId: not self.logicalId.oclIsUndefined()
```

Заключение

- Реализован парсер языка OCL
- Создан интерпретатор OCL-выражений
- Реализованные парсер и интерпретатор апробированы в проекте WMP