Laboratórios de Desenvolvimento de Software - LEI

Windows Communication Foundation (WCF)

Ricardo Santos Nuno Macedo Ângelo Costa
nfm@estgf.ipp.pt

November 13, 2013

Service-oriented Architecture

- *Service-oriented Architecture* (SOA) is a software design pattern comprised of discrete application functionalities;
- These functionalities are exposed to the world as *services*;
- The goal is to provide seamless collaboration and code reuse.

Service-oriented Architecture

- A *service* is a piece of code that provides a functionality defined in a *contract*;
- This simplicity allows service reuse and combination, resulting in more complex services;
- The logic of the implementation of the service is hidden from the clients;
- Services must agree on the communications protocol and data format.

Web Services

- *Web services* may be used to achieve a SOA;
- Provide functionalities over standard Internet protocols, independent of the underlying infrastructure.

Windows Communication Foundation

- *Windows Communication Foundation* (WCF) is a unified communication layer for .NET applications.
- Enables technology-agnostic communication;
- Web services are a particular application of WCF.

Windows Communication Foundation

- Exposed operations are defined through *contracts*;
- Contracts specify the information available to the clients:
  - Are an extended interface;
- The implementation is hidden from the client.

Contracts

- *Service contract*: details the available operations;
- *Data contract*: details the structure of the data passed;
- *Message contract*: details the structure of the messages passed.

Contracts

- Contracts are defined by *attributes*;
- Recall: attributes provide extra information to the compiler regarding program items.

Service Contract

- `OperationContract` attribute defines exposed operations;

```
[ServiceContract (Namespace="unique_namespace")]
public interface ICompanyService {
  [OperationContract]
  int AddProduct (ProductData product);

  [OperationContract]
  ProductData GetProduct (string productName);

  ...
}
```

- Class `CompanyService` should implement these operations.

Data Contract

- Data is represented in the `ProductData` class;
- The `DataMember` attribute states that a property must be serialized;

```
[DataContract (Namespace="unique_namespace")]
public class ProductData {
  [DataMember] public string productName { get; set; }
  [DataMember] public int productQuantity { get; set; }
  ...
}
```

Configuring

- A WCF service is comprised by the Address, the Binding and the Contract (ABC):
  - *Address*: the location of the service (the where);
  - *Binding*: the protocol and encoding (the how);
  - *Contract*: the capabilities of the service (the what);
- These define an *endpoint*.

Configuring

- Endpoints are defined in the *App.config* file;
  - Visual Studio provides an editor;
- Selecting the binding (the communication protocol) depends on the purpose of the service;

Hosting

- WCF libraries can be run from Visual Studio for debugging;
- Deployment technology depends on the goal of the service;
- A server application could be run over the *Internet Information Services* (IIS);
- If the application is to communicate directly with other applications, it can be self-hosted.

```
static void Main(string[] args) {
  using (ServiceHost serviceHost =
                new ServiceHost(typeof(CompanyService)))
  try {
    serviceHost.Open()
    ...
  }
}
```

Consuming

- Just add the service reference to the application;
- Then, instantiate the service class and call the contracted operations:

```
CompanyService service = new CompanyService();
ProductData p = service.GetProduct("Milk");
```

Bibliography


N. Randolph, D. Gardner, M. Minutillo and C. Anderson
Professional Visual Studio 2010.
Wrox, 2010.