Эволюция архитектуры компиляции

Евгений Зуев, ETH Zürich

Современные информационные технологии и ИТ-образование Москва, МГУ, 21 сентября 2005

План

Традиционный компилятор

Многоплатформенная архитектура

Задача компиляции

«Продвинутая» архитектура

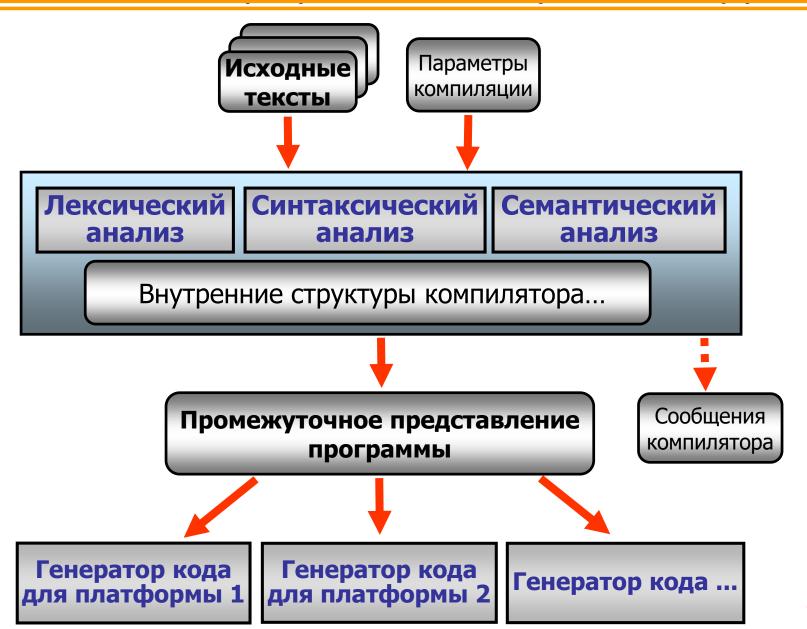
Семантическое представление: двоичный или открытый формат?

Интеграция компилятора в среду разработки

Традиционный компилятор



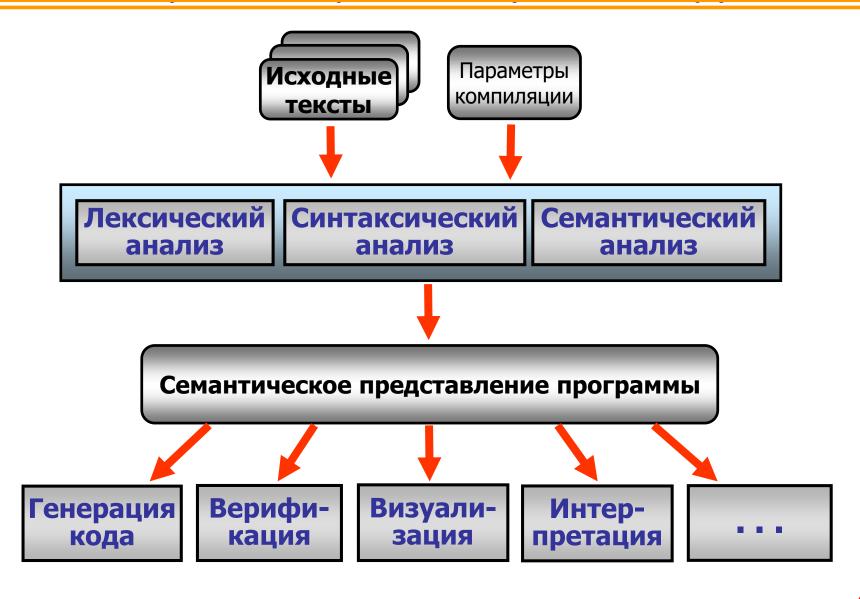
Многоплатформенная архитектура



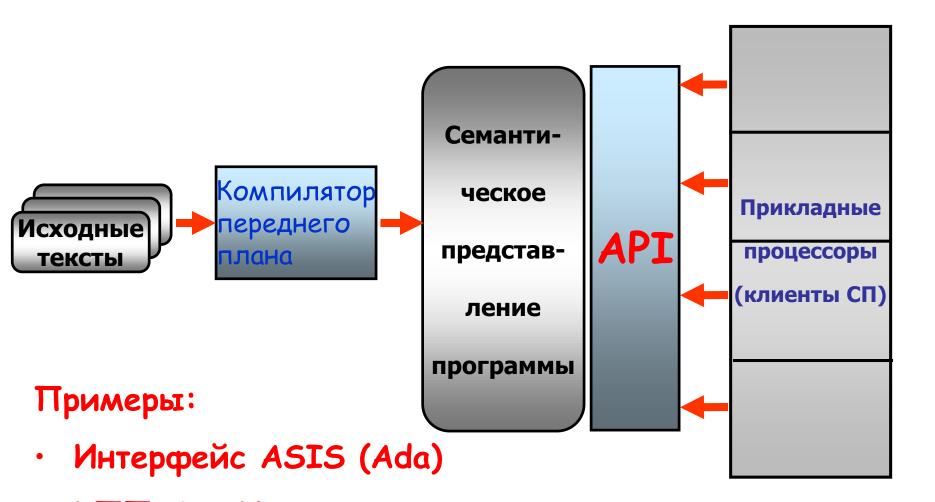
Задача компиляции

- Генерация объектного кода (компиляция в узком смысле)
- Верификация
- · Статический анализ
- Рефакторинг
- Интерпретация
- «Визуализация»
 (понимание программ человеком)

«Продвинутая» архитектура



Семантическое представление: двоичный формат и API



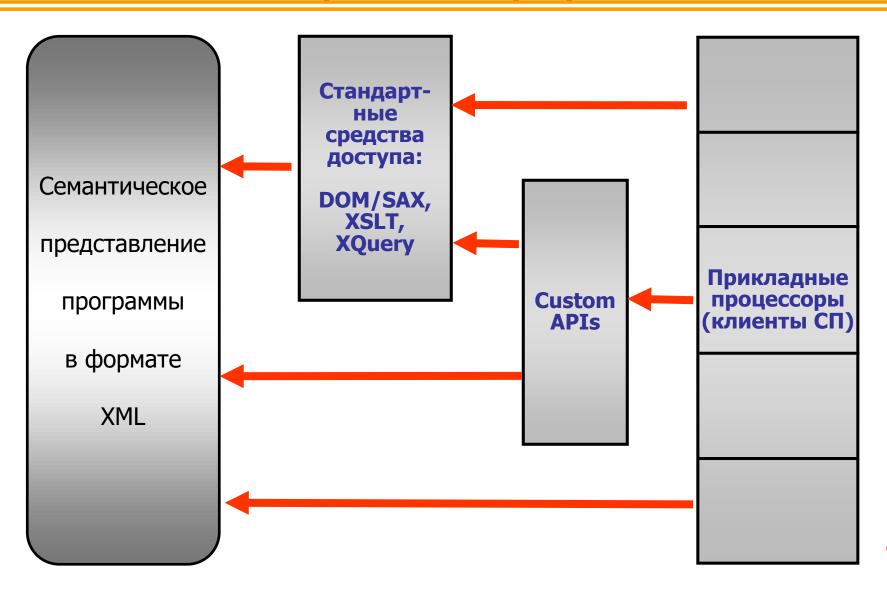
КПП С++ Интерстрон

Семантическое представление: открытый формат

Почему XML?

- Открытый формат
- Расширяемость
- Предельно простая модель
- · Промышленный стандарт де-факто
- Широкий спектр средств доступа

Семантическое представление: открытый формат



XML-представление: пример

```
while ... loop
  x := ...;
                    <while-statement>
  P(...);
                       <condition>
end loop;
                          <expression> ... </expression>
                       <condition>
                       <statement-sequence>
                          <assignment>
                             <name>x</name>
                             <expression> ... </expression>
                          </assignment>
                          <call>
                             <arguments>
                                <expression> ... </expression>
                             </arguments>
                          </call>
                       </statement-sequence>
                    </while-statement>
```

XML-представление: пример

```
<while-statement In="1" col="1">
   <condition>
      <expression In="1" col="7"> ... </expression>
   <condition>
   <statement-sequence>
      <assignment In="2" col="4">
         <name |n="2" col="4">x</name>
         <expression In="2" col="9"> ... </expression>
      </assignment>
      <call In="3" col="4">
         ocedure-name In="3" col="4">P
         <arguments>
             <expression In="3" col="5"> ... </expression>
         </arguments>
      </call>
   </statement-sequence>
</while-statement>
```

Интеграция компилятора в среду

- Совместное функционирование компилятора и компонент среды
- Отображение лексической, синтаксической и семантической структуры
- Автозавешение конструкций,
 «умный» поиск
- · Динамический анализ семантических связей («intellisense», идентификация типов)
- **Фоновая компиляция etc.**

Интеграция компилятора в среду

Пример свойства "Intellisense"

```
3704
   3705油
                   public static ARRAY TYPE create ( )
   3706
   3707
                        ARRAY TYPE arrayType = new ARRAY TYPE();
   3708
   3709
                        arrayType.base type = null;
                    // arrayType.dimensions -- already initialized
   3710
                        arrayType.enclosing = null;
   3711
                       arrayType.m
   3712
   3713
                       arrayTy 🛶 GetType
                    // arrayTy 🙌 internal_type
   3714
                                                     TREE.scanner.getSourceCo:
   3715
                                isOpen
   3716
                                MemberwiseClone
                        return
   3717
                                  modifiers
                                                    MODIFIERS NODE, modifiers
   3718
                                 🕡 name
                   public void @ node
   3719 占
                                                    lemType )
                                🖦 report
   3720
                       for ( i 1 to report_extra
   3721
                                                    hs.Length: i<n: i++ \
                                🖦 report short
Output
Ceneral
```

Архитектура ССІ-компилятора

