

Графическая технология управления “умной теплицей”

Кузьмина Е.В., 471 группа

Научный руководитель: к.т.н., доцент кафедры системного
программирования Литвинов Ю.В.

Рецензент: Перешеина А.О.

Введение

- Концепция интернета вещей
- Что такое “умная теплица”
- Определение правил с помощью простого для понимания визуального языка программирования

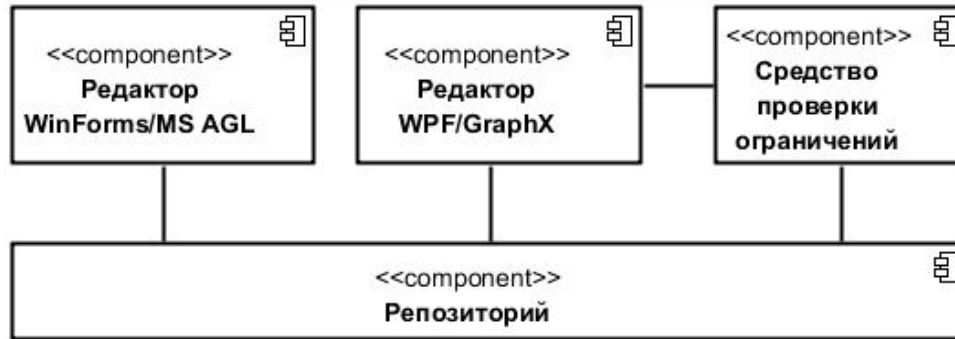


Существующие решения

- HortiMaX-Go!
- OpenHAB
- Node-RED
- Samsung ARTIK

REAL.NET

- DSM*-платформа
- На платформе .NET
- Репозиторий содержит метамодель языка



* Domain Specific modeling

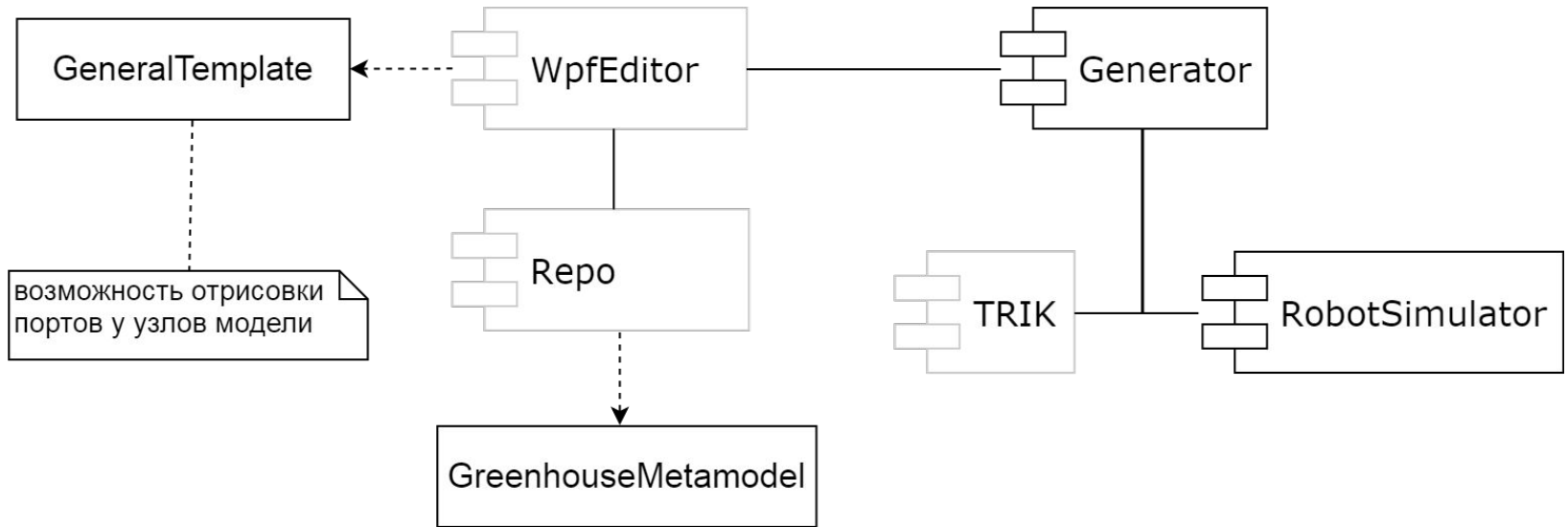
Цель

Целью работы является создание на основе платформы REAL.NET прототипа системы автоматического управления теплицей, включающего в себя предметно-ориентированный визуальный язык программирования для задания пользователем сценариев работы приборов

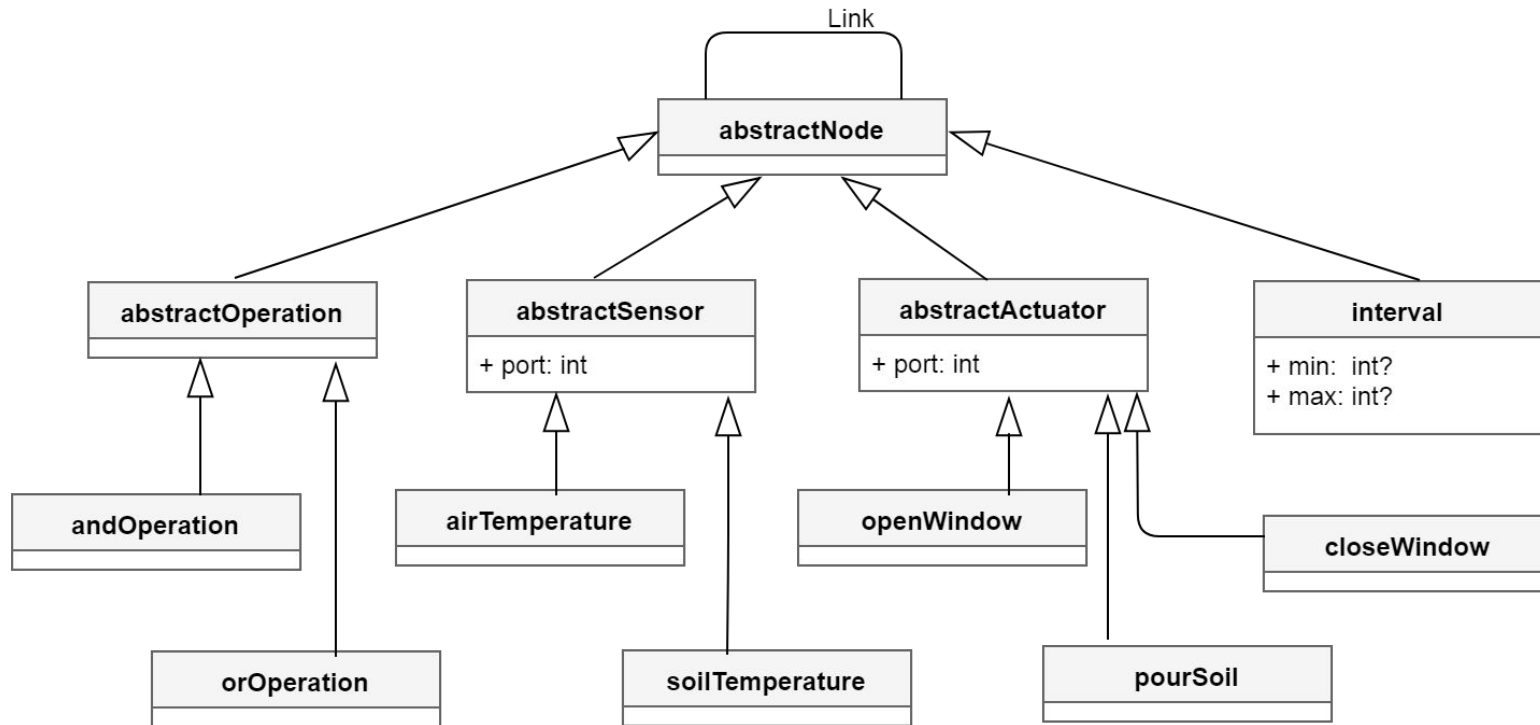
Задачи

- Выполнить проектирование целевой системы и входящего в нее предметно-ориентированного визуального языка
- Реализовать редактор для моделирования пользователями сценариев работы теплицы
- Реализовать генератор для преобразования графической модели в сценарий на текстовом языке
- Провести апробацию прототипа

Архитектура

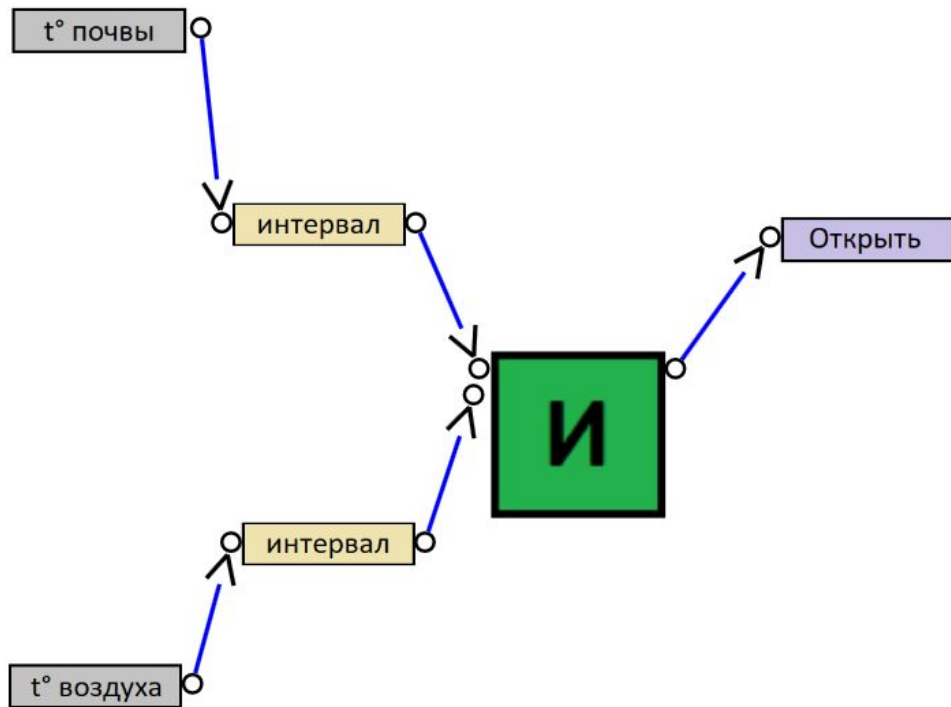


Метамоделъ языка “умной теплицы”



Редактор

- На основе WPF/GraphX редактора платформы REAL.NET, включающего следующие компоненты:
 - сцена с моделью
 - палитра с элементами метамодели
 - редактор свойств
 - список всех элементов
 - окно для вывода ошибок
- Отрисовка портов у узлов модели
 - у сенсоров только для исходящих ребер/данных
 - у актуаторов — только для входящих ребер/данных



Link

- Interval
- SoilTemperature
- AirTemperature
- PourSoil
- CloseWindow
- OpenWindow
- ИЛИ OrOperation
- И AndOperation

Elements:

- aOpenWindow
- И aAndOperation
- aInterval
- aInterval

Selected entity:

Name	Type	Value
max	Int	20
min	Int	10
instanceMetatype	String	Metatype
isAbstract	Boolean	false
shape	String	View/Pic

Constraints

Execute

StopGener

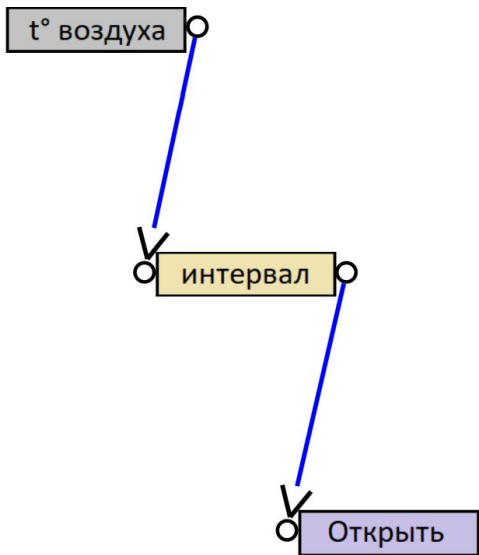
Generate

Sample add-on successfully launched

Messages Errors

Генератор

- Текстовые шаблоны Text Template Transformation Toolkit (T4)
- Библиотека RX.NET
- Рассмотрение ребер моделей как потоков данных
 - актуаторы — Observer
 - сенсоры — Observable
 - все остальные элементы — Subject



```
1. element0 = new Actuator(0);
2. IObservable<int> reactElement0 = Observer.Create<int>(x => element0.Action(x));
3.
4. element1 = new Interval(1);
5. element1.Min = 20;
6. element1.Max = null;
7. IObservable<int> observable1 = Observable.FromEventPattern<int>(
8.     h => element1.Event += h,
9.     h => element1.Event -= h)
10.    .Select(e => e.EventArgs)
11.    .Synchronize().DistinctUntilChanged();
12. IObservable<int> observable2 = Observable.FromEventPattern<int>(
13.     h => element2.Event += h,
14.     h => element2.Event -= h)
15.    .Select(e => e.EventArgs)
16.    .Synchronize().DistinctUntilChanged();
17. IObservable<int> reactElement2 = Subject.Create<int>(observable1, observable2);
18.
19. element2 = new Sensor(2);
20. IObservable<int> reactElement2 = Observable.FromEventPattern<int>(
    h => element2.Event += h,
    h => element2.Event -= h)
    .Select(e => e.EventArgs)
    .Synchronize().DistinctUntilChanged();
```

Апробация

Использование
прототипа
представителем
целевой аудитории

The screenshot displays a software development environment with three main components:

- Terminal Window (Top Left):** Shows a sequence of log messages:

```
Sensor at Port 1, new value : 24
Actuator at Port 5 has acted!

Sensor at Port 1, new value : 6
Actuator at Port 4 has acted!

Sensor at Port 1, new value : 17

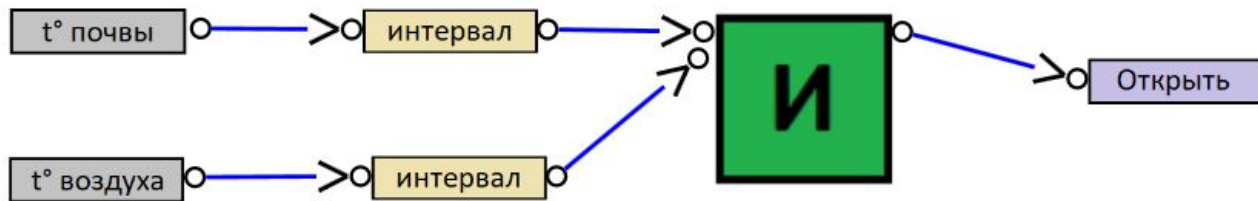
Sensor at Port 1, new value : 24
Actuator at Port 5 has acted!

Sensor at Port 1, new value : 22
```
- Graphical Diagram (Center):** A state transition diagram with the following elements:
 - Initial State:** A box labeled `t° воздуха` with the identifier `aAirTemperature` below it.
 - Transitions:** Two blue arrows originate from the initial state. The upper arrow points to a yellow box labeled `интервал` with the identifier `aInterval` below it. The lower arrow points to another yellow box labeled `интервал` with the identifier `aInterval` below it.
 - Final States:** From the upper `интервал` box, an arrow points to a purple box labeled `Открыть` with the identifier `aOpenWindow` below it. From the lower `интервал` box, an arrow points to a purple box labeled `Закреть` with the identifier `aCloseWindow` below it.
- Component Palette (Right):** A vertical list of components with a 'Link' button on the left of each item:
 - Interval
 - SoilTemperature
 - AirTemperature
 - PourSoil
 - CloseWindow
 - OpenWindow
 - ИЛИ OrOperation
 - И AndOperationBelow this list is an 'Elements:' section containing:
 - aAirTemperature
 - aInterval
 - aInterval
 - aCloseWindowThe 'Selected entity:' section shows:
 - Value: null
 - Value: 20
 - Metatype.Node: false
 - View/Pictures/Greenhouse/interval.pngAt the bottom of the palette are buttons for 'Constraints', 'Execute', 'StopGener', and 'Generate'.

At the bottom of the interface, a status bar shows the message 'Sample add-on successfully launched' and tabs for 'Messages' and 'Errors'.

Апробация

- Микроконтроллере TRIK и библиотека Trik-Sharp
- Клиент WinSCP для загрузки сценария
- Клиент PuTTY для запуска сценария



Результаты

- Спроектирована архитектура прототипа графической технологии управления "умной теплицей" и нового визуального языка
- Реализован редактор на основе редактора системы REAL.NET для построения пользователем модели сценария работы теплицы
- Реализован генератор, преобразующий созданную модель в сценарий на языке C#
- Проведена апробация прототипа на пользователе, не обладающем навыками программирования