Вычисление типов в императивных динамически типизированных языках.

Михаил Калугин, студент 3 курса ММФ

Научные руководители:
Игорь Николаевич Скопин
Андрей Викторович Платов

В рамках проекта Dynamic Languages Toolkit (www.eclipsedltk.org),
Xored Software, Inc.

8 декабря 2006

Проблемы динамической типизации

- Сложность чтения программ
- Ошибки
- Низкая скорость получаемого кода
- Сложность анализа

```
class Foo:
    def doo(self):
        print "wow"

def callFoo(a):
    a.doo()

z = Foo()
callFoo(z)

x = 5
x = “string”
```

Использование информации о типах

- Верификация
- IDE
  - Code completion
  - Caller/callee иерархии
  - Подсказки о типах в outline, …
- Оптимизация

Цель

Система для вычисления типов.

Основные задачи

- **Разработка архитектуры**
- Реализация общей части
  - Eclipse
  - Dynamic Languages Toolkit
- Реализации для Python, Ruby

Архитектура

- Обобщенность
  - Модель
  - Алгоритм
- Расширяемость
  - Python, Ruby
  - Любой другой динамический язык
- Управляемость
  - Скорость
  - Точность

Существующие подходы

- Управляемый запросами анализ (Alexander Spoon, Demand-Driven Type Inference with Subgoal Pruning)
  - Задачи и подзадачи, отсечение
- Итеративный анализ, построение графа ограничений (Ole Agesen, Concrete Type Inference: Delivering Object-Oriented Applications)
- Type feedback

Основные задачи

- Разработка алгоритма
- **Реализация общей части**
  - **Eclipse**
  - **Dynamic Languages Toolkit**
- Реализации для Python, Ruby

Dynamic Languages Toolkit

- Общая модель
- Работа с кодом
  - Поиск
    - Caller/callee иерархии
    - Ссылки/объявления
  - Рефакторинг
- Отладка и запуск
- Интерфейс
  - Редактор, outline, страницы настроек
- Eclipse
- Open Source
- Языки:
  - Tcl
  - Python
  - Ruby
  - Perl

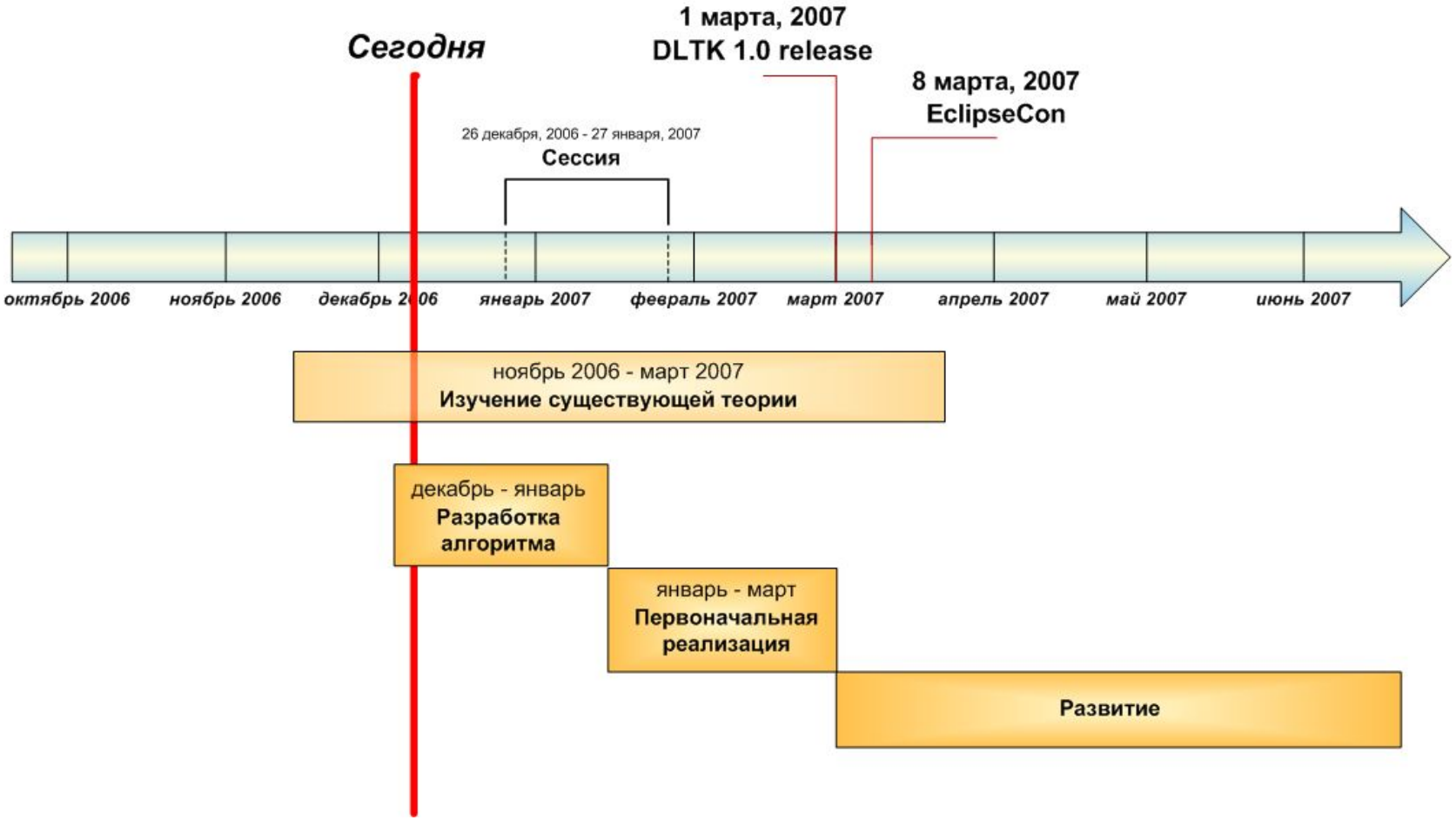
Основные задачи

- Разработка алгоритма
- Реализация общей части
  - Eclipse
  - Dynamic Languages Toolkit
- **Реализации для Python, Ruby**

Проблемы

- Модульность
- eval-блоки
- Глобальные переменные
- Метапрограммирование
- Многопоточность

План работы


Сегодня
1 марта, 2007
DLTK 1.0 release
8 марта, 2007
EclipseCon
26 декабря, 2006 - 27 января, 2007
Сессия
октябрь 2006
ноябрь 2006
декабрь 2006
январь 2007
февраль 2007
март 2007
апрель 2007
май 2007
июнь 2007
ноябрь 2006 - март 2007
Изучение существующей теории
декабрь - январь
Разработка алгоритма
январь - март
Первоначальная реализация
Развитие

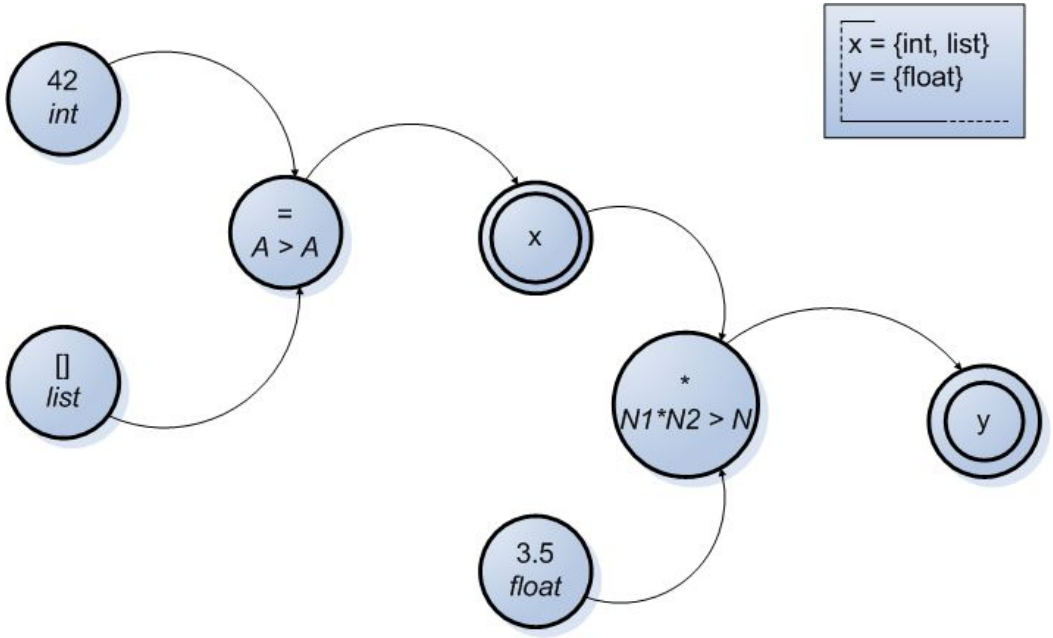
Спасибо за внимание

Вопросы?

Алгоритм

- Управляемый запросами анализ (Alexander Spoon, Demand-Driven Type Inference with Subgoal Pruning)
  - Задача, подзадача
    - Caller/callee иерархии
    - Тип (литералы, указатели, self, присваивания, параметры, вызовы)
    - Использования
  - Отсечение подзадач
    - Время
    - Глубина
  - Реализация для Smalltalk

Алгоритм

- Итеративный анализ, построение графа ограничений (Ole Agesen, Concrete Type Inference: Delivering Object-Oriented Applications)

```
def foo():
    x = 42
    y = x * 3.5
    x = []
```