

# Постановка и решение изобретательских задач в программировании на основе методов ТРИЗ

Струсь Глеб Игоревич

532 группа

Научные руководители:

ст. пр. Одинцов И.О.

ст.н.с. Рубин М.С.

Санкт-Петербургский Государственный Университет

2010

# Цель и задачи работы

- **Цель:** обоснование применимости методов ТРИЗ для решения изобретательских задач в программировании и создание программного продукта, основанного на алгоритмах постановки изобретательских задач
- **Задачи:**
  - Исследовать Алгоритм Решения Изобретательских Задач (АРИЗ)
  - Обосновать применимость АРИЗ для решения изобретательских задач в программировании
  - Исследовать алгоритмы постановки изобретательских задач
  - Определить методы анализа систем, необходимые для осуществления алгоритмов постановки
  - Адаптировать выбранные алгоритмы и методы для программной реализации
  - Создать программный продукт, основанный на выбранных алгоритмах

# Основы ТРИЗ

- АРИЗ – инструмент решения задач, содержащих противоречие, используется в основном в технической сфере.

В программировании существует множество сложных задач, для решения которых могли бы подойти методы ТРИЗ.

- Формулировка задачи в программировании, как задачи с противоречием, часто затруднена.

ТРИЗ располагает методами, позволяющими упростить постановку изобретательской задачи.

# Решение задач из программирования по АРИЗ 2010

Успешное решение ряда задач из области программирования убедительно доказывает возможность эффективного применения методов ТРИЗ для решения изобретательских задач в программировании.

# Задача об ускорении сортировки массива

## Постановка

Массив, содержащий целые числа, можно отсортировать методом пузырька. Время выполнения алгоритма растет квадратично длине массива. Как ускорить сортировку, используя существующий алгоритм сортировки?

## Решение

Разбить массив на части и применить алгоритм сортировки пузырьком к каждой части по отдельности.

# Задача о многорежимности в графическом редакторе

## Постановка

В редакторе геометрических фигур существует два режима редактирования, между которыми приходится часто переключаться. Первый режим позволяет передвигать вершины фигуры. Второй – добавлять новые вершины путем разбиения ребра на две части. Как сделать использование этих двух режимов удобным?

## Решение

Фигура разбивается на зоны вершин и зоны ребер. В зависимости от местонахождения курсора мыши включается соответствующий режим редактирования.

# Задача о защите общедоступной программы от несанкционированного доступа

## **Постановка**

Необходимо защитить от несанкционированного использования программу, доступ к которой не ограничен. То есть гарантировать использование программы только определенным кругом лиц.

## **Решение**

Создать специальный формат входных данных.

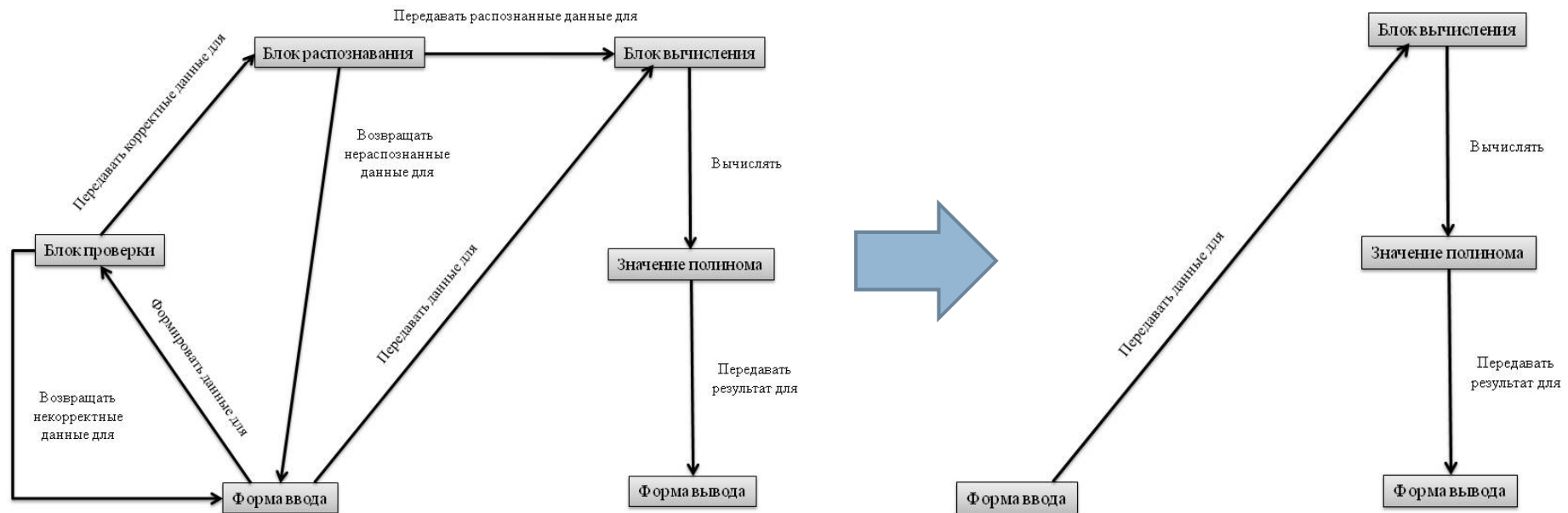
# Алгоритм проведения свертывания

- Вспомогательные алгоритмы
  - ▣ Построение компонентной модели
  - ▣ Построение структурной модели
  - ▣ Построение функциональной модели
  - ▣ Построение причинно-следственных цепочек
- Алгоритм свертывания

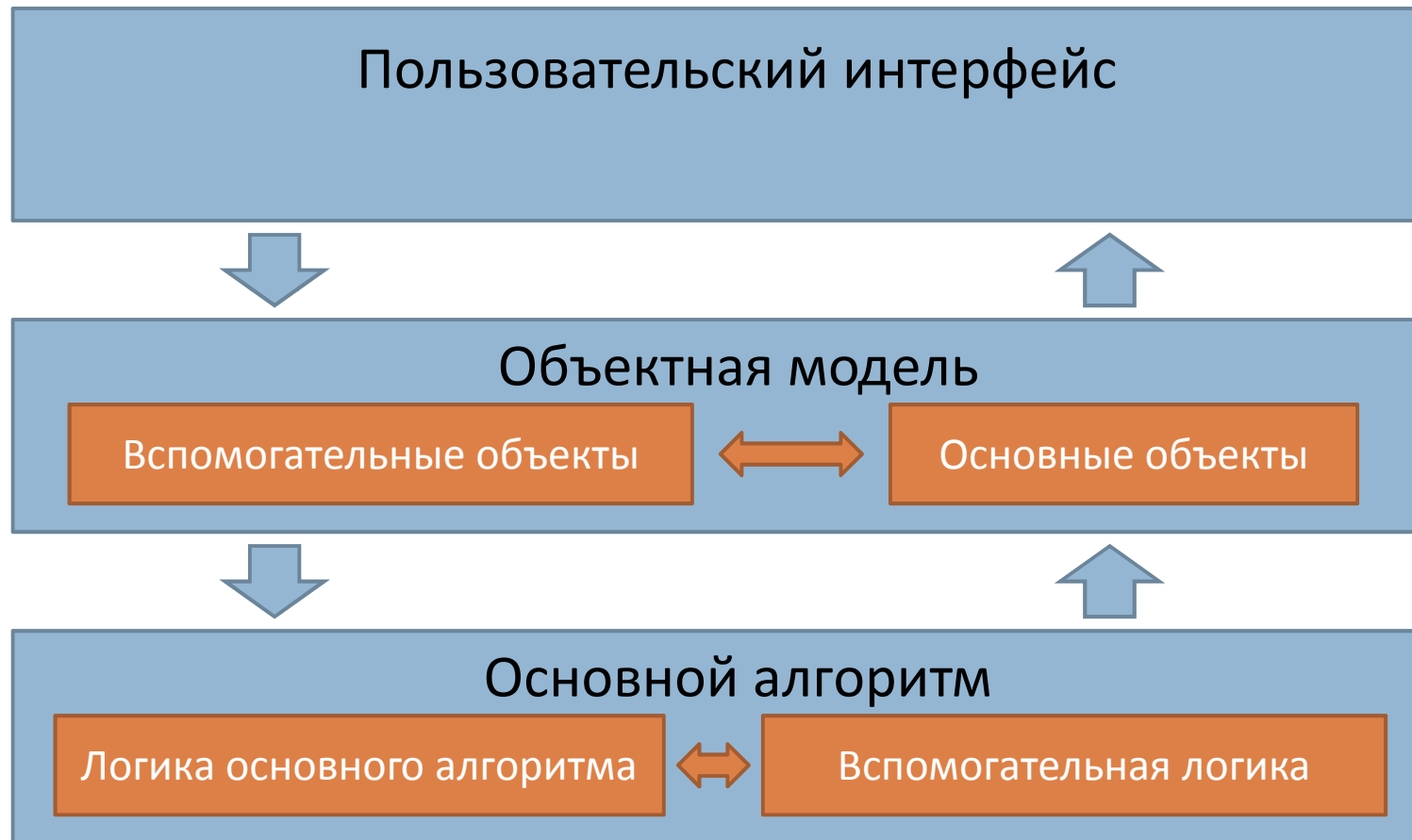


# Применение свертывания для задач из программирования

На примере программы, вычисляющей значение полинома, показана процедура свертывания с выходом на постановку изобретательской задачи и ее дальнейшим решением



# Архитектура разработанной системы



# Работа с пользовательским интерфейсом системы

The screenshot shows a software application window titled "Алгоритм свертки Вычисление полинома". The window has a menu bar with "File" and a tabbed interface with the following tabs: "Компонентная модель", "Структурная модель", "Функциональная модель", "Причинно-следственный анализ", "Диагностическая таблица", and "Граф системы".

The main area is divided into four sections:

- Название системы:** A text box containing "Вычисление полинома".
- Главная функция:** A text box containing "числить значение полинома". Below it, "Объект главной функции" is a text box containing "значение полинома".
- Компоненты системы:** A list box containing "форма ввода", "блок проверки", "блок распознавания", "блок вычисления", "значение полинома", and "форма вывода". Below the list is an empty text box and a "Добавить" button.
- Внесистемные компоненты:** An empty list box. Below it is an empty text box and a "Добавить" button.

At the bottom right of the window is a "Далее" button.

# Работа с пользовательским интерфейсом системы

Алгоритм свертки Вычисление полинома

File

Компонентная модель Структурная модель функциональная модель Причинно-следственный анализ Диагностическая таблица Граф системы

	форма ввода	блок проверки	блок распознавания	блок вычисления	значение полинома	форма вывода
форма ввода		+	+	+	—	—
блок проверки	+		+	—	—	—
блок распознавания	+	+		+	—	—
блок вычисления	+	—	+		+	—
значение полинома	—	—	—	+		+
форма вывода	—	—	—	—	+	

Далее

# Работа с пользовательским интерфейсом системы

Алгоритм свертки Вычисление полинома

File

Компонентная модель Структурная модель **Функциональная модель** Причинно-следственный анализ Диагностическая таблица Граф системы

**форма ввода**

Функция	Объект	Ранг	Уровень выполнения
формировать данные для	блок проверки	B1	Адекватный
передавать данные для	блок вычисления	B2	Адекватный
	форма ввода	O1	Недостаточный

Функция в качестве параметра

**блок проверки**

Функция	Объект	Ранг	Уровень выполнения
возвращать некорректные да...	форма ввода	B1	Недостаточный
передавать корректные данн...	блок распознавания	B3	Недостаточный
	форма ввода	O1	Недостаточный

**блок распознавания**

Функция	Объект	Ранг	Уровень выполнения
возвращать нераспознанные ...	форма ввода	B1	Недостаточный
передавать распознанные да...	блок вычисления	O2	Адекватный
	форма ввода	O1	Недостаточный

**блок вычисления**

Функция	Объект	Ранг	Уровень выполнения
вычислять	значение полинома	O1	Адекватный
	форма ввода	O1	Недостаточный

**значение полинома**

Функция	Объект	Ранг	Уровень выполнения
передавать результат для	форма вывода	O1	Недостаточный
	блок вычисления	O1	Недостаточный

**форма вывода**

Функция	Объект	Ранг	Уровень выполнения
	значение полинома	O1	Недостаточный

Далее

# Работа с пользовательским интерфейсом системы

Алгоритм свертки Вычисление полинома

File

Компонентная модель Структурная модель Функциональная модель Причинно-следственный анализ **Диагностическая таблица** Граф системы

```
graph TD; A[Большая сложность проверки и распознавания полинома] --> B[трудности проверки корректности введенной строки]; A --> C[трудности распознавания введенной строки]; B --> D[проверка на наличие запрещенных символов]; C --> E[разбиение на слагаемые]; C --> F[выделение коэффициентов]; C --> G[выделение степеней];
```

Описание недостатка

выделение степеней

Ключевой недостаток

Все компоненты системы

форма ввода  
блок проверки  
блок распознавания  
блок вычисления  
значение полинома  
форма вывода

Компоненты недостатка

блок распознавания

Далее

# Работа с пользовательским интерфейсом системы

Алгоритм свертки Вычисление полинома

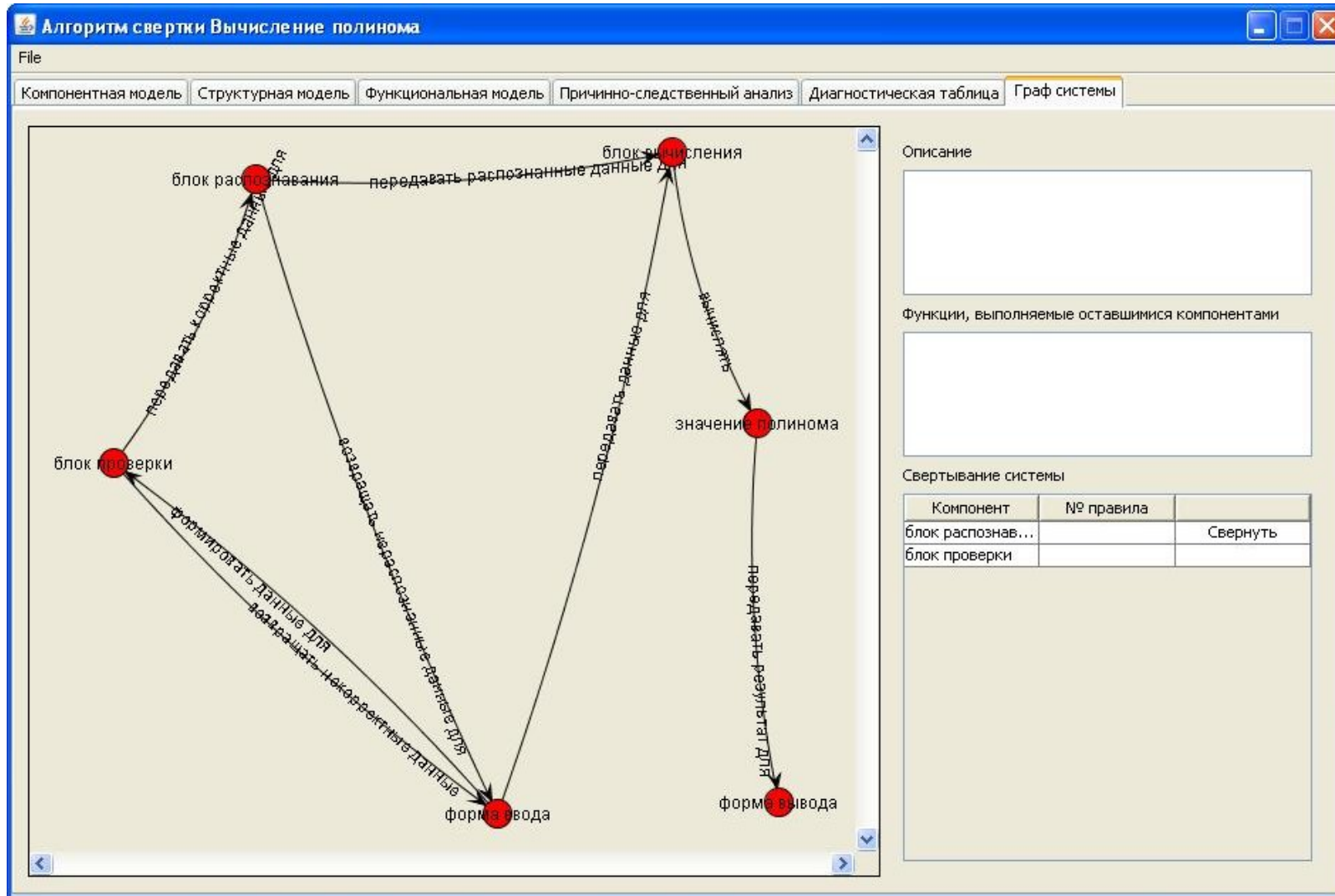
File

Компонентная модель Структурная модель Функциональная модель Причинно-следственный анализ **Диагностическая таблица** Граф системы

№	Компонент	Количество ключевых недостатков	Порядок свертывания
1	форма ввода	0	
2	блок проверки	1	2
3	блок распознавания	3	1
4	блок вычисления	0	
5	значение полинома	0	
6	форма вывода	0	

Далее

# Работа с пользовательским интерфейсом системы





# Основные результаты

- Доказана возможность применения методов ТРИЗ для постановки и решения изобретательских задач в программировании
  - Показано, что АРИЗ 2010 представляет эффективный инструмент решения задач в программировании
  - Проведена свертка системы в программировании с выходом на изобретательскую задачу
- Разработан программный продукт, основанный на алгоритмах постановки изобретательских задач, который автоматизирует вспомогательные процессы и облегчает выполнение этих алгоритмов
  - Проанализированы методы постановки изобретательских задач
  - Адаптирован для программной реализации алгоритм свертки и сопутствующие ему алгоритмы



Спасибо!

Вопросы?