

# **UniMod: метод и средство разработки реактивных объектно-ориентированных программ с явным выделением состояний**

**Вадим Гуров, Максим Мазин,  
Андрей Нарвский, Анатолий Шальто**

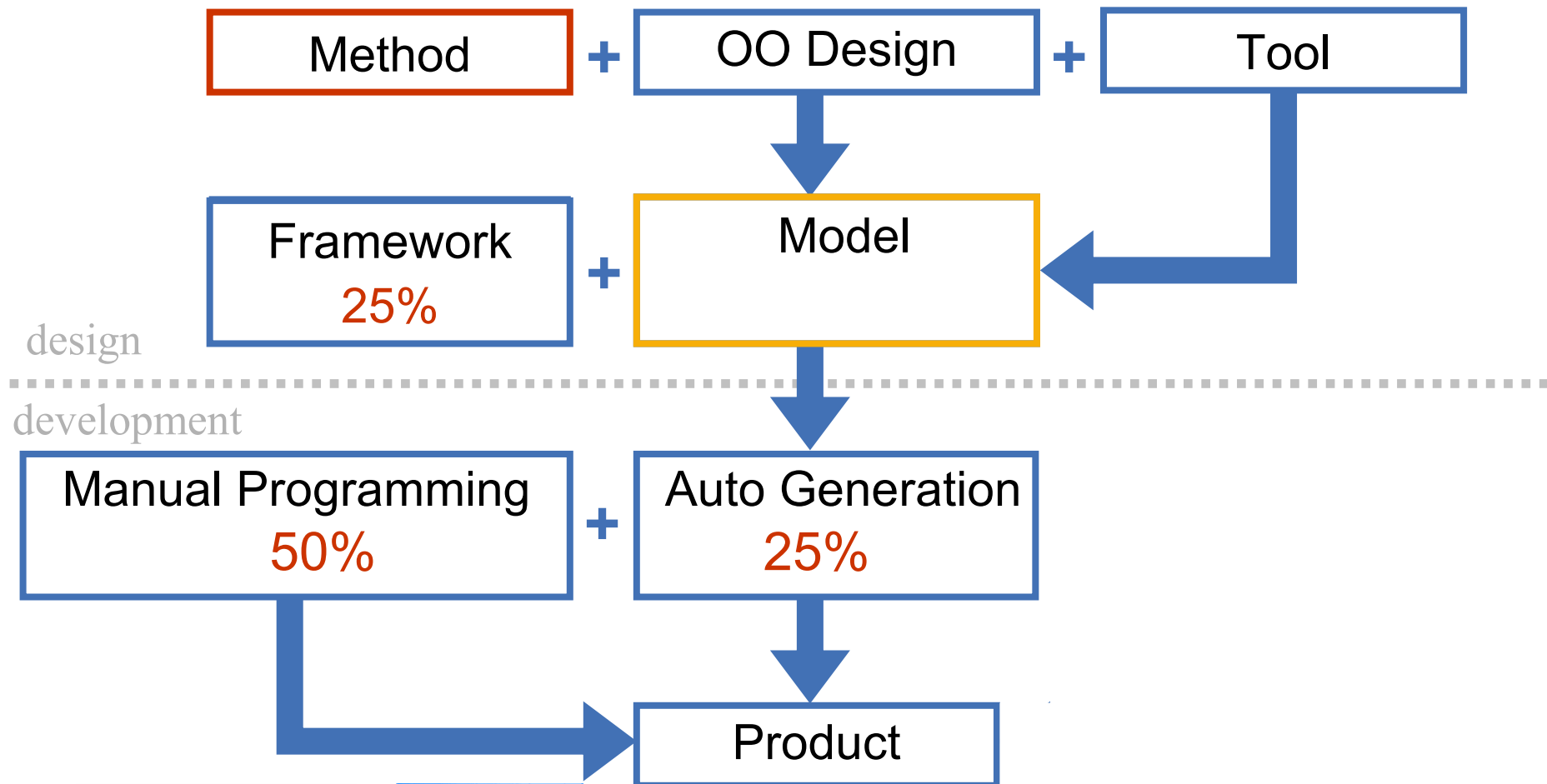
**СПб ГУ ИТМО  
eDevelopers Corp.**



## Постановка задачи (1)

- UML + SWITCH-технология
- Затрагиваются только фазы проектирования и реализации
- Рассматриваются только реактивные системы
- Основное внимание уделяется моделированию логики поведения системы
- Исполняемый графический язык
- Отсутствие обратной генерации модели по коду
- Проверка правильности модели
  - Полнота и непротиворечивость графа переходов
  - Достижимость состояний
  - Правильность построения потока данных

## Постановка задачи (2)



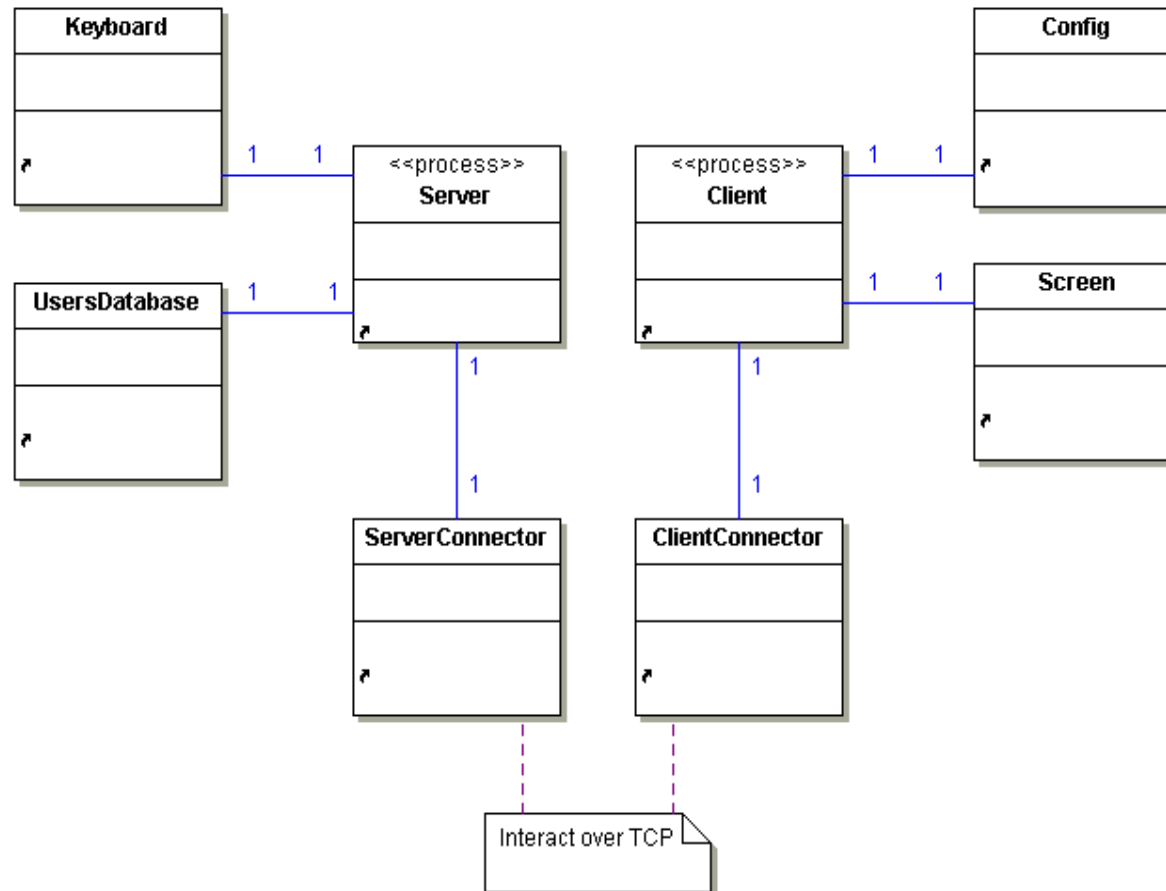
# Предлагаемый метод

Проектирование

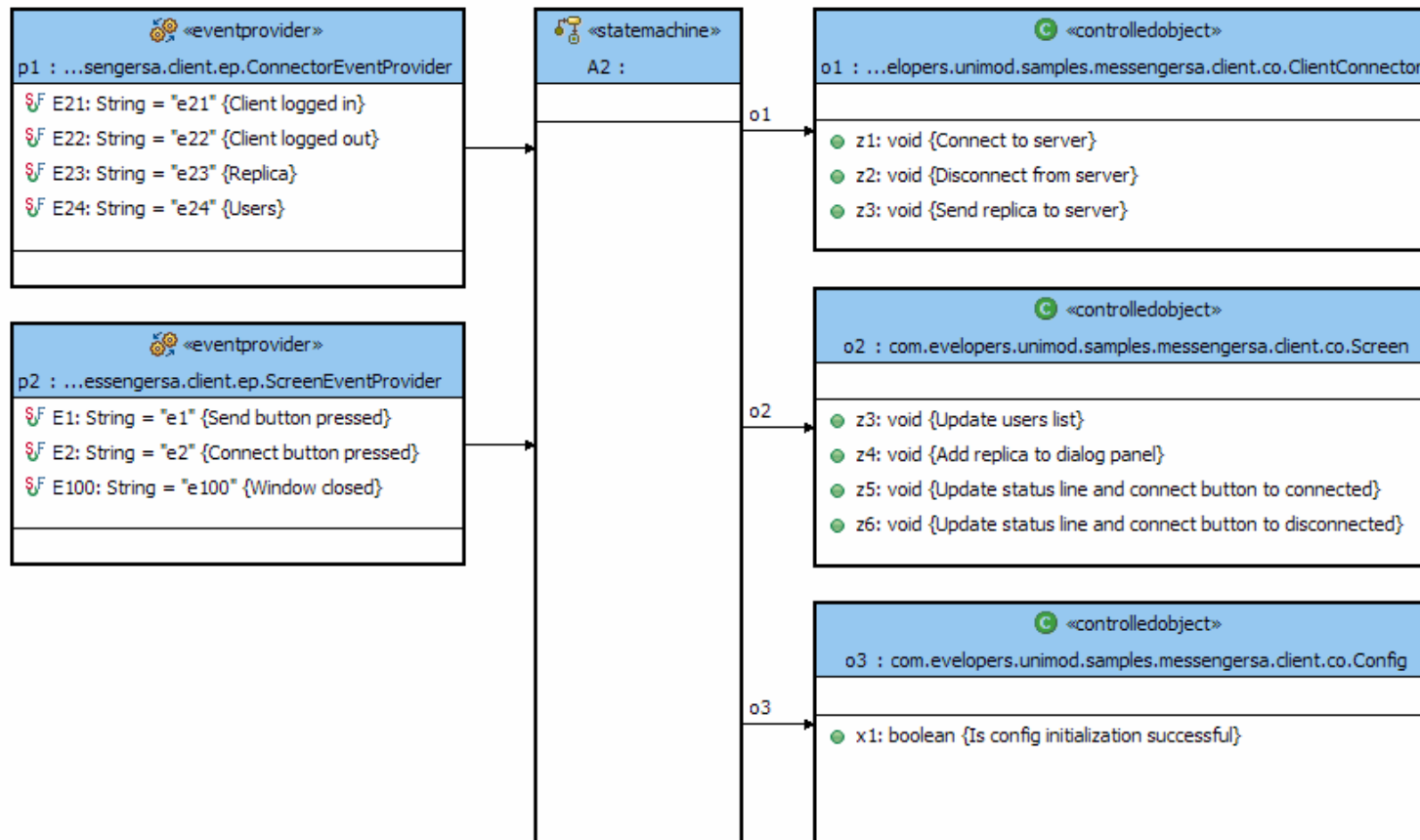
Реализация

1. Создать концептуальную модель предметной области в виде UML диаграммы классов используя классические методы
  2. Выделить поставщики событий, автоматы и объекты управления. Оставшиеся классы – контейнеры для данных.
  3. Создать диаграмму связей автоматов
    - ↘ Поставщики событий поместить справа
    - ↘ Объекты управления слева
    - ↘ Автоматы посередине
    - ↘ Создать связи между поставщиками событий, автоматами и объектами управления
  4. В поставщиках событий определить множество производимых событий ( $e_x$ )
  5. В объектах управления создать два множества методов соответствующих:
    - ↘ Входным воздействиям ( $x_j$ )
    - ↘ Выходным воздействиям ( $z_k$ )
  6. Для каждого автомата создать диаграмму состояний
- 
7. Реализовать поставщики событий и объекты управления вручную или выбрать готовые из репозитория
  8. Запустить модель прямо из среды разработки, либо преобразовать модель в XML-описание для последующей интерпретации, либо преобразовать модель в код на целевом языке программирования для последующей компиляции

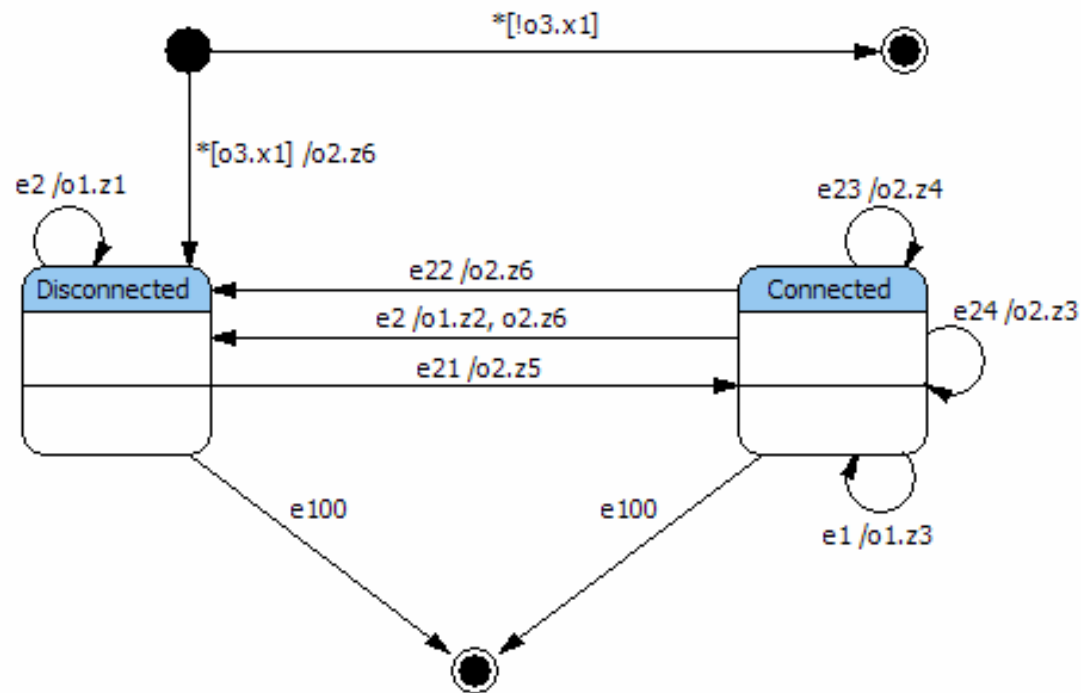
# Пример использования метода (1)



# Пример использования метода (2)

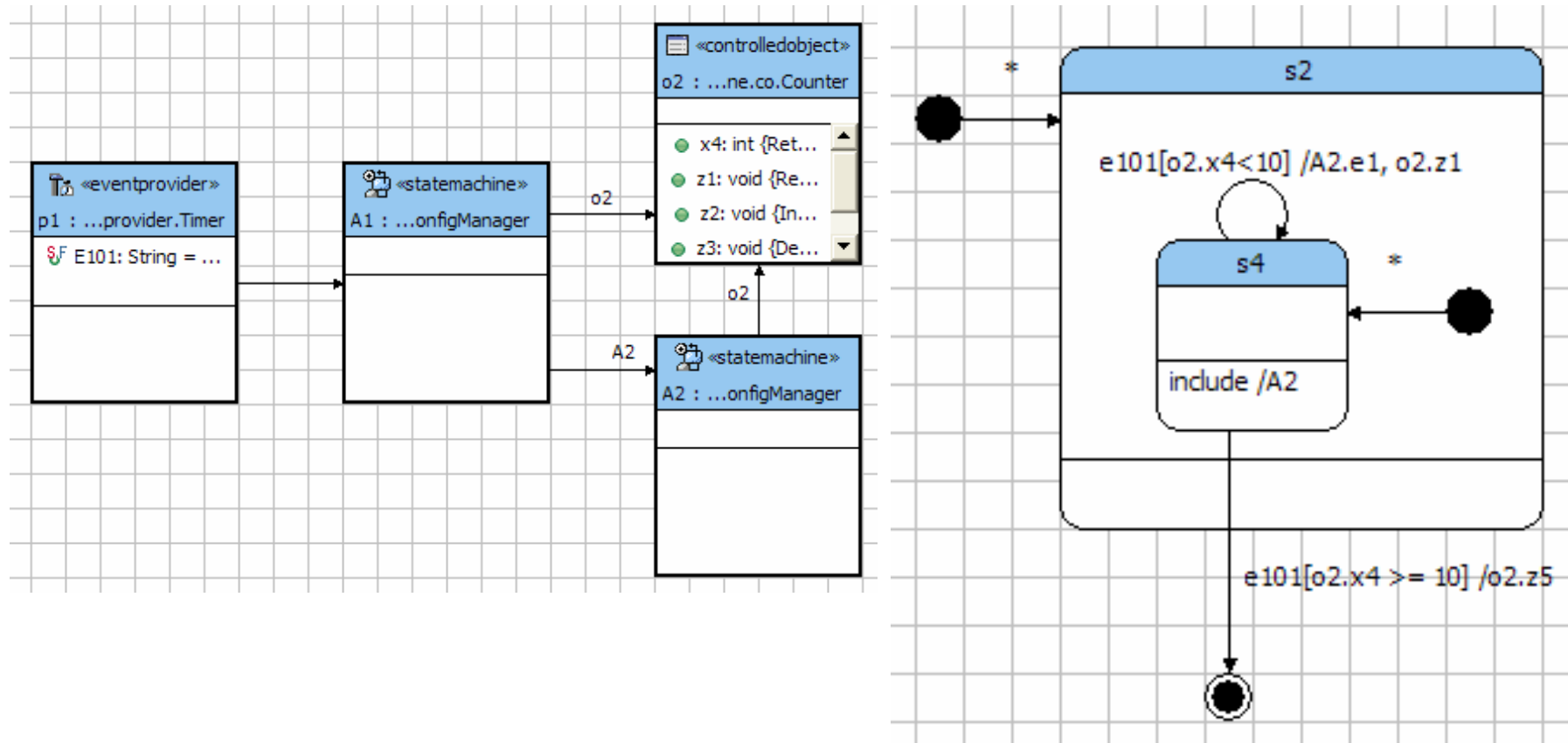


# Пример использования метода (3)





# Синтаксис диаграмм

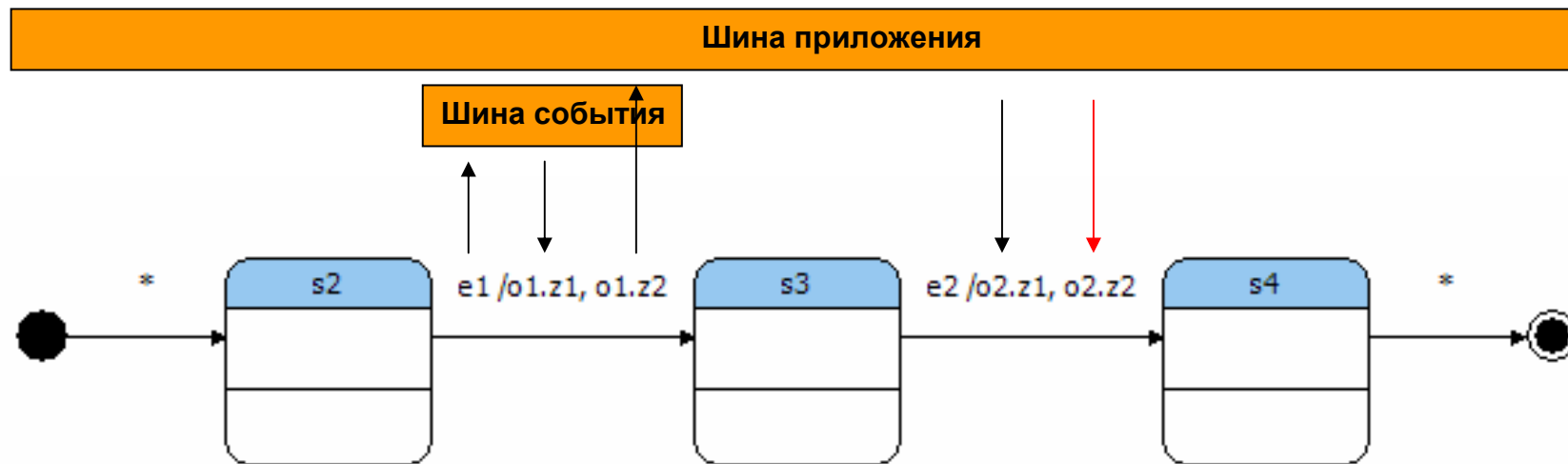


## Операционная семантика модели

- Поставщики событий – активные объекты, обладающие потоками выполнения
- События могут обрабатываться синхронно и асинхронно
- Интерпретационный и компилируемый подходы

# Потоки данных

- Моделируется «поверх» графа переходов
- Проверяется на стадии проектирования
- Реализуется с помощью нетипизированной шины данных
- Три шины данных: события, пользователя, приложения



# Программный пакет UniMod

- Ядро
  - ↳ Мета-модель конечного автомата
  - ↳ Алгоритмы разбора, трансляции и интерпретации условий на переходах
  - ↳ Алгоритмы валидации модели
  - ↳ Интерпретатор XML-описания модели
  - ↳ Генератор кода для целевого языка на основе XML-описания модели
  - ↳ Базовая версия отладчика модели
- Eclipse Plug-in
  - ↳ Графический редактор UML-диаграммы классов и состояний
  - ↳ Интерактивный валидатор модели
  - ↳ Средства запуска и графической отладки модели

# Демонстрация работы редактора

Демонстрация работы редактора

<http://unimod.sourceforge.net/viewlet/animated-demo-rus.html>

# Демонстрация работы отладчика

## Демонстрация работы отладчика

<http://unimod.sourceforge.net/viewlet/debugger-demo-rus.html>

## Область применения (1)

- J2EE приложения (интерпретируемый и компилируемый подходы)
- J2SE приложения (интерпретируемый и компилируемый подходы)
- J2ME приложения (только компилируемый подход)
- Symbian C++ приложения (только компилируемый подход)

## Область применения (2)

- Система автоматического завершения ввода
- Отладчик диаграмм
- Интерпретатор/компилятор блок схем

Property	Value
Event	e9
Guard	! o1.x1 &&
Name	
Output	

(

!

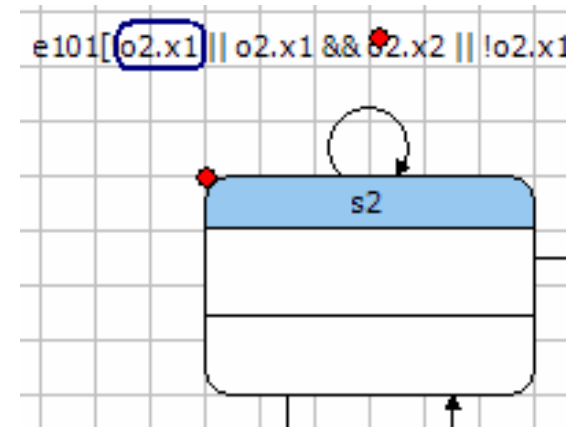
o1

o2

o22

true

false





## Преимущества предлагаемого подхода

- Использует UML-нотацию и, частично, операционную семантику
- Позволяет уменьшить семантический разрыв между фазами проектирования и реализации
- Использует формальные методы проверки модели системы, что позволяет избежать появления многих ошибок на стадии реализации
- Ускоряет процесс разработки и позволяет избежать проблемы синхронизации технической спецификации и кода
- Самодокументируемый, т.к. диаграммы созданные на стадии проектирования являются документацией и программой одновременно



## Общая информация о проекте

- Проект ведется специалистами компании eVelopers Corp. <http://www.evelopers.com/>
- UniMod является проектом с открытым исходным кодом и размещен на сайте SourceForge  
<http://unimod.sourceforge.net/>
- Проект развивается в рамках движения за открытую проектную документацию  
<http://is.ifmo.ru/>



**Спасибо!**