



Веб-служба на базе Workflow Foundation

Павел Цытович

ptsytovich@luxoft.com

<http://facebook.com/pavel.tsytovich>

<http://vk.com/pavel.tsytovich>

Luxoft Training

Luxoft Training – лидер в области обучения дисциплинам Software Engineering

Обучение на базе Luxoft Training – это возможность карьерного и профессионального роста, повышения квалификации и усовершенствования существующих умений знаний и навыков для каждого специалиста по программной инженерии.

Формат обучения:

- ❖ Открытые очные модульные курсы и вебинары
- ❖ Корпоративные программы
- ❖ Тренинговые программы (классы подготовки специалистов)
- ❖ Мастер-классы экспертов с мировым именем

План работы

- ❑ Domain Specific Language – что, где, когда
- ❑ Workflow Foundation как способ реализации DSL
- ❑ WCF служба на базе Workflow Foundation
- ❑ Организация продолжительных во времени процессов Workflow Foundation



Domain Specific Language

Domain Specific Language



DSL – язык программирования, основанный на описании абстракций конкретной предметной области, специально разработанный для решения задач в этой конкретной предметной области.

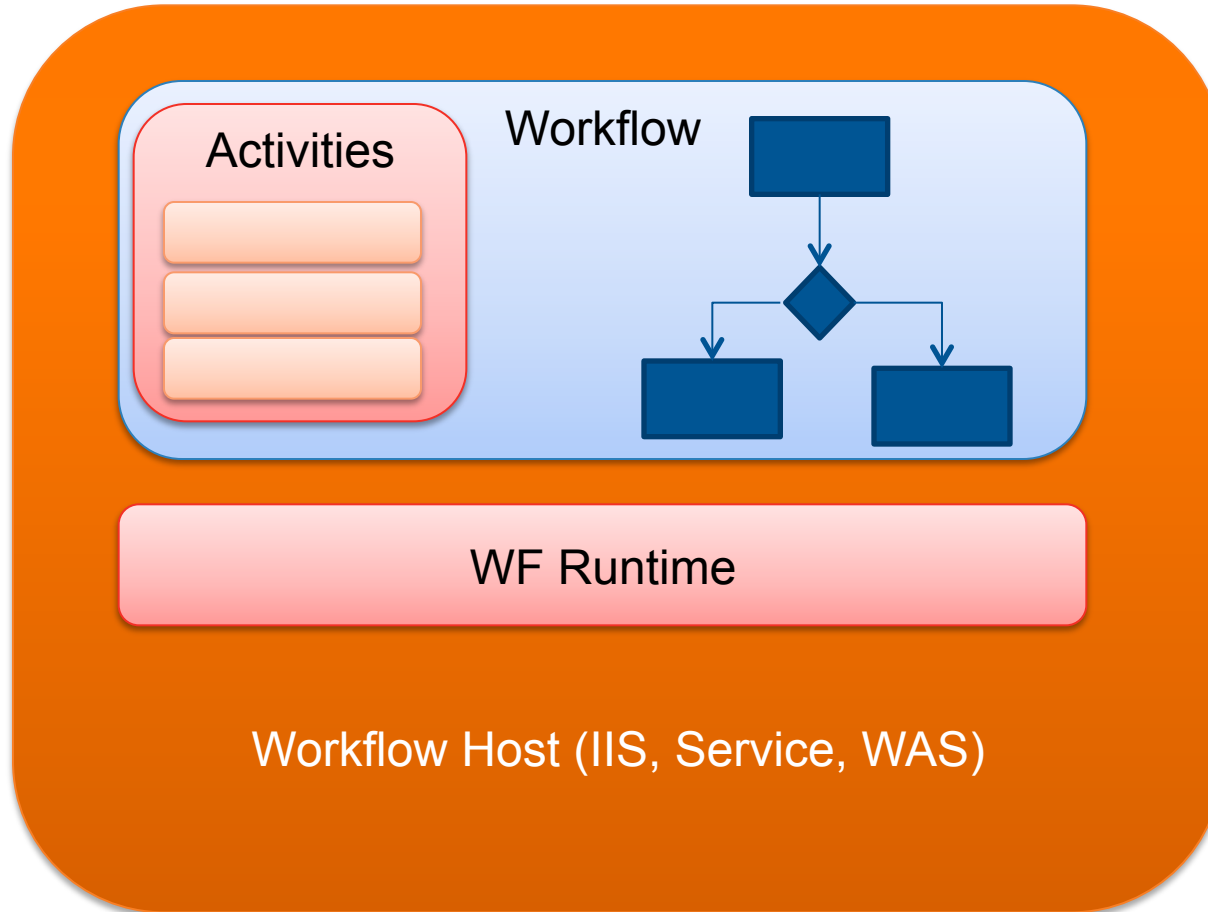
Платформы реализации DSL

- ❑ Metaprograming System (Java)
- ❑ BPEL (Java)
- ❑ DSL Tools (.NET Framework, Visual Studio)
- ❑ Microsoft Workflow Foundation (.NET Framework)



Workflow Foundation как способ реализации DSL

Workflow Foundation



Примитивы Workflow

- ☐ Sequence
- ☐ If
- ☐ Do While / While
- ☐ ForEach<T> / ParallelForEach<T>
- ☐ Pick
- ☐ Switch<T>

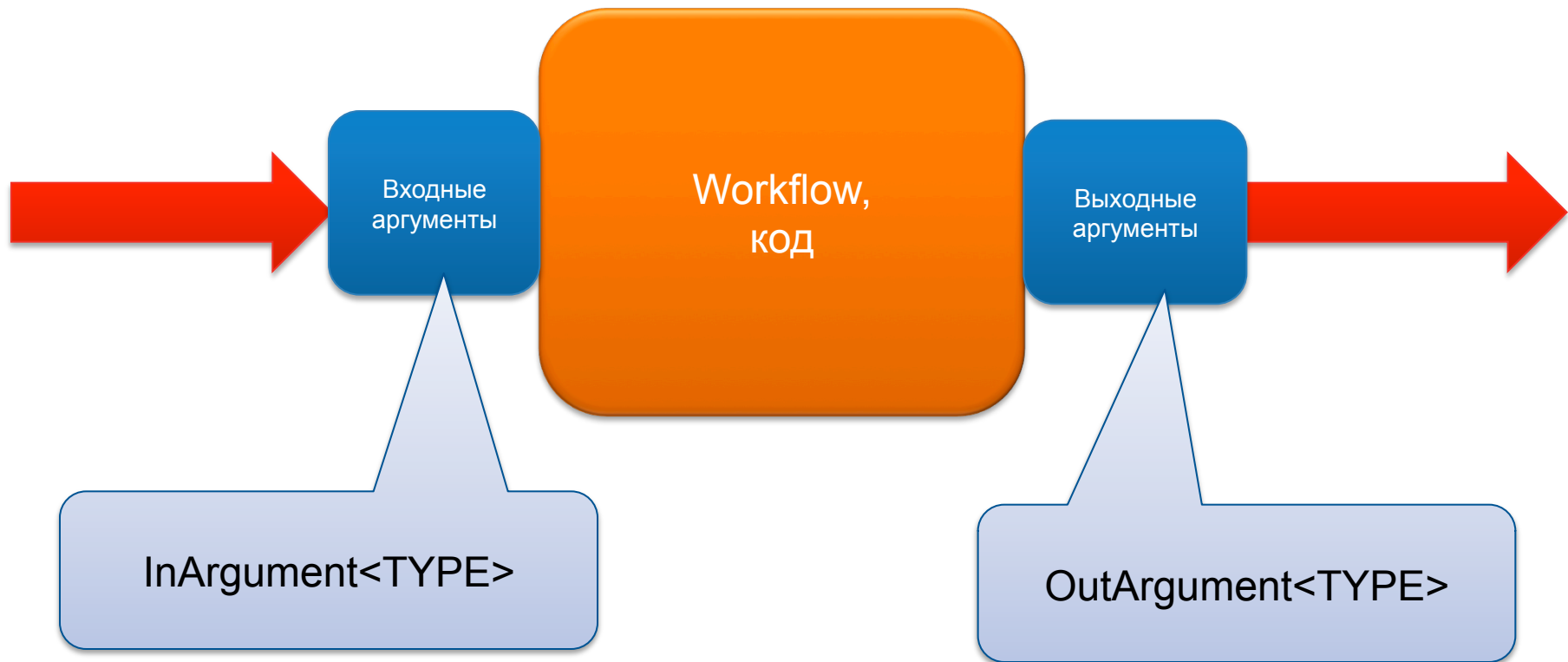
Flowchart – реинкарнация блок-схем

- ☐ Flowchart
- ☐ FlowDecision
- ☐ FlowSwitch<T>

Проблема реализации DSL на основе Windows Workflow Foundation

- ❑ Workflow Foundation содержит лишь набор примитивов для описания рабочих процессов, не пригодных для использования аналитиками непосредственно
- ❑ Разработчик должен подготовить свой набор custom activities для описания бизнес-процессов в терминах предметной области
- ❑ Разработчик должен предоставить внешний дизайнер Workflow Foundation, если рабочие процессы будут разрабатываться самим пользователем

Custom code activities



Шаблон code activity - Синхронный

```
namespace TravelCoordinator.Activities
{
    public sealed class MailSend : CodeActivity
    {
        // Определите входной аргумент действия типа string
        public InArgument<string> Text { get; set; }
        public InArgument<string> EmailFrom { get; set; }
        public InArgument<string> EmailTo { get; set; }
        public InArgument<string> Subject { get; set; }

        // Если действие возвращает значение, создайте класс, производный от CodeActivity<TResult>
        // и верните значение из метода Execute.
        protected override void Execute(CodeActivityContext context)
        {
            // Получите значение входного аргумента Text во время выполнения
            string text = context.GetValue(this.Text);
        }
    }
}
```

Шаблон code Activity – асинхронный

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Activities;
using TravelCoordinator.Entities;
using System.Net;
using System.IO;
using System.Xml;

namespace TravelCoordinator.Activities
{
    public sealed class HotelBookingService : AsyncCodeActivity
    {
        public InArgument<string> Country { get; set; }
        public InArgument<TravelRequest> TravelRequest { get; set; }
        public OutArgument<Hotel> BookedHotel { get; set; }
        protected override IAsyncResult BeginExecute(AsyncCodeActivityContext context,
            AsyncCallback callback, object state)
        {
            HttpWebRequest request = (HttpWebRequest)WebRequest.
                Create("http://distribution-xml.booking.com/...");
            request.Credentials = new NetworkCredential("pavlin-co@yandex.ru", "****");
            context.UserState = request;
            return request.BeginGetResponse(callback, state);
        }

        protected override void EndExecute(AsyncCodeActivityContext context, IAsyncResult result)
        {
        }
    }
}
```

Создание собственного внешнего вида Custom Activity

MailSend

From
"ptsytovich@luxoft.com"

To
"pavel.tsytovich@gmail.com"

Subject
"test workflow designer"

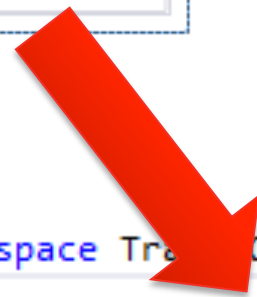
Body
"this is body"



XAML



CS



```
namespace TestCoordinator.Activities
{
    [Designer(typeof(SendMailDesigner))]
    public sealed class MailSend : CodeActivity
    {
        // Определите входной аргумент действия тиг
```



WCF служба на базе Workflow Foundation

Прием и передача сообщений

 Receive From Finance dept

OperationName

Content [Просмотреть сообщение](#)

▽

 SendReplyToReceive

Request

Content [Просмотреть сообщение](#)

▽

- ☐ Content
- ☐ DisplayName
- ☐ OperationName
- ☐ ServiceContractName

- ☐ Content
- ☐ DisplayName
- ☐ Request

Хостирование WCF с WF

☐ IIS/WAS

☐ WorkflowServiceHost



Организация продолжительных во времени процессов Workflow Foundation

Что необходимо сделать для поддержки длительных бизнес-процессов?

- ☐ Обеспечить сохранение состояния сеанса
- ☐ Обеспечить загрузку нужно состояния сеанса

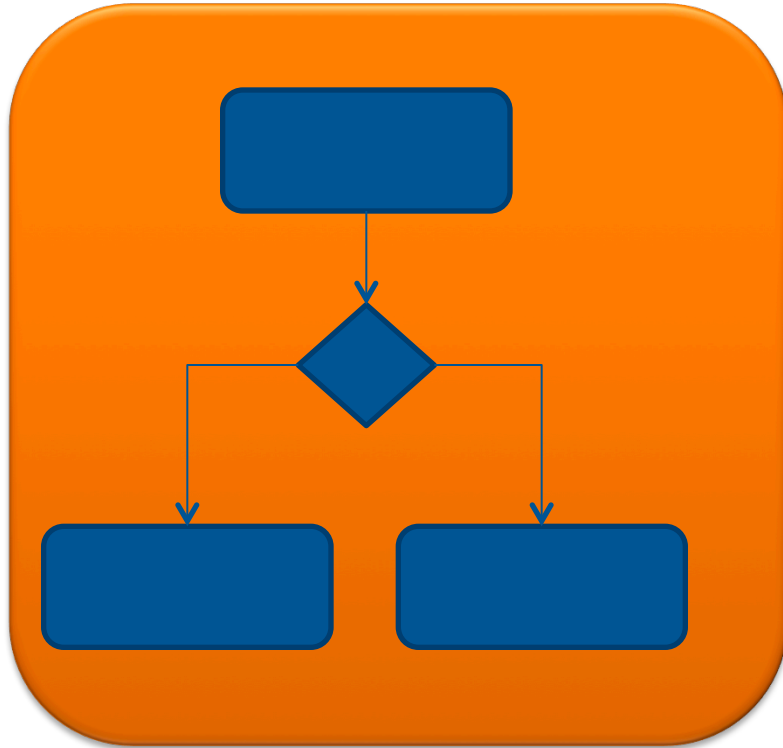
Сохранение сеанса в MS SQL

```
<serviceBehaviors>
  <behavior>
    <!--Чтобы избежать раскрытия метаданных, до развертывания задайте следующему параметру значение "false". -->
    <serviceMetadata httpGetEnabled="true"/>
    <!--Чтобы при сбое получать подробные сведения об исключении для отладки, задайте следующему параметру значение "true". Чтобы информация об исключении не выдавалась, за-
    <serviceDebug includeExceptionDetailInFaults="true"/>
    <sqlWorkflowInstanceStore
      connectionString="Data Source=PIETER\SQLEXPRESS;Initial Catalog=TravelCoordinator;Integrated Security=True;Connect Timeout=15;Encrypt=False;TrustServerCertificate=False"
      instanceCompletionAction="DeleteAll"/>
  </behavior>
</serviceBehaviors>
```

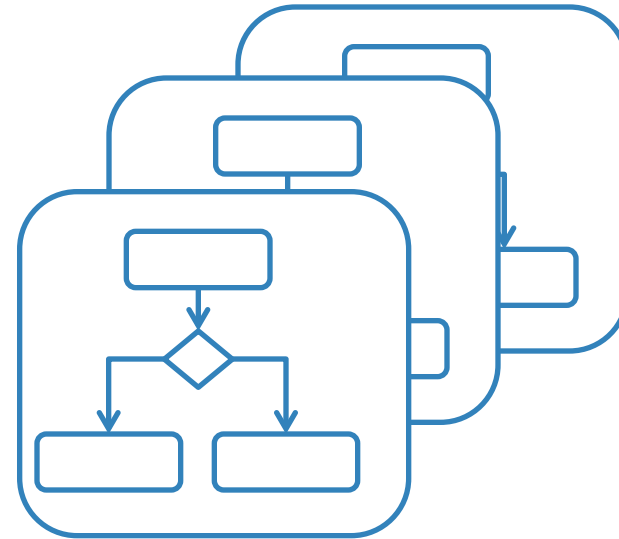
Activity, обеспечивающие сохранение состояния

- ☐ Persist
- ☐ Delay
- ☐ SendReply //свойство PersistBeforeSend

Сопоставление рабочих процессов



Workflow



WorkflowInstance



Корреляция



Корреляция

- ❑ Способ группировки сообщений. Классическим примером являются сессии в WCF или более часто используемый механизм обмена сообщениями – запрос/ответ.
- ❑ Способ сопоставления сообщений (данных) с определенным экземпляром службы (workflow). Например, можно создать соотношение между SessionId и InstanceId.
- ❑ Выполняется с помощью свойств Receive:
 - ❑ CorrelatesOn
 - ❑ CorrelatesWith
- ❑ Используется специальный тип переменной CorrelationHandler

Виды корреляций

- ☐ На основе механизма запрос/ответ
- ☐ Корреляция на основе контекста
- ☐ Корреляция на основе бизнес-логики

Корреляция на основе бизнес-логики

Content Definition

Сообщение

Данные сообщения: data

Тип сообщений: TravelCoordinator.Entities.TravelRequest

Параметры

OK Отмена

Определение CorrelatesOn

CorrelatesWith: *Дескриптор корреляции*

Запросы XPath

Ключ	Запрос
key1	sm:body()/xg0:TravelRequest/xg0:Id

Дважды щелкните свойство

Выберите свойство для создания XPath

OK Отмена



Поддержка транзакций



Действия для транзакций

- ☐ TransactionScope
- ☐ TransactedReceiveScope
- ☐ CompensableActivity
- ☐ Confirm
- ☐ Compensate

Где почитать о Windows Workflow Foundation

- ❑ Andrew Zhu. Microsoft Windows Workflow Foundation 4.0 Cookbook
- ❑ <http://alexey.raga.name/post/2010/06/06/workflow-foundation-4-flowcharts.aspx>
- ❑ <http://mschnlnine.vo.llnwd.net/d1/pdc08/WMV/TL06.wmv>
- ❑ <http://www.codeproject.com/Articles/38202/Windows-Workflow-WF-as-a-WCF-Service>



Благодарю за внимание!



Вопросы?