# РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПАРСИНГА ТЕКСТОВ И ГЕНЕРАЦИИ UML МОДЕЛЕЙ

студентка: Моисеенко С.А.

руководитель: доц. Ермолаев В.А.

## Цель работы:

Разработка алгоритмического и программного обеспечения для получения структурированных представлений знаний из коротких семантически насыщенных текстов на естественном (английском) языке.

## Задачи:

- разработка эвристики и алгоритма преобразования текстов на естественном (английском) языке в модель представления данных на языке UML диаграммы классов
- разработать ПО реализующее эти преобразования в репрезентацию на языке XMI

# Технологии, библиотеки

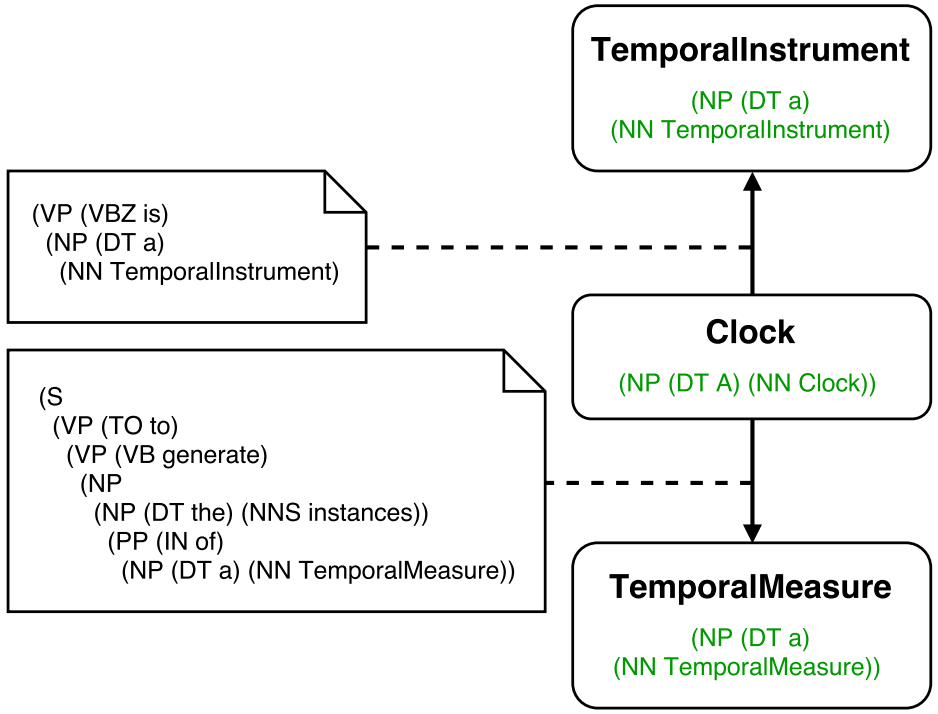
- Java
- Maven
- Stanford Core NLP
- JGraphT
- JAXB
- XMI, XML, UML

# Новизна

1. Использование отношений NP,VP в Stanford Core NLP для трансформации текста
2. Разработка правил преобразования

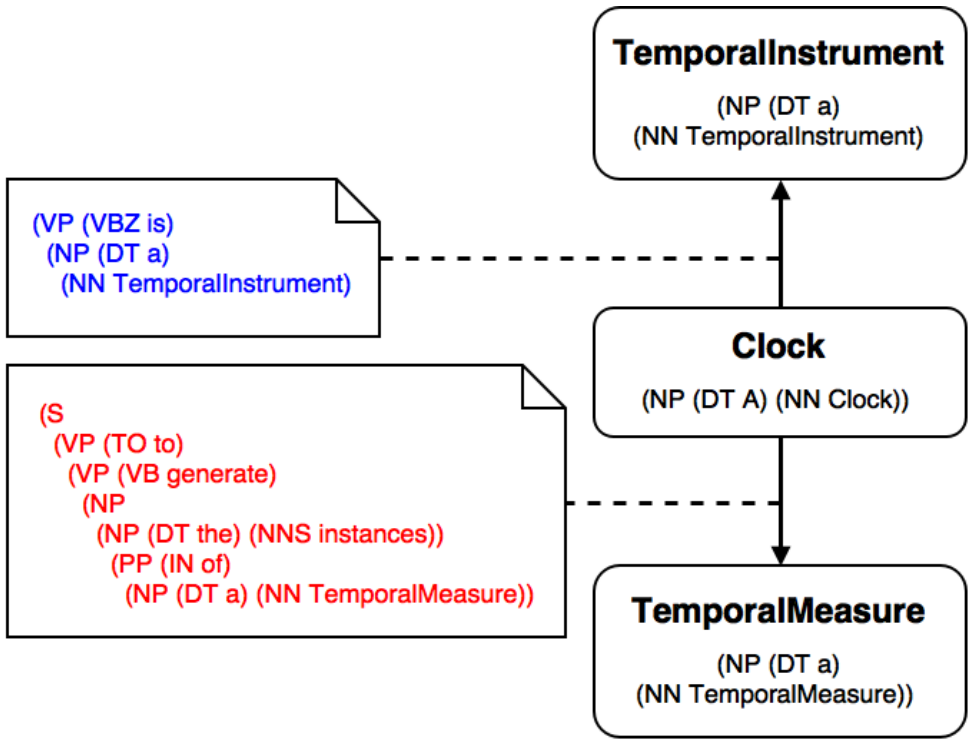
# 1. NP – вершины

```
(ROOT
  (S
    (NP (DT A) (NNClock))
      (VP (VBZis)
        (NP (DT a) (NNTemporalInstrument)
          (S
            (VP (TO to)
              (VP (VBgenerate)
                (NP
                  (NP (DT the)
(NNSinstances))
                  (PP (IN of)
                    (NP (DT a) (NNTemporalMeasure))
 (. .)))
```



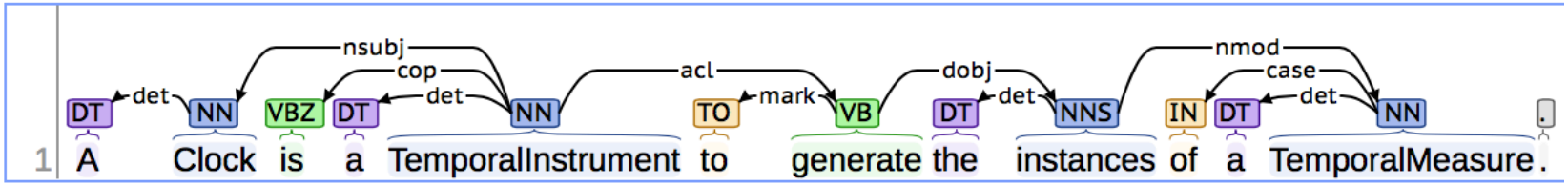
# VP – ребра

```
(ROOT
  (S
    (NP (DT A) (NNClock))
      (VP (VBZis)
        (NP (DT a) (NNTemporalInstrument)
          (S
            (VP (TO to)
              (VP (VB generate)
                (NP
                  (NP (DT the)
(NNSinstances))
                  (PP (IN of)
                    (NP (DT a) (NNTemporalMeasure))
(. .)))
```

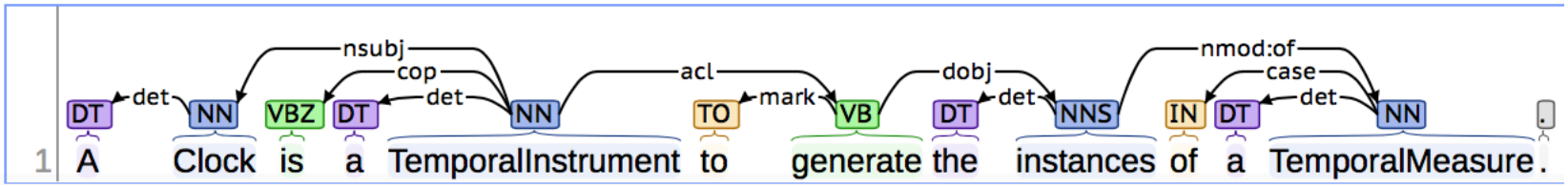


# Предыдущие решения реализации парсинга зачастую использовали привязку к зависимостям

**Basic Dependencies:**




**Enhanced Dependencies:**




**Недостаток:** обход такого графа часто приводил к большому количеству рекурсий, что значительно влияло на время выполнения алгоритма.

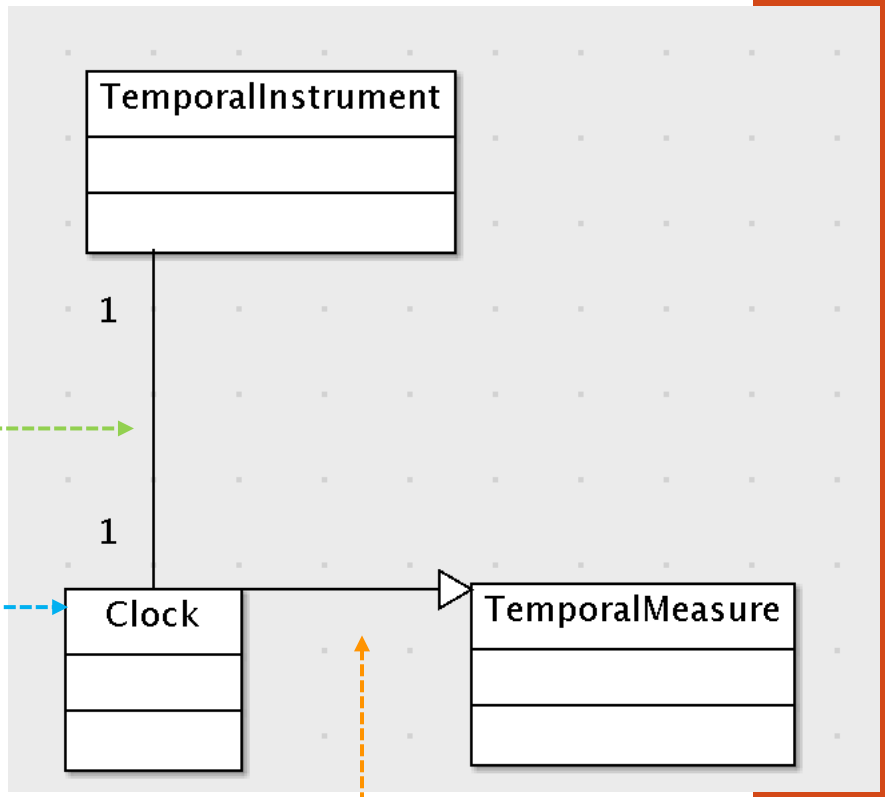
# 2. Правила преобразования промежуточного графа в UML граф

| Части речи | UML сущности |
|---|---|
| **NN, NNP, PRP, NNS** | Классы |
| **JJ, CD, RB** | Атрибуты классов |
| **VBP, VBN, VBG, IN, TO, VBZ, ADVP, VB** | Зависимости между классами (ассоциация, агрегация, генерализация) |
| **ADJP, PP, SBAR** | Дополнительная информация для зависимостей между классами, которая влияет на их последующую конвертацию |
| **IN** | Указывает на агрегацию или генерализацию в зависимости контекста |
| **CC** | Соединение одинаковых по типу зависимостей |

# Полученные результаты

A Clock is a TemporalInstrument to generate the instances of a TemporalMeasure.

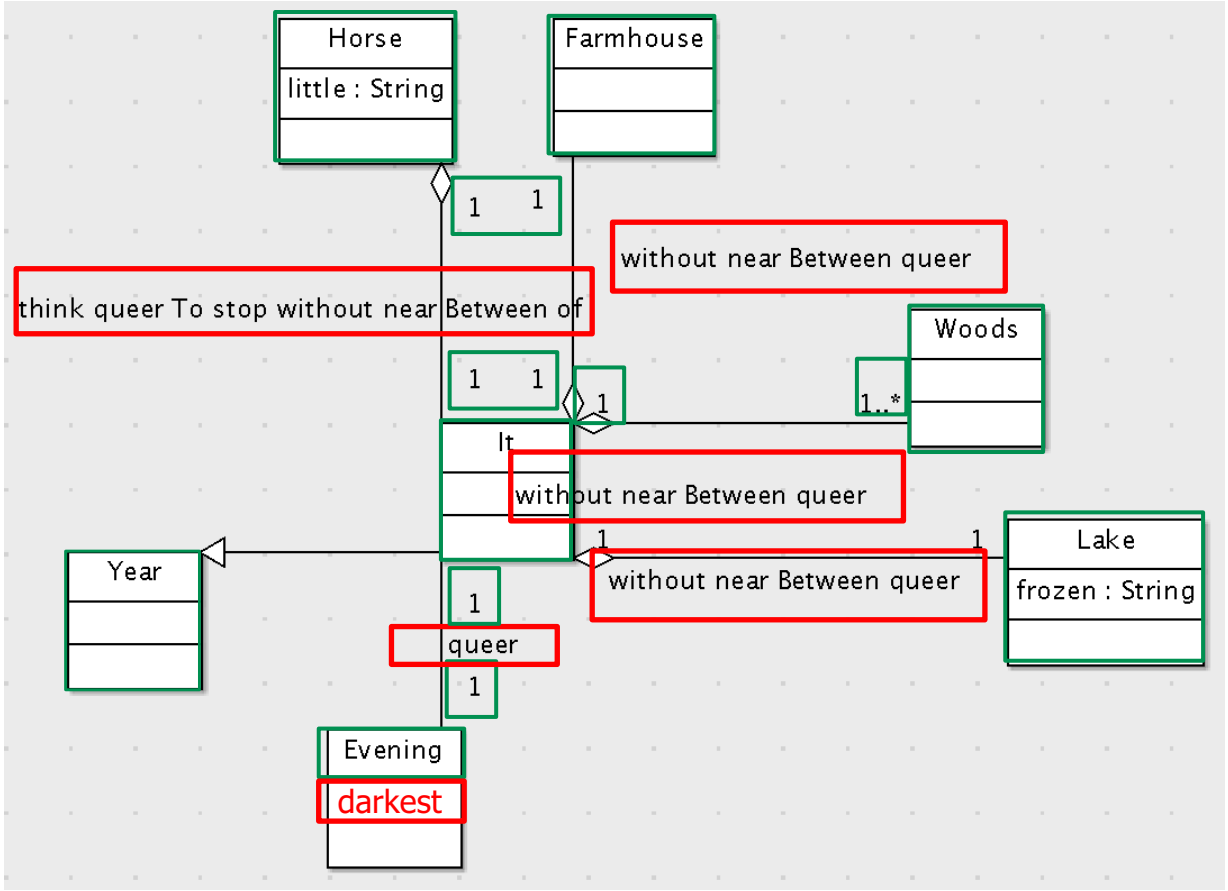
```
(ROOT
  (S
    (NP (DT A) (NN Clock))
    (VP (VBZ is)
      (NP (DT a) (NN TemporalInstrument)
        (S
          (VP (TO to)
            (VP (VB generate)
              (NP
                (NP (DT the) (NNS instances))
                (PP (IN of)
                  (NP (DT a) (NN TemporalMeasure))
(. .)))
```



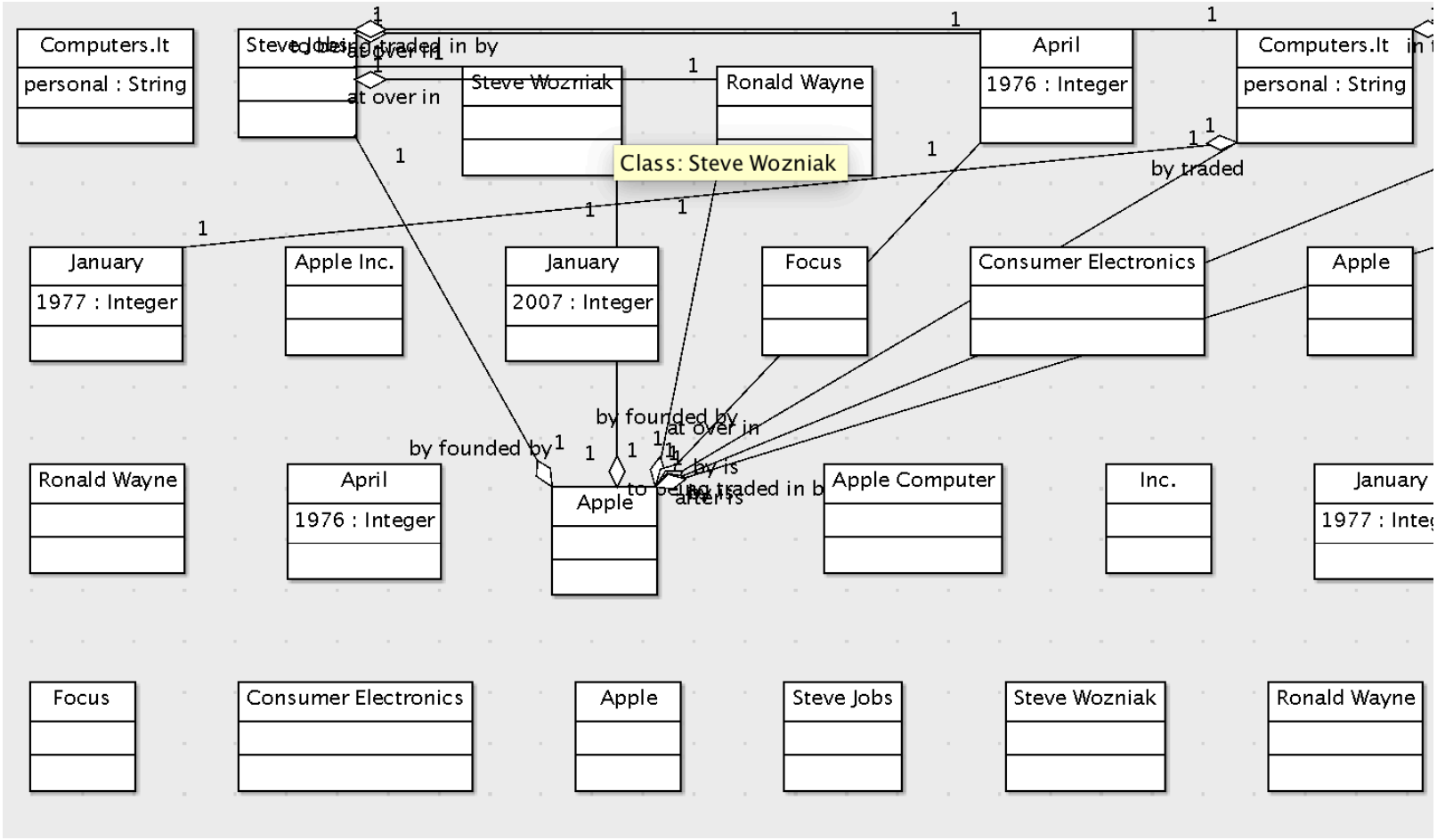
# Stopping by Woods on a Snowy Evening

My little horse must think it queer
To stop without a farmhouse near
Between the woods and frozen lake
The darkest evening of the year.

**Robert Frost**



# Разумные границы применимости подхода



# Выводы:

- разработаны правила преобразования текста в модель представления данных на языке UML диаграммы классов;
- разработан алгоритм трансформации текста в UML диаграммы;
- разработано ПО выполняющее базовые функции преобразования текста на естественном языке в UML диаграммы.