Разработка и реализация веб-среды визуальной разработки фрагментированных программ

Автор: Ижицкий Р. Л.
Научный руководитель: Киреев С. Е., ИВМиМГ СО РАН

Новосибирск

29.05.2019 г.

Уровень программирования

- Проблема повышения уровня программирования
- Особо сложная для параллельного программирования

Технология фрагментированного программирования

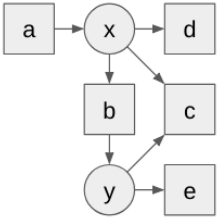
- Проект “Технология фрагментированного программирования”
  - ИВМиМГ СО РАН
  - цель: повышение уровня параллельного программирования
  - система фрагментированного программирования LuNA
  - язык LuNA, основанный на высокоуровневом представлении алгоритма

Язык LuNA

- Фрагментированный алгоритм - элементами алгоритма являются целые блоки данных (фрагменты) и операции над фрагментами
- Множество переменных единственного присваивания и операций единственного срабатывания, связанных отношениями in и out
- Исполнительная семантика — data flow (по готовности данных)
- Можно представить графом информационных зависимостей

```
df x, y;

cf a: func1(out: x)
cf b: func2(in: x, out: y)
cf c: func3(in: x, y)
cf d: func4(in: x)
cf e: func4(in: y)
```



Проблема

Низкая доступность системы LuNA внешним пользователям

- Требуется установка системы LuNA
- Существует определенный порог вхождения в язык LuNA

Цель и задачи

Цель: создание визуальной веб-среды разработки LuNA-программ

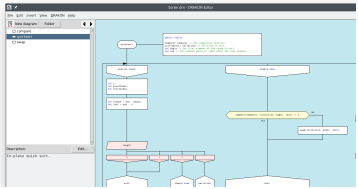
Задачи:

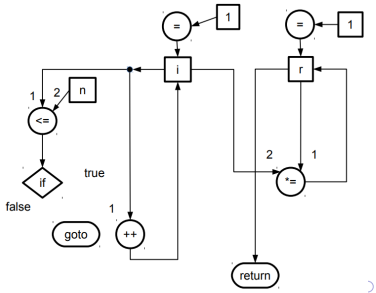
- Разработка визуального языка на базе языка LuNA
- Реализация визуальной веб-среды разработки
- Реализация транслятора из визуального представления в текстовое

Родственные работы

Примеры существующих визуальных языков

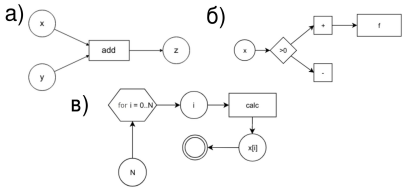
- DRAKON (control-flow) (сверху)
- LabVIEW (data-flow)
- Simulink (data-flow)
- Pifagor (data-flow) (снизу)





Визуальный язык

- (а) Переменная - круг с именем переменной
- (а) Операция - прямоугольник с именем реализующей ее подпрограммы
- (а) Информационной связь между компонентами - сплошная дуга
- (б) Условный оператор - блок условия и два индикатора ветки
- (в) Цикл - один блок начала цикла (шестиугольник) и несколько блоков границ цикла (двойная окружность)



Выбор средства разработки

- Язык: JavaScript
- Библиотека рисования графов

Требования к библиотеке

- Возможность рисовать заданное графическое представление фрагментированного алгоритма, спроектированное на основе языка LuNA
- Бесплатность
- Возможность динамического редактирования графов
- Развитая документация библиотеки

Выбор библиотеки

Не свободно распространяемые

JointJS	Rappid	MxGraph	GoJS	jsUML2
Mindfusion Diagram	Nomnoml	Mermaid.js	Diagram.js	State.js
D3	Raphael	Draw2D	Fabric.js	Paper.js
JsPlumb	p5.js	Cytoscape.js	dagre-d3	vis.js

Выбор библиотеки

Невозможность рисовать заданное графическое представление

		MxGraph		jsUML2
	Nomnoml	Mermaid.js	Diagram.js	State.js
D3	Raphael	Draw2D	Fabric.js	Paper.js
	p5.js	Cytoscape.js	dagre-d3	vis.js

Выбор библиотеки

Невозможность динамического редактирования графов

		MxGraph		
		Mermaid.js	Diagram.js	
D3	Raphael	Draw2D	Fabric.js	Paper.js
	p5.js		dagre-d3	vis.js

Выбор библиотеки

Векторный редактор графических примитивов

		MxGraph		
			Diagram.js	
D3	Raphael	Draw2D	Fabric.js	Paper.js
	p5.js			vis.js

Выбор библиотеки

Неудобный интерфейс для пользователя (Создание вершин и ребер)

		MxGraph		
			Diagram.js	
D3		Draw2D		
				vis.js

Выбор библиотеки

Небольшая документация

		MxGraph		
			Diagram.js	
		Draw2D		

Выбор библиотеки

		MxGraph		

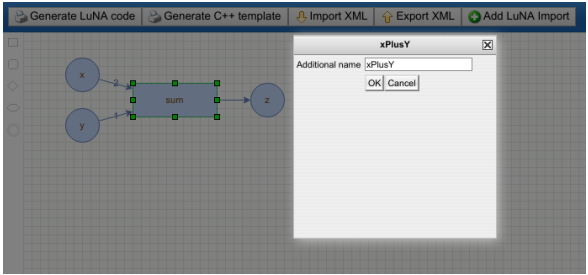
Внутреннее представление элементов

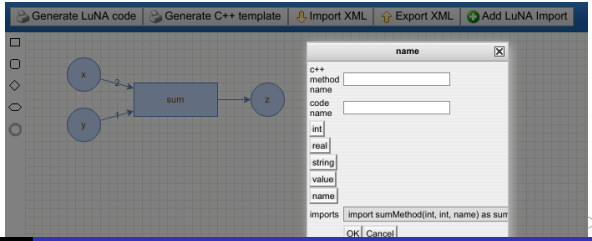
- XML-элементы: Variable, Operation, Condition, SubConditionPositive, SubConditionNegative, LoopIn, LoopOut
- Атрибуты XML-элементов
- Пример XML-элемента:

```
<Operation name="sum" addName="xPlusY" id="4">
    <mxCell style="" vertex="1" parent="1">
    <mxGeometry x="5" y="8" width="8" height="4" as="geometry"/>
    </mxCell>
</Operation>
```

Элементы IDE

- Меню с командами
- Панель элементов
- Холст для элементов
- Атрибуты элементов (в отдельном окне)
- Список подпрограмм





Генерация кода

Программа

- Код на LuNA
  - Структура разработанного алгоритма
- Код на C++
  - Реализация операций из LuNA-программы
  - На основе списка подпрограмм

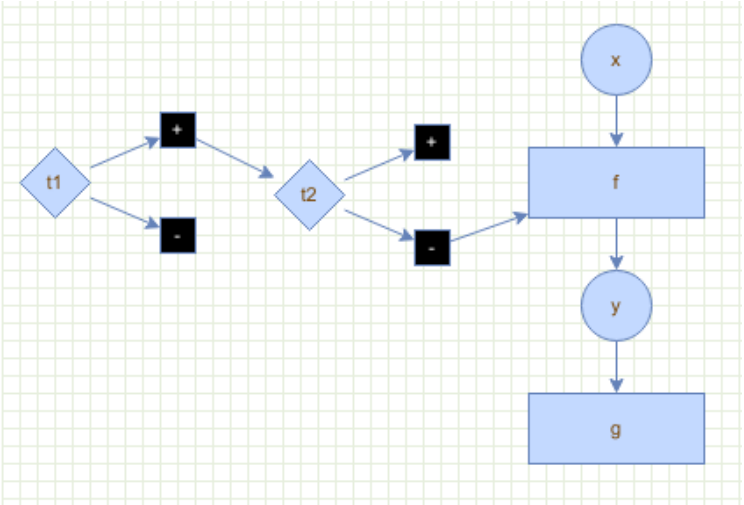
- LuNA

```
sub main() {
df x, y, z;
cf xPlusY: sum(#in y, x, #out z);
}
```

- C++

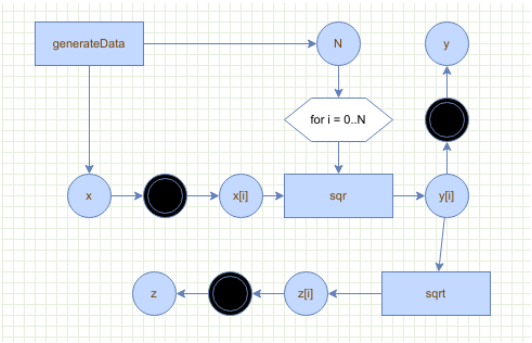
```
#include "ucenv/ucenv.h"
extern "C"
void sumMethod(int i0, int i1,
               OutputDF& dfo0) {
dfo0.setValue<int>(0);
}
```

Условные операторы




```
sub main() {
if (!t2 && t1) {
f(#in x, #out y);
}
g(#in y);
}
```

Циклы




```
sub main() {
df x, N, y, z;
for i = 0..N{
sqr(#in x[i], #out y[i]);
sqrt(#in y[i], #out z[i]);
}
generateData(#in #out N, x);
}
```

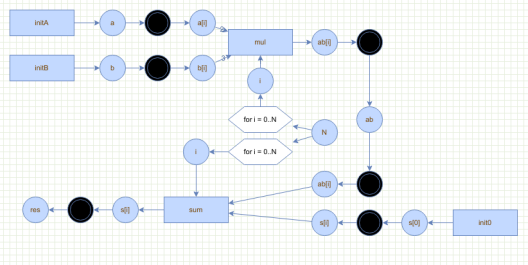
Алгоритм генерации LuNA-программы

1. Получение списка всех элементов (вершин и ребер)
2. Для каждого стартового элемента цикла генерация кода этого цикла (с пометкой посещенных элементов)
   - Обход каждого цикла в глубину по прямым дугам, обрабатывая только операции
3. Генерация кода для непомеченных операций

Генерация кода для операций:

1. Для операции ищутся все условные операторы, от которых она зависит, и при их наличии формируется код условия
2. Генерация кода по входным и выходным переменным этой операции

Пример: скалярное произведение




```
sub main() {
df a, b, ab, s[0], res, N;
for i = 0..N{
mul(#in i, a[i], b[i], #out ab[i]);
}
for i = 0..N{
sum(#in ab[i], i, s[i], #out s[i]);
}
initA(#in #out a);
initB(#in #out b);
init0(#in #out s[0]);
}
```

Результаты

- Создан прототип визуального языка, позволяющий разрабатывать фрагментированные алгоритмы, поддерживающие невложенные циклы
- Создан прототип среды разработки, позволяющий разработанный визуальный язык
- Реализована трансляция графического представления фрагментированных алгоритмов в текст программы на языке LuNA и в шаблон модуля на языке C++
- Реализована возможность сохранять и загружать визуальные программы на разработанном визуальном языке

Планы

- Реализация вложенных циклов
- Реализация while-циклов
- Проверка на ошибки
- Возможность запуска кода из IDE
- Повышение удобства интерфейса

Апробация работы

Работа была представлена на 57-й Международной научной студенческой конференции «Студент и научно-технический прогресс» (МНСК-2019) в виде устного доклада

Спасибо за внимание

- Репозиторий проекта: https://github.com/ksilobait/Graph-LUNA
- Контакты:
  - ksilobait@gmail.com (Ижицкий Руслан Леонидович)
  - kireev@ssd.sscc.ru (Киреев Сергей Евгеньевич)