

Разработка метамоделирования “на лету” в системе QReal

Птахина Алина, 445 гр.

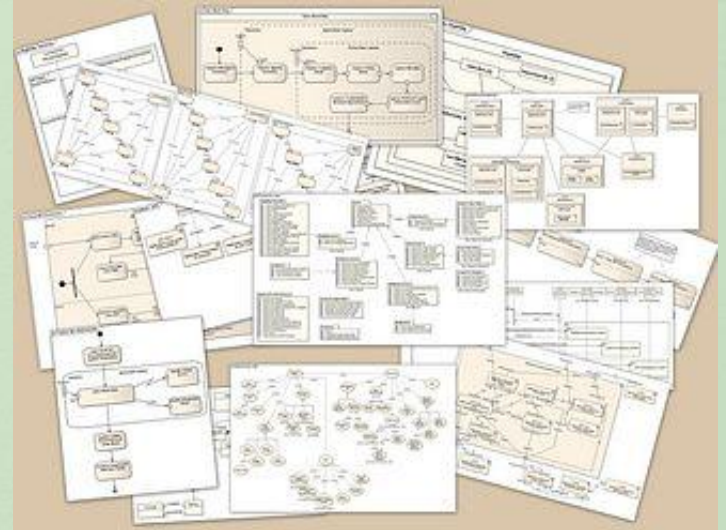
Научный руководитель:
ст. преп. Литвинов Ю.В.

Проблемы

- ⌘ Фиксированные наборы сущностей и диаграмм у большинства визуальных редакторов



- ⌘ Пользователь отдает предпочтение ручке и листу бумаги



Интерпретация и генерация редакторов

- ⌘ DSM-платформы
- ⌘ Мета모델 – модель синтаксиса языка
- ⌘ Способы работы с метамоделью:
 - ⌘ Генерация кода на основе метамоделли
 - ⌘ Интерпретация

Постановка задачи

- ⌘ В режиме интерпретируемой метамодели предоставить пользователю возможность:
 - ⌘ Добавлять/изменять/удалять свойства элементов
 - ⌘ Изменять графическое представление элементов
 - ⌘ Создавать/удалять элементы языка
 - ⌘ Создать свой собственный язык с нуля

Обзор существующих решений

- ⌘ MetaEdit+
- ⌘ MetaLanguage
- ⌘ QReal

MetaEdit+ Domain-Specific Modeling (DSM) environment

MetaEdit+ Workbench
Design your modeling language

The screenshot shows the MetaEdit+ Workbench interface. The main window is titled "Symbol Editor - State [Watch]". It features a toolbar with various symbols and a palette on the left with options like Rectangle, Text, Line, and Image. The central workspace contains a state symbol with the text "State name" and two green dashed boxes labeled "Bli" and "DisplayFn". Below the workspace is a "Property" table and a "Text" field.

Property	Value
X, Y	34, 33
Extent	105 x 40
Condition	
Text	

Below the main window is the "Object Tool State [Watch]" window, which includes a "Property" table and a "Description" field.

Local name	Property name	Data type	Unique?
"State name"	Name [Watch]	String	F
DisplayFn	DisplayFnRef	DisplayFn	F
Blinking	Time unit	String (Overridable L.F)	

Description:
A State is an intermediate state in state automaton that defines a logical watch application. State has certain persistency and certain activities that are carried out while the state is active.

MetaEdit+ Modeler
Use your modeling language

The screenshot shows the MetaEdit+ Modeler interface. The main window is titled "WatchApplication: Stopwatch 2. mansakuuta 2000, 13:20". It displays a state transition diagram for a stopwatch application. The diagram includes states "Stopped" and "Running", transitions labeled "Down" and "Up", and a "Mode" transition. The "Stopped" state has a "stopTime" property, and the "Running" state has a "running" property. The diagram also shows a "sysTime" property and a "startTime" property. A red arrow points from the "Object Tool State [Watch]" window in the previous screenshot to the "Stopped" state in this diagram.

Below the main window is a "Property" table and a "Text" field.

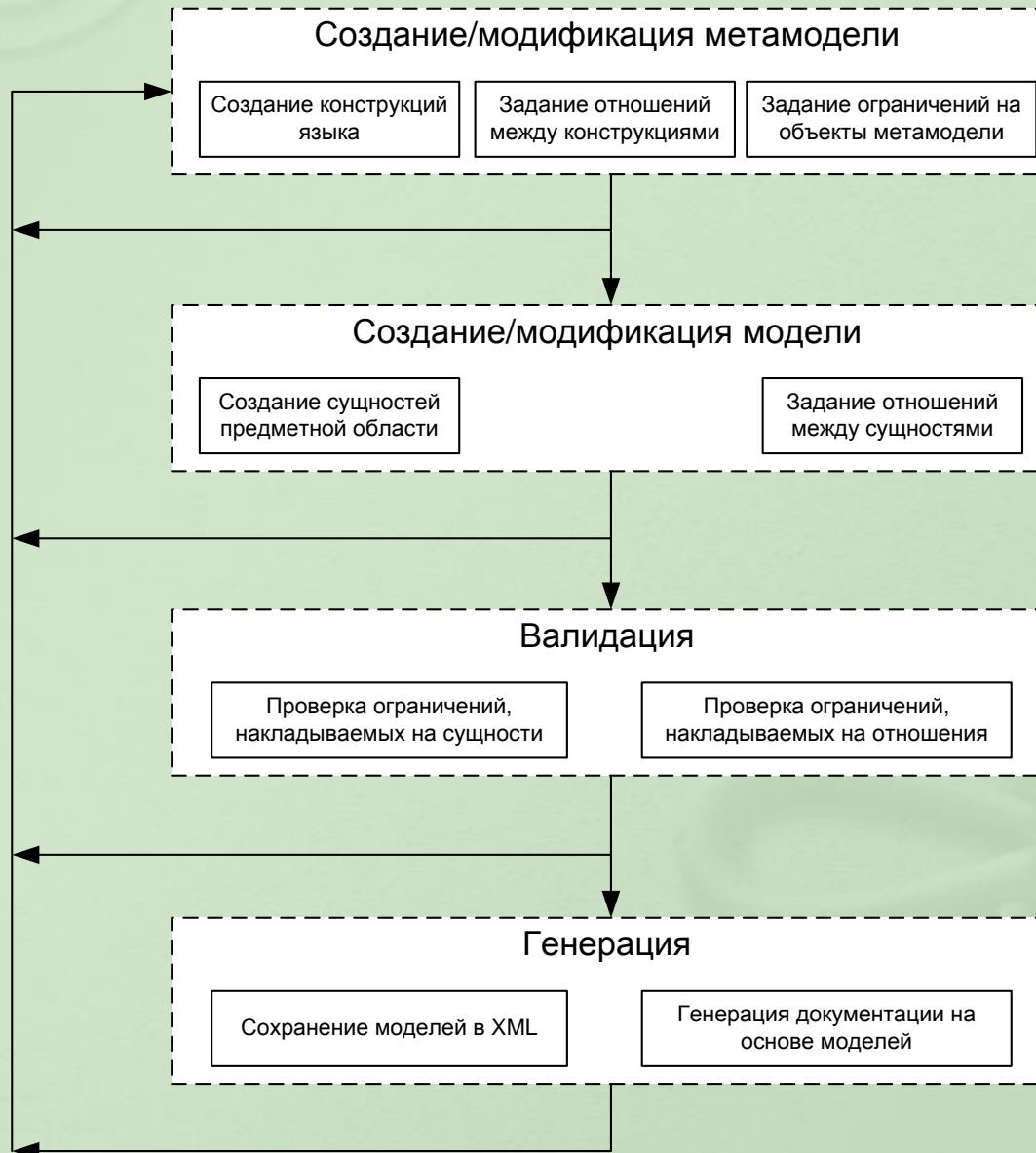
Property	Value
stopTime	?? : SEC : ??
sysTime	?? : SEC : ??
startTime	?? : SEC : ??

Below the main window is the "Object Tool State [Watch]" window, which includes a "Property" table and a "Description" field.

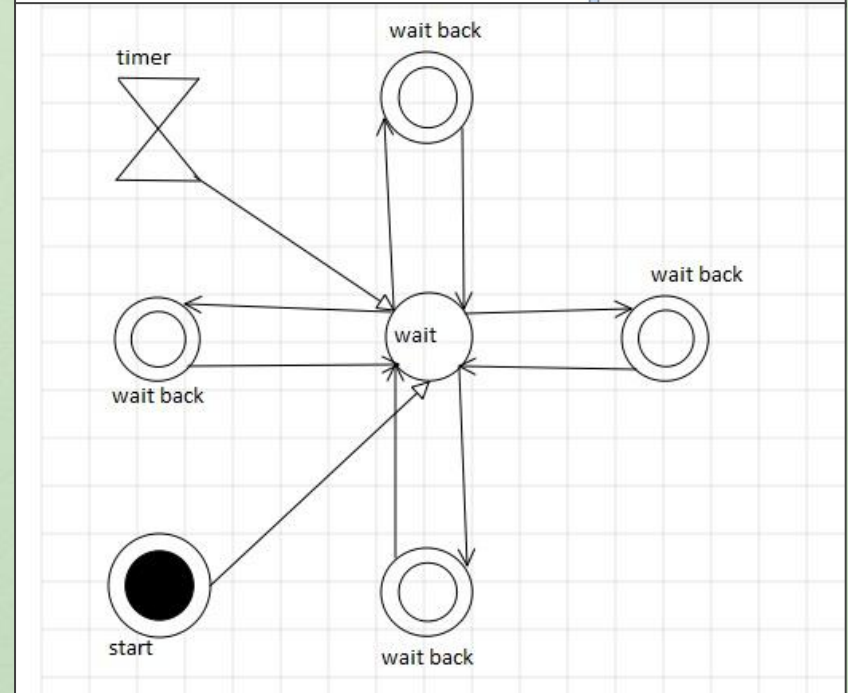
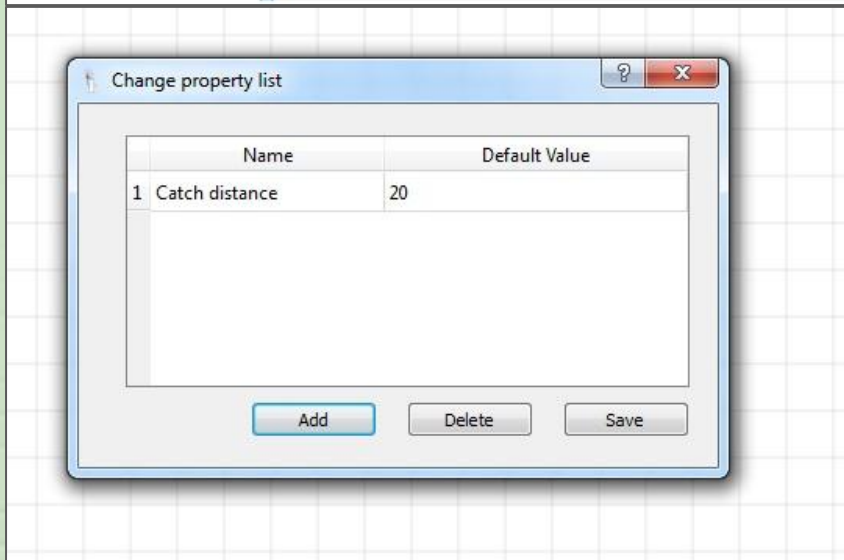
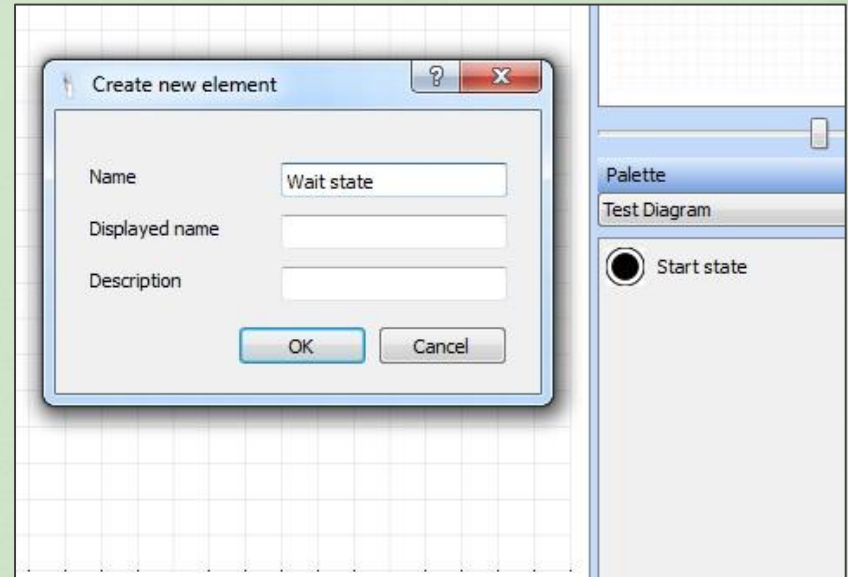
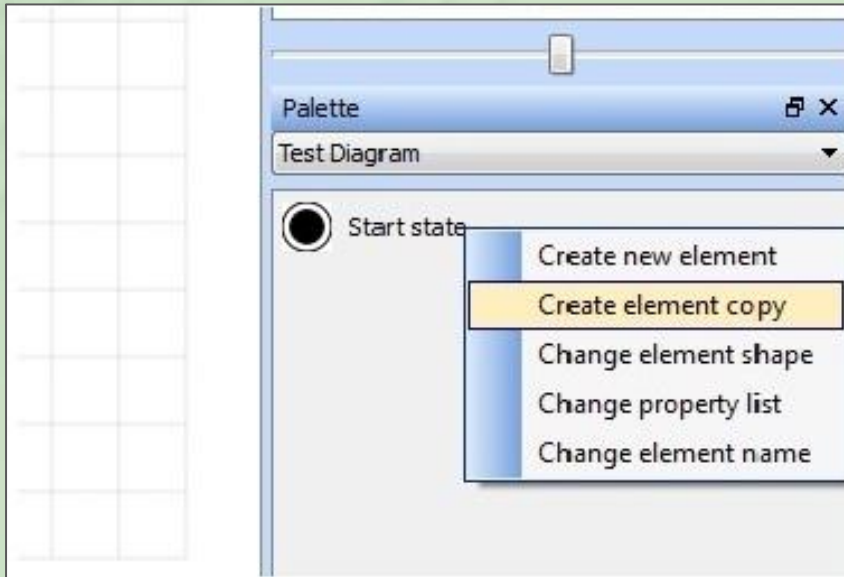
Local name	Property name	Data type	Unique?
"State name"	Name [Watch]	String	F
DisplayFn	DisplayFnRef	DisplayFn	F
Blinking	Time unit	String (Overridable L.F)	

Description:
In this state the Stopwatch is stopped and current stop time is shown on the display.

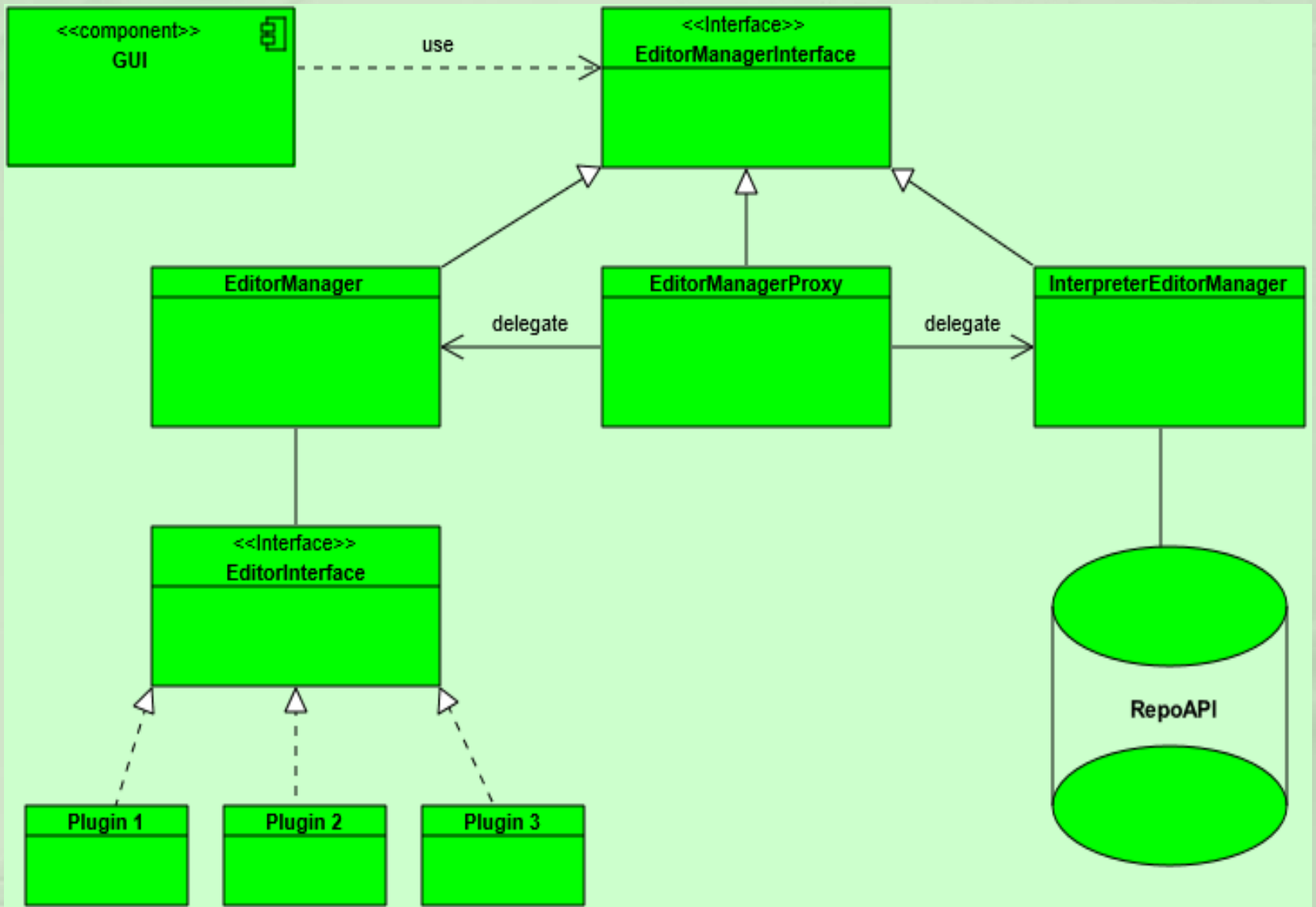
MetaLanguage



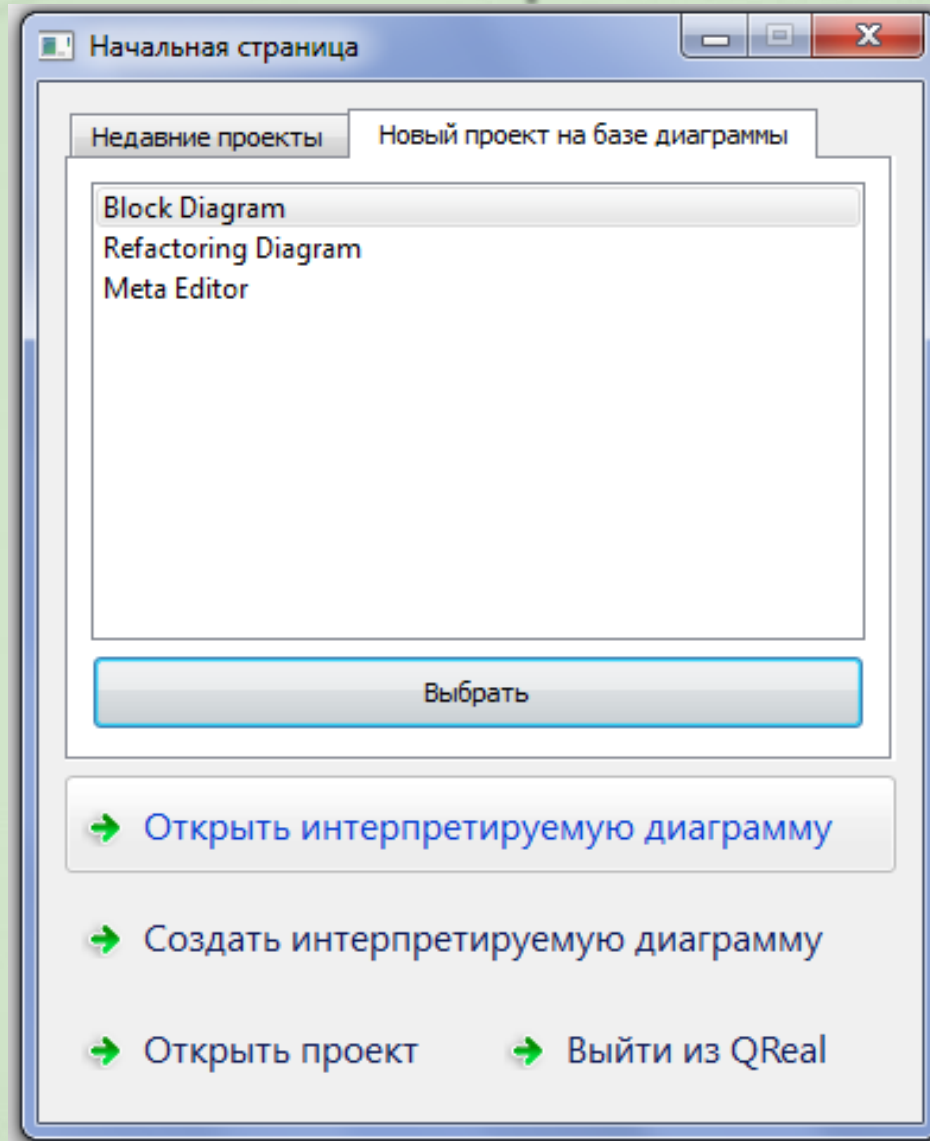
QReal



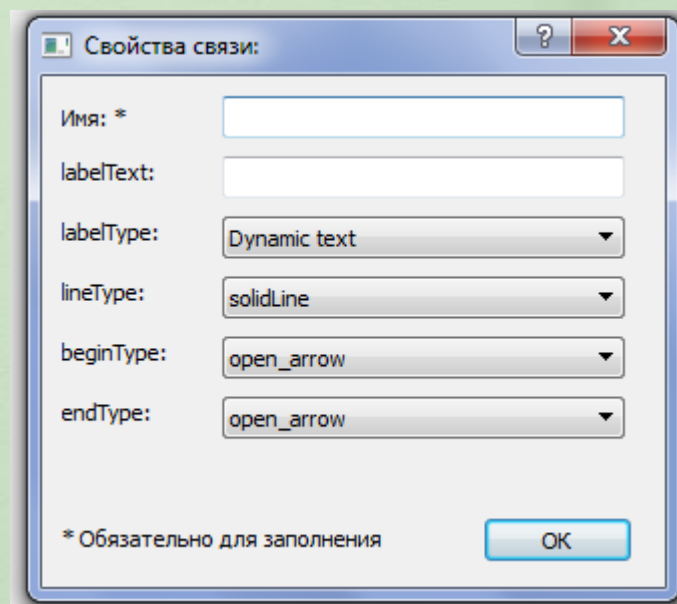
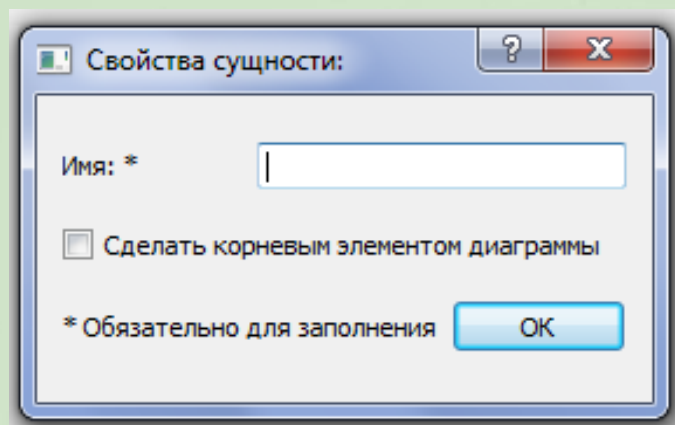
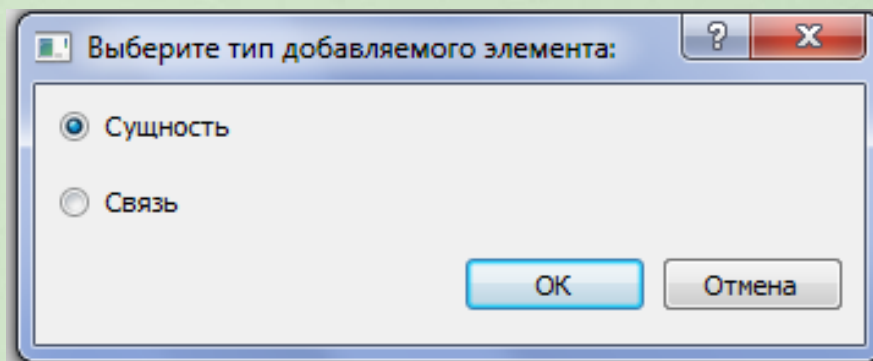
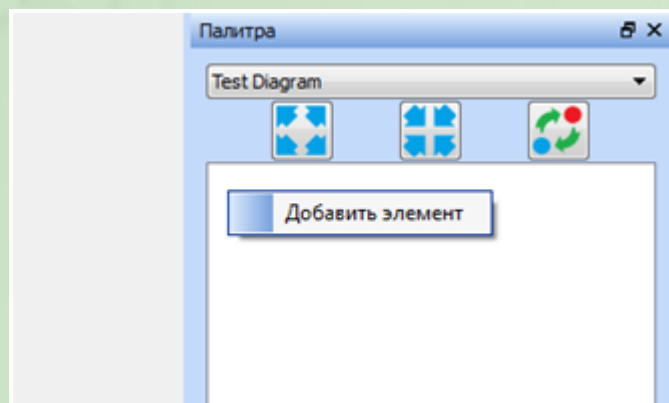
Архитектура



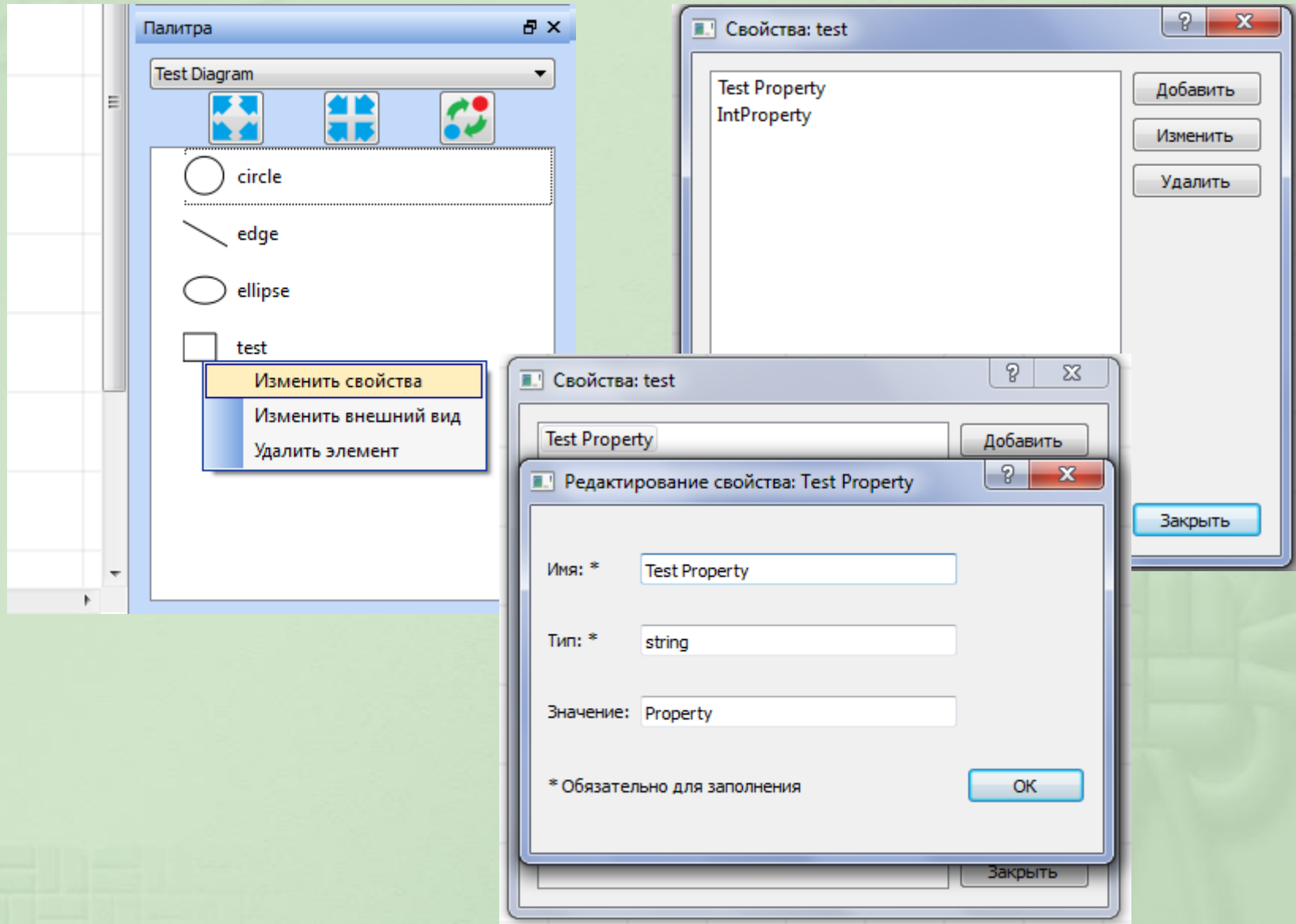
Начало работы



Добавление нового элемента



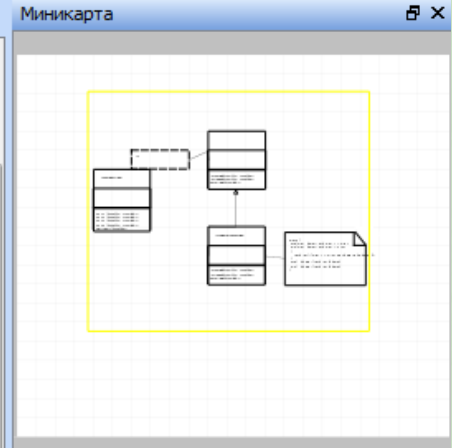
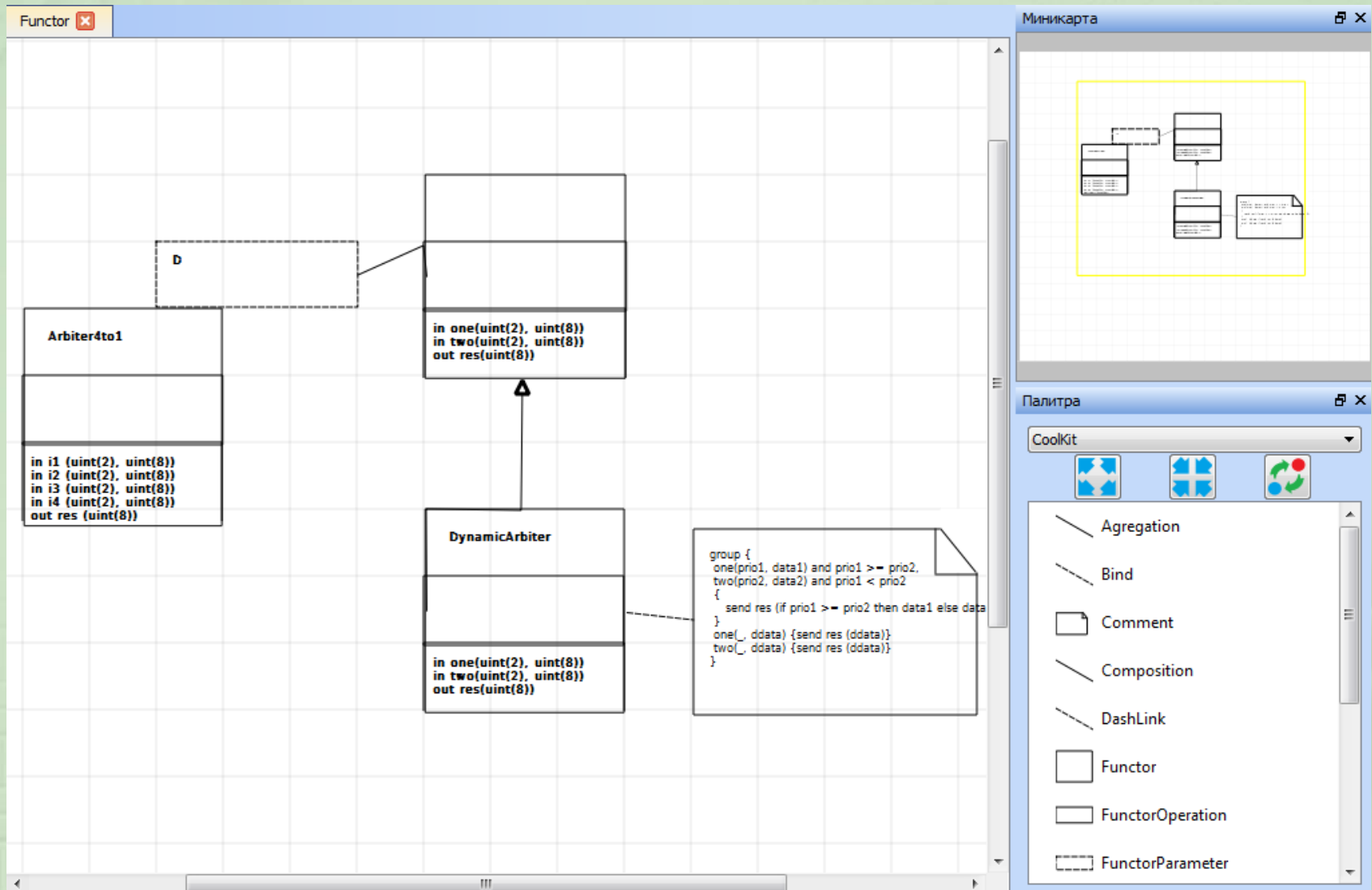
Изменение свойств элемента



Апробация (1)

- ⌘ Задача об арбитраже динамических приоритетов 4 в 1
 - ⌘ На один из четырех входов поступают данные, первый параметр пришедших данных – приоритет. Если в одном такте данные поступили на несколько входов, на выход выдаются данные с наибольшим приоритетом, остальные входы объявляются неготовыми. Если приходит только одно сообщение, оно отправляется на выход.

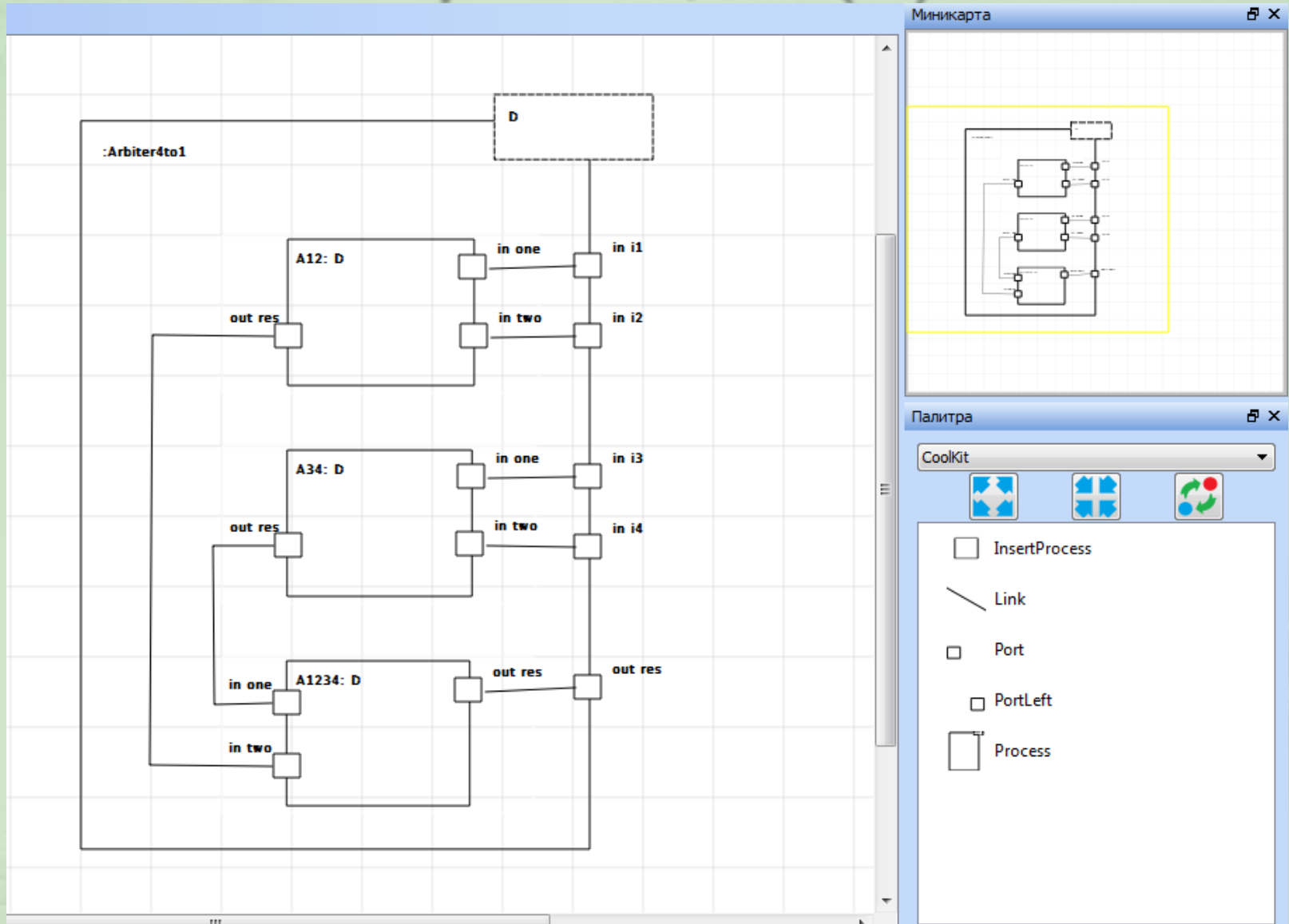
Апробация (2)



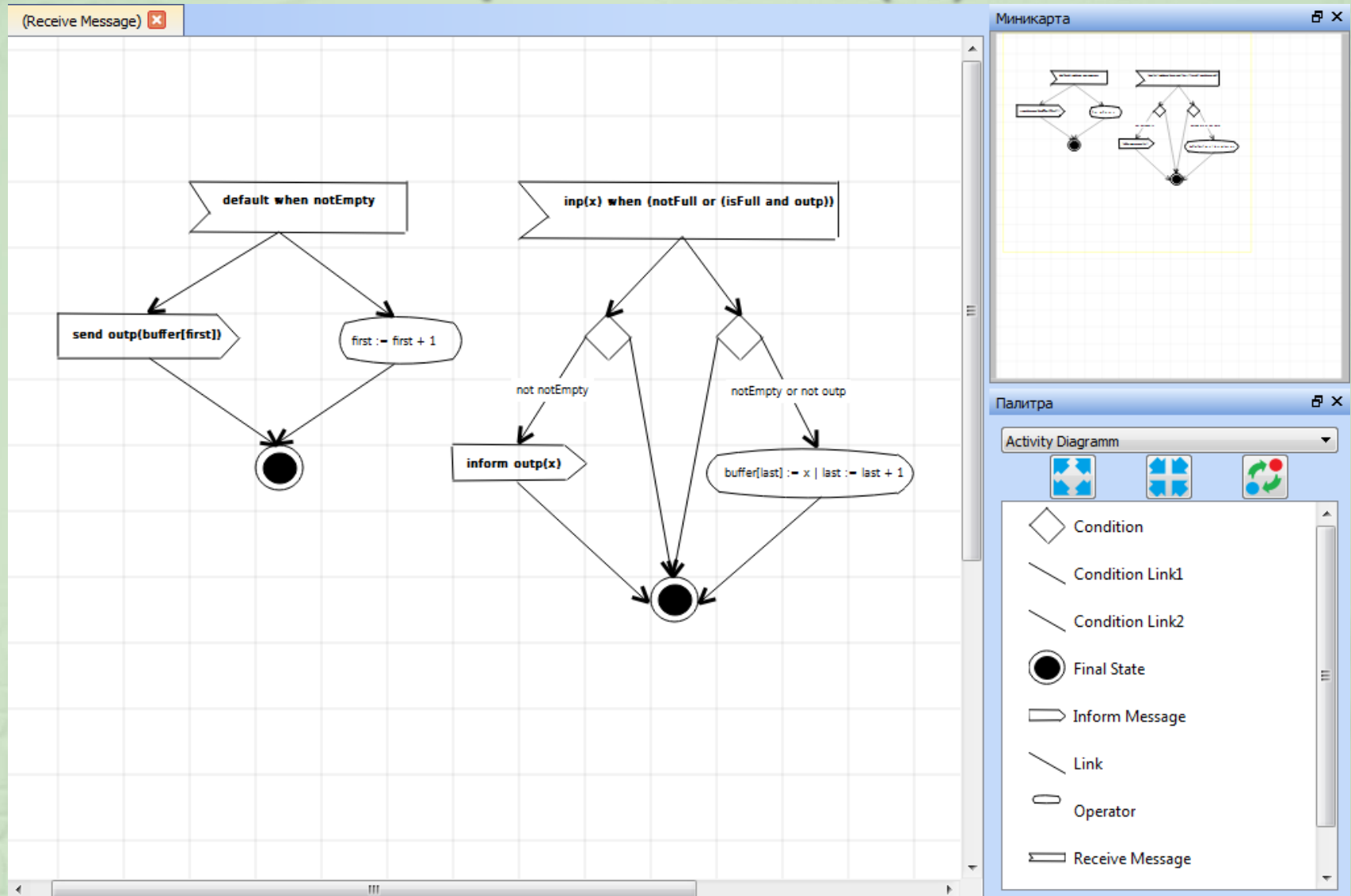
Палитра (Palette) for CoolKit, listing various UML symbols and their functions:

- Aggregation
- Bind
- Comment
- Composition
- DashLink
- Functor
- FunctorOperation
- FunctorParameter

Апробация (3)



Апробация (4)



Результаты

- ⌘ В режиме интерпретируемой метамодели реализовано:
 - ⌘ Добавление, удаление и изменение свойства элемента
 - ⌘ Изменение графического изображения элемента
 - ⌘ Добавление нового и удаление существующего элемента языка
 - ⌘ Возможность создания собственного языка с нуля
- ⌘ Проведена апробация на реальной задаче