

TTCN-3 for .NET

Ulrich Grude

Metarga GmbH, TFH Berlin
grude@tfh-berlin.de

Friedrich Wilhelm Schröer

Fraunhofer FIRST
f.w.schroerer@first.fraunhofer.de

Peter Enskonatus

Fraunhofer FIRST
peter.enskonatus@first.fraunhofer.de



Overview

- Introduction
- The TTCN-3.NET Compiler
- The TTCN-3.NET Runtime
- Conclusion

Introduction

- Combining TTCN-3 and .NET
- Some Advantages of .NET
 - Unifies many Languages (by CIL, CRT, CTS)
 - ECMA and ISO Standard
 - Independently implemented by Novell
 - Runs under Windows, Linux, Mac OS, ...

The TTCN-3.NET Compiler

- Target language C#
 - Adapters and Codecs in any .NET language (e.g. C/C++, C#, Java, ...)
- Edition 2 of TTCN-3
- Standards
 - **ETSI ES 201 873-1 v2.2.1**
TTCN-3 Core Language
 - **ETSI ES 201 873-5 v1.1.1**
TTCN-3 Runtime Interface (TRI)
 - **ETSI ES 201 873-6 v1.1.1**
TTCN-3 Control Interface (TCI)

Tools used to build the Compiler

- **Gentle**

An integrated System for Analysis, Transformation and Synthesis

- **Accent**

A Compiler Compiler for the Entire Class of Context-free Grammars

The Parser is directly based on the Standard

- **Valis**

A Compiler Validation System

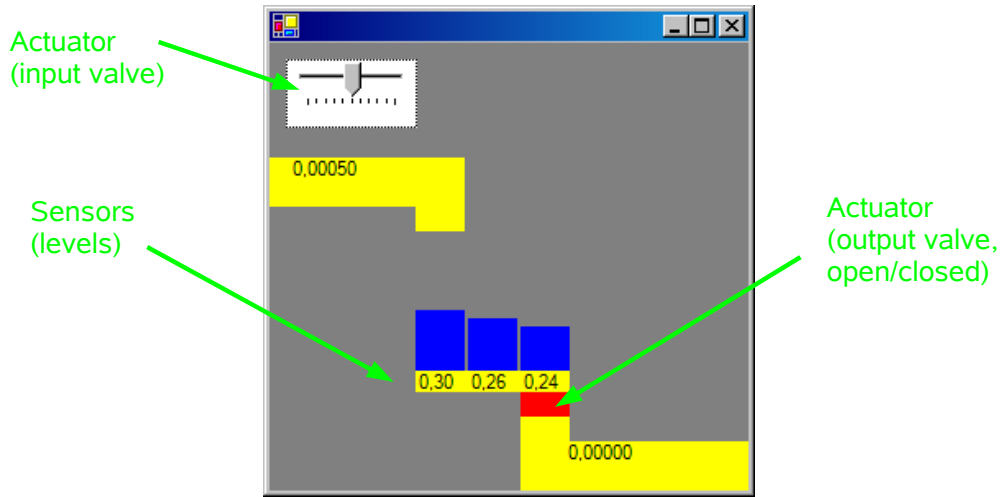
Test-driven Development

About 100.000 Lines of TTCN-3 Code

The TTCN-3.NET Runtime

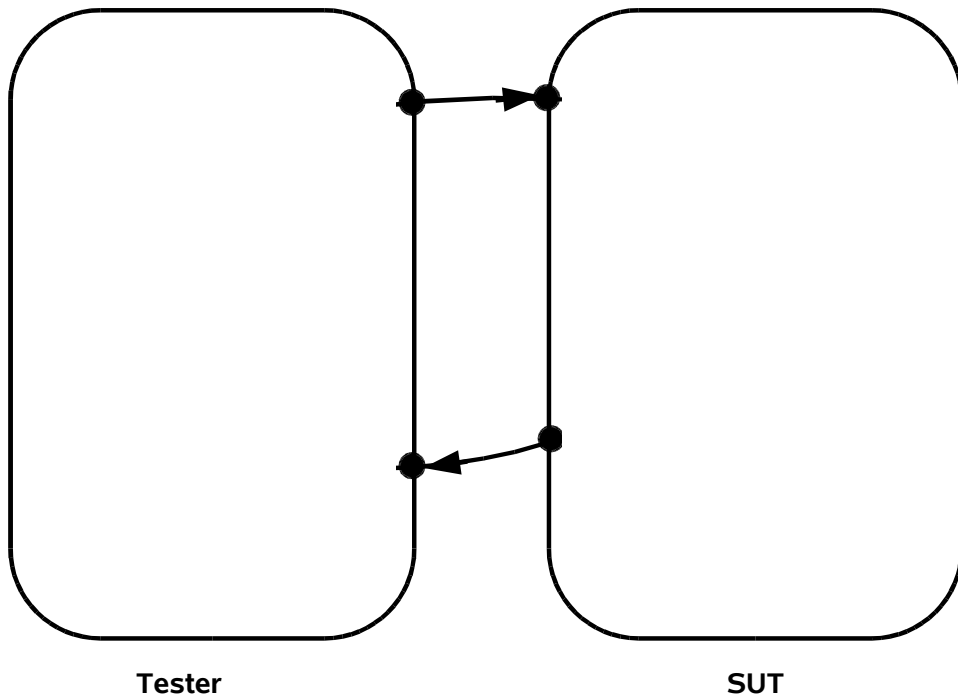
- Implements the Standards TRI and TCI
- Connections to Modelling System Modelica
- Distributed Load Tests
- Continuous Systems

An Example

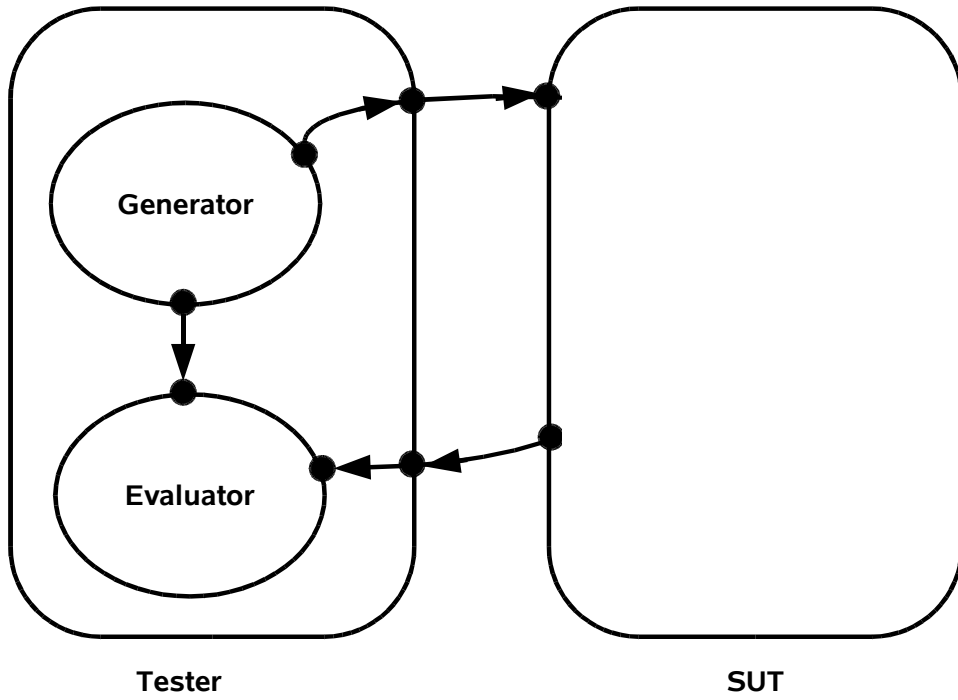


Three connected Tanks

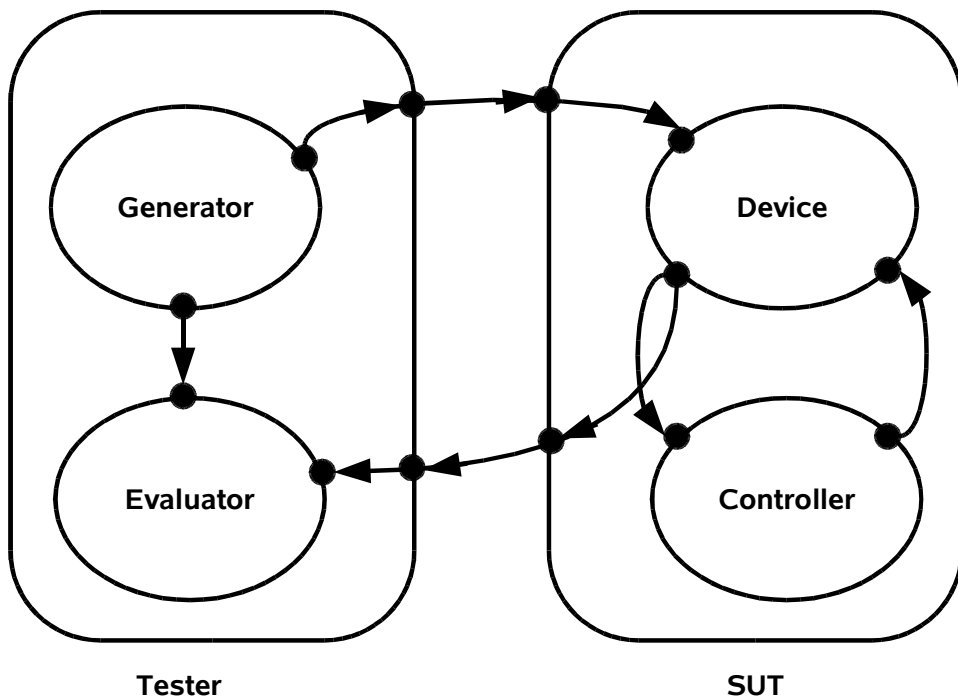
An Example



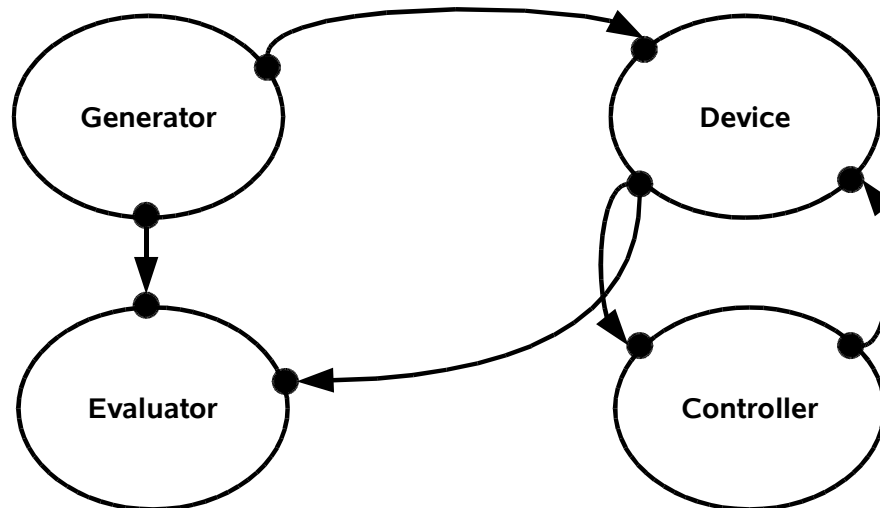
An Example



An Example



An Example



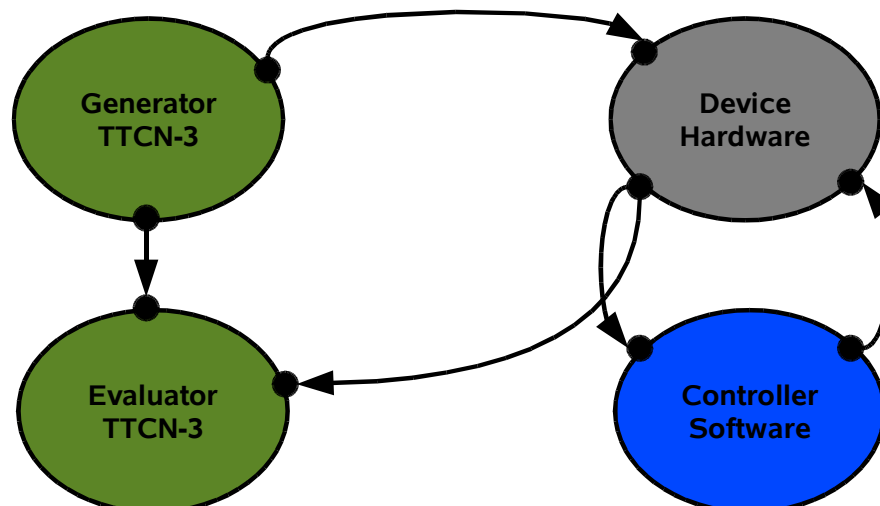
Late Technology Binding Possibilities and Advantages

- A Component in different Project Phases: Model, Mockup, Product
- Test a Model against a Tester
- Test a Tester against a Model
- Use TTCN-3 as a Modeling Language (e.g. for the Controller in the Example)
- Distribute Components of the Tester

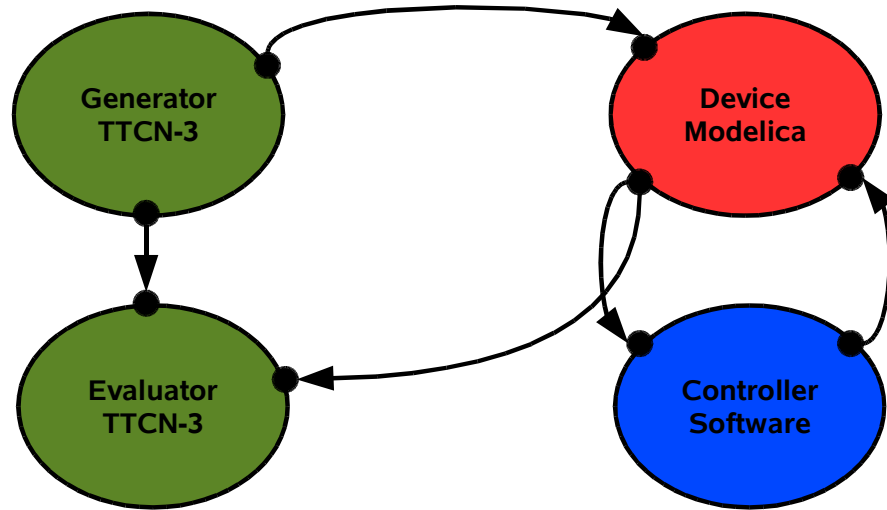
Technologies for Components

- TTCN-3 Component
- Tested Software Component
- Tested Hardware Component
- Model written in Modelica

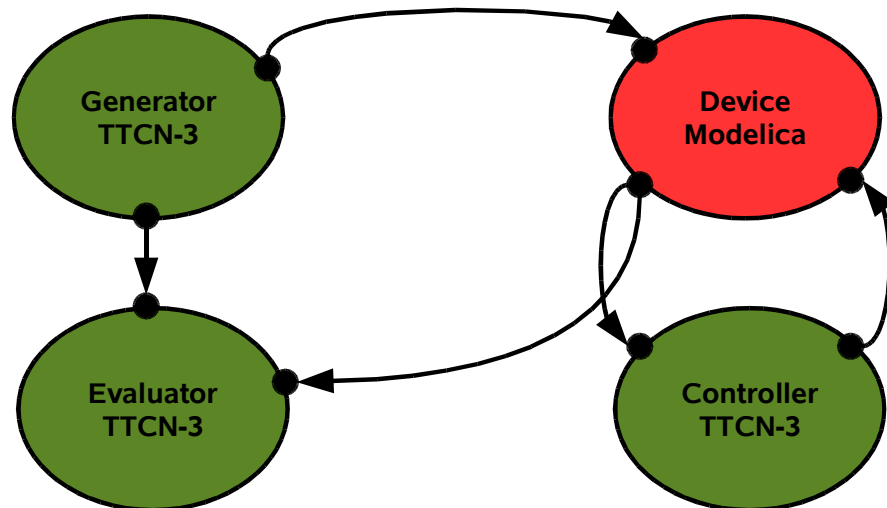
An Example



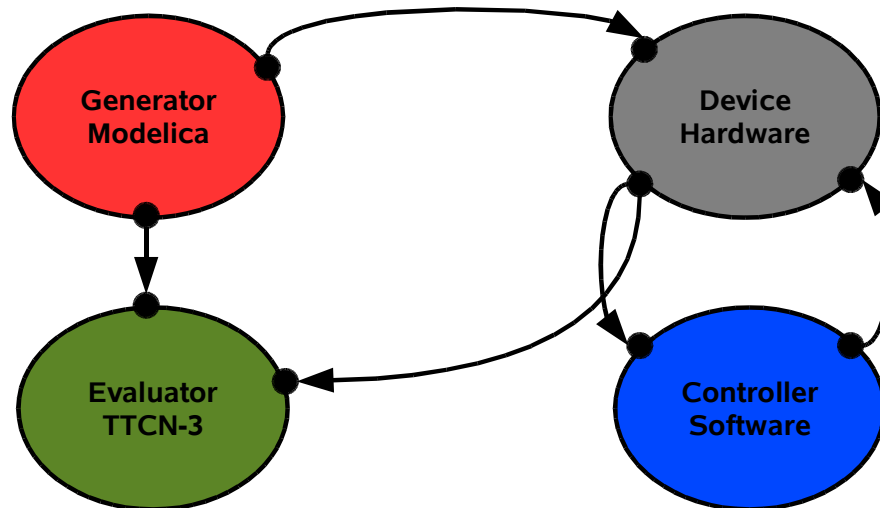
An Example



An Example



An Example



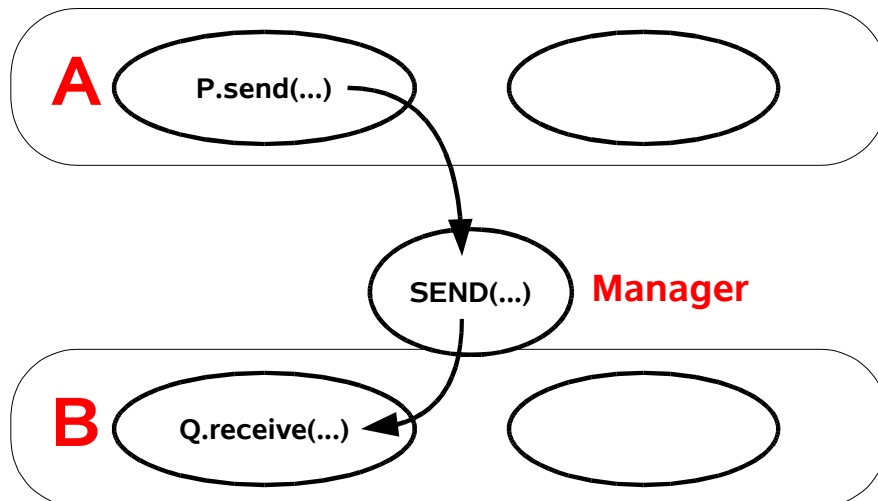
How do we get there? With Managers!

- Every Component is assigned to a **Manager**.
- Managers **create** and manage **Components**
- Messages from Component **A to B** are sent via the **Manager of B**.

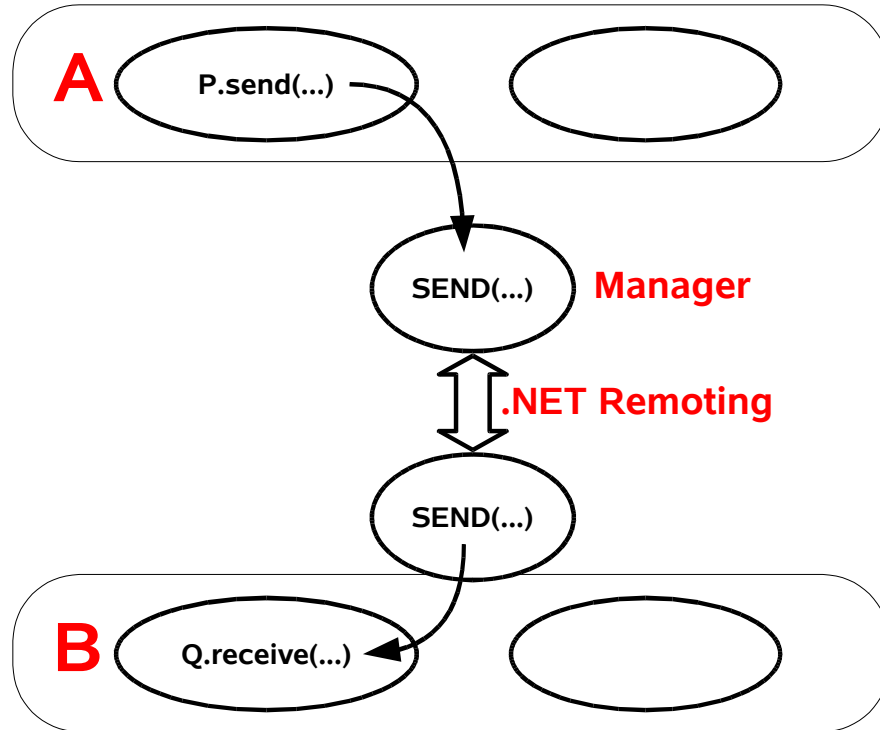
User may choose Technology

- User may write **AssignManager** function
- User may write **Manager Objects** to access arbitrary Components realized in any Technology
- All Managers must implement the new interface **TTCN3 . IComponentManager**

A Message from A to B



A Message from A to remote B



Conclusion

TTCN-3 .NET

- **Independant** of Languages
- **Open** for Heterogenous Technologies
- **Ready** for Distribution