

TTCN-3 User Conference 2010
Beijing, China

Test System Modelling and TTCN-3 Code Generation

-Xinyi Chen, Kao Bi, Ji Wu
Software Engineering Institute,
Beihang University



Agenda

- Motivation
- Model-Driven Testing
- U2TP & Test System Model
- Test System Modelling Example
- TTCN-3 Code Generation
- Conclusion

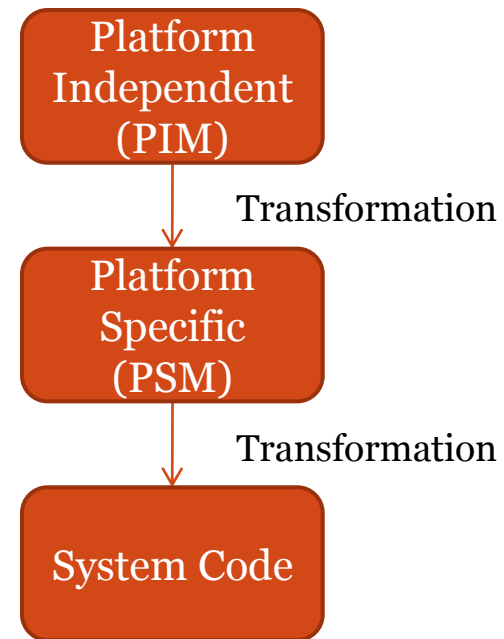


Motivation

- How to capture the implicit idea in testers mind
- How to make test design easier to be understood and maintained
- How to improve efficiency of test system design and implementation

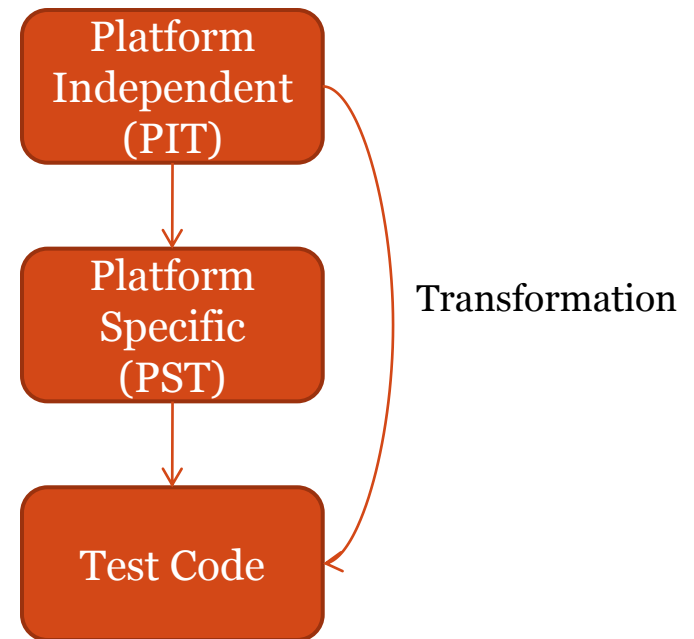
MDA(Model-Driven architecture)

- Model-driven architecture (MDA) is a software design approach for the development of software systems.
- It was launched by the Object Management Group (OMG) in 2001.
- It provides a set of guidelines for the structuring of specifications, which are expressed as models.
- Model-driven architecture is a kind of domain engineering, and supports model-driven engineering of software systems.



MDT (Model-Driven Testing)

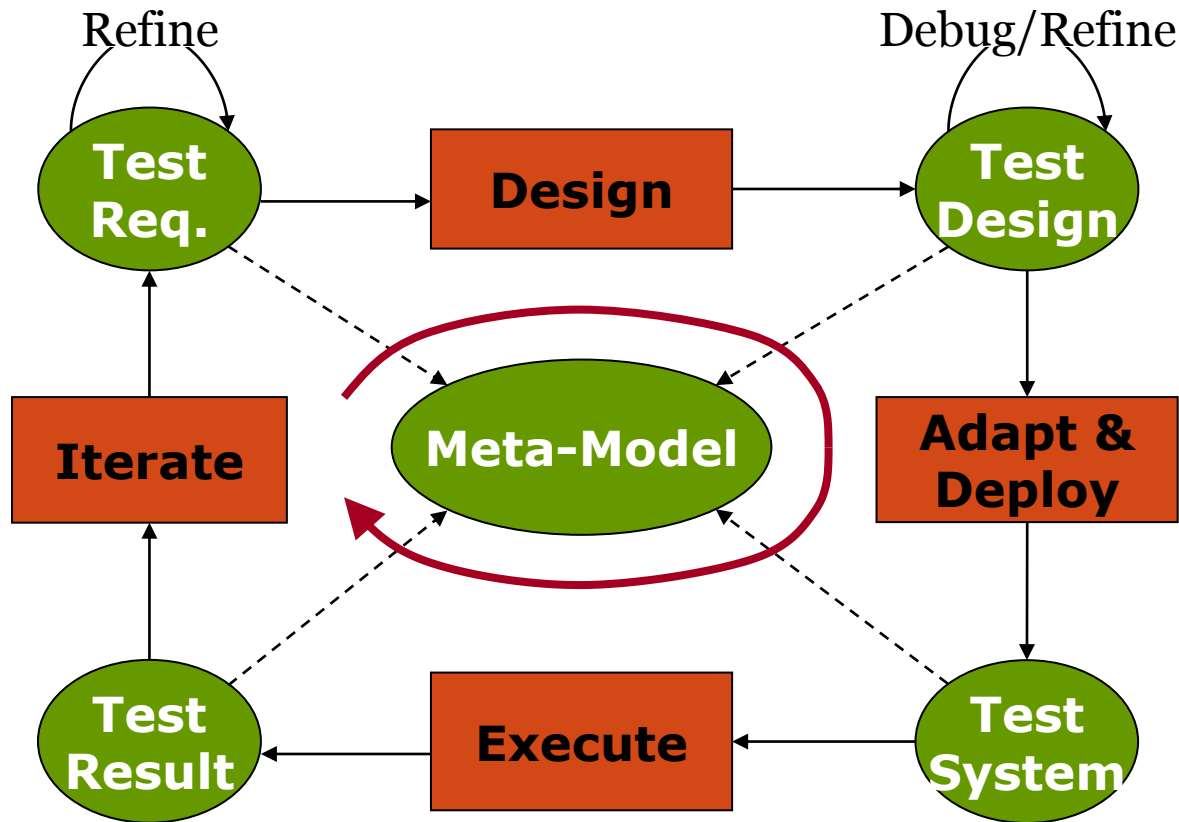
- The philosophy of MDA can also be applied on test modelling, which is called Model-driven testing.
- In MDT, testers use test system model to analyze test requirement and design the test system.
- Test system model describes all aspects of the test data, test cases and the test configure.
- Finally, test system model(PIT & PST) can be transformed to test code.



Why MDT?

- **Abstraction**
 - works on a high level of abstraction thus makes tester focus on test design
 - easy to be understood and maintained
- **Automation**
 - generate executable test code directly from test model

MDT Framework



U2TP

- The UML 2.0 Testing Profile (U2TP) is a UML 2.0 profile for the testing.
- has become an official OMG standard since March 2004.
- defines a language for designing, visualizing, specifying, analyzing, constructing and documenting the artifacts of test systems.

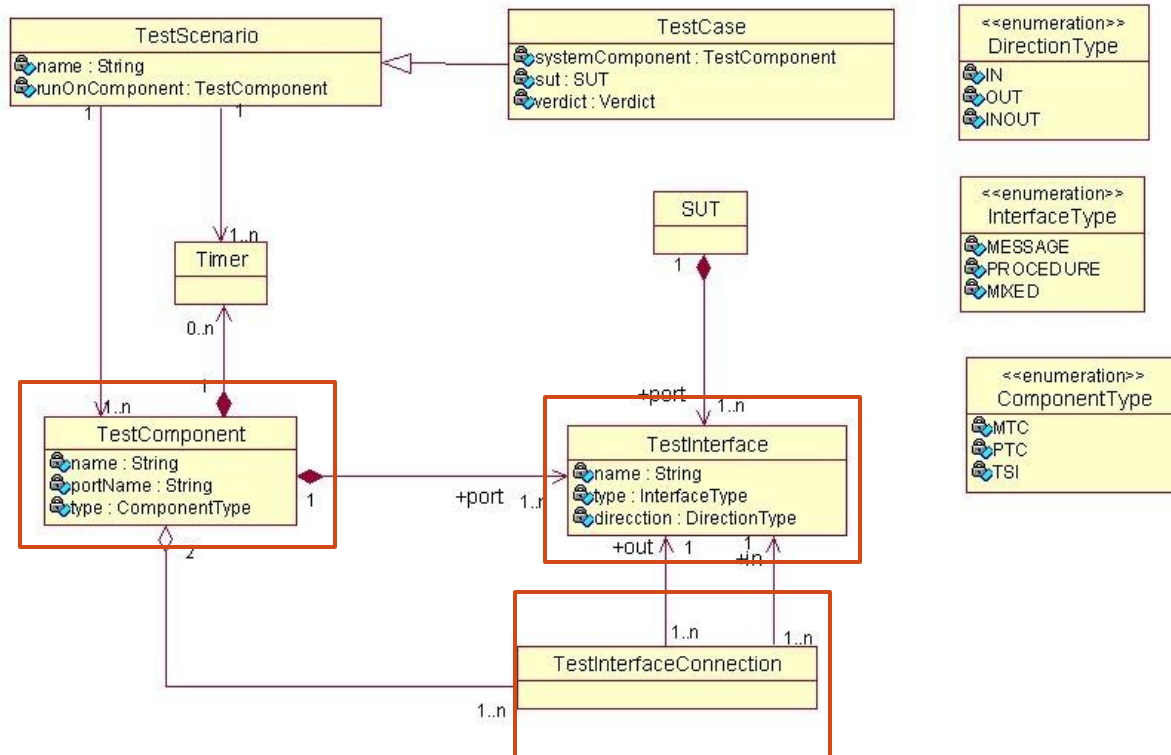
U2TP Concepts

- U2TP introduces four logical concept groups covering the aspects:
 - Test Architecture
 - concepts for test structure and test configuration
 - Test Data
 - concepts for test data used in test procedures
 - Test Behavior
 - concepts for the dynamic aspects of test procedures and addressing observations and activities during a test
 - Time
 - concepts for a time quantified definition of test procedures

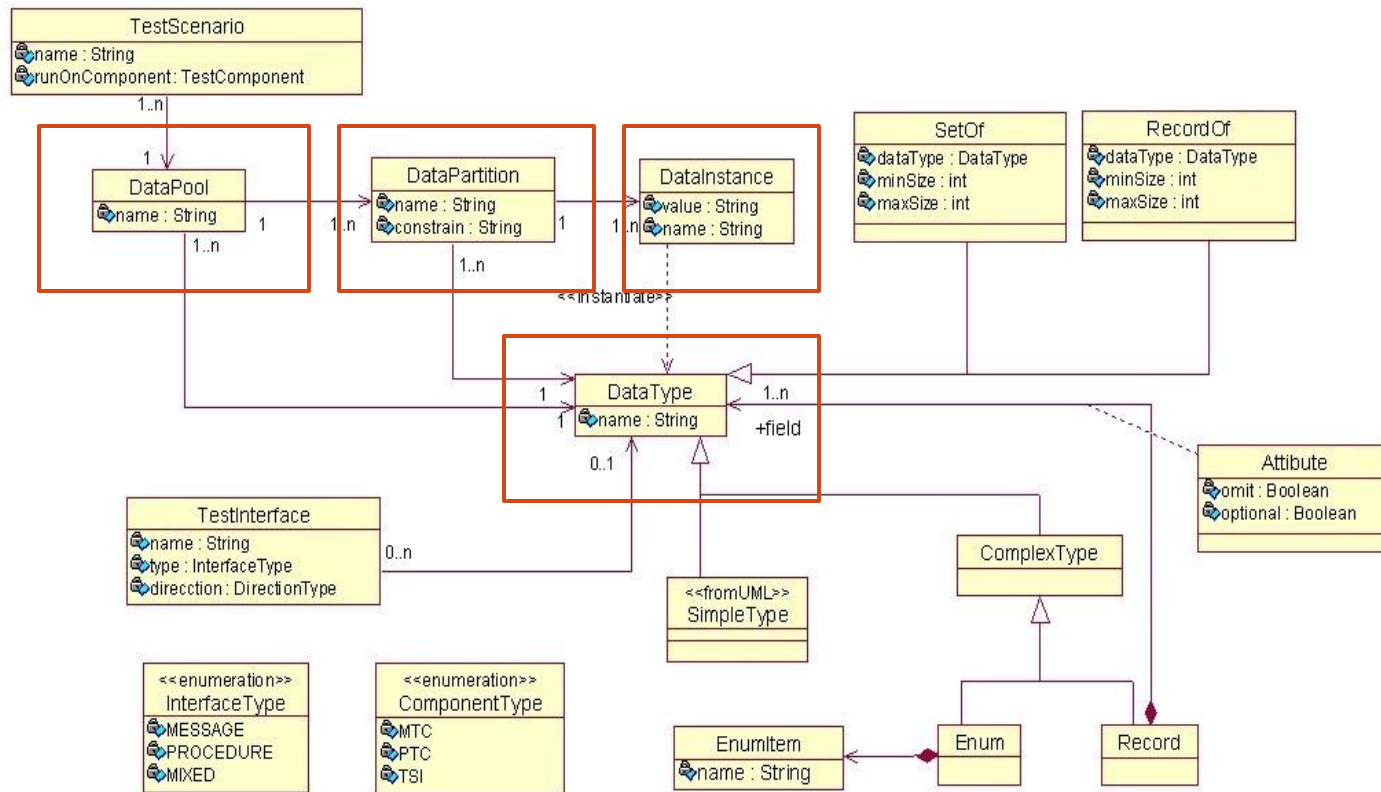
Test System Model

- How to define test system model
 - U2TP is a test modelling language on a higher level
 - TTCN-3 is focus on test execution details
- Meta-model
 - Test Configure
 - Test Data
 - Test Behavior

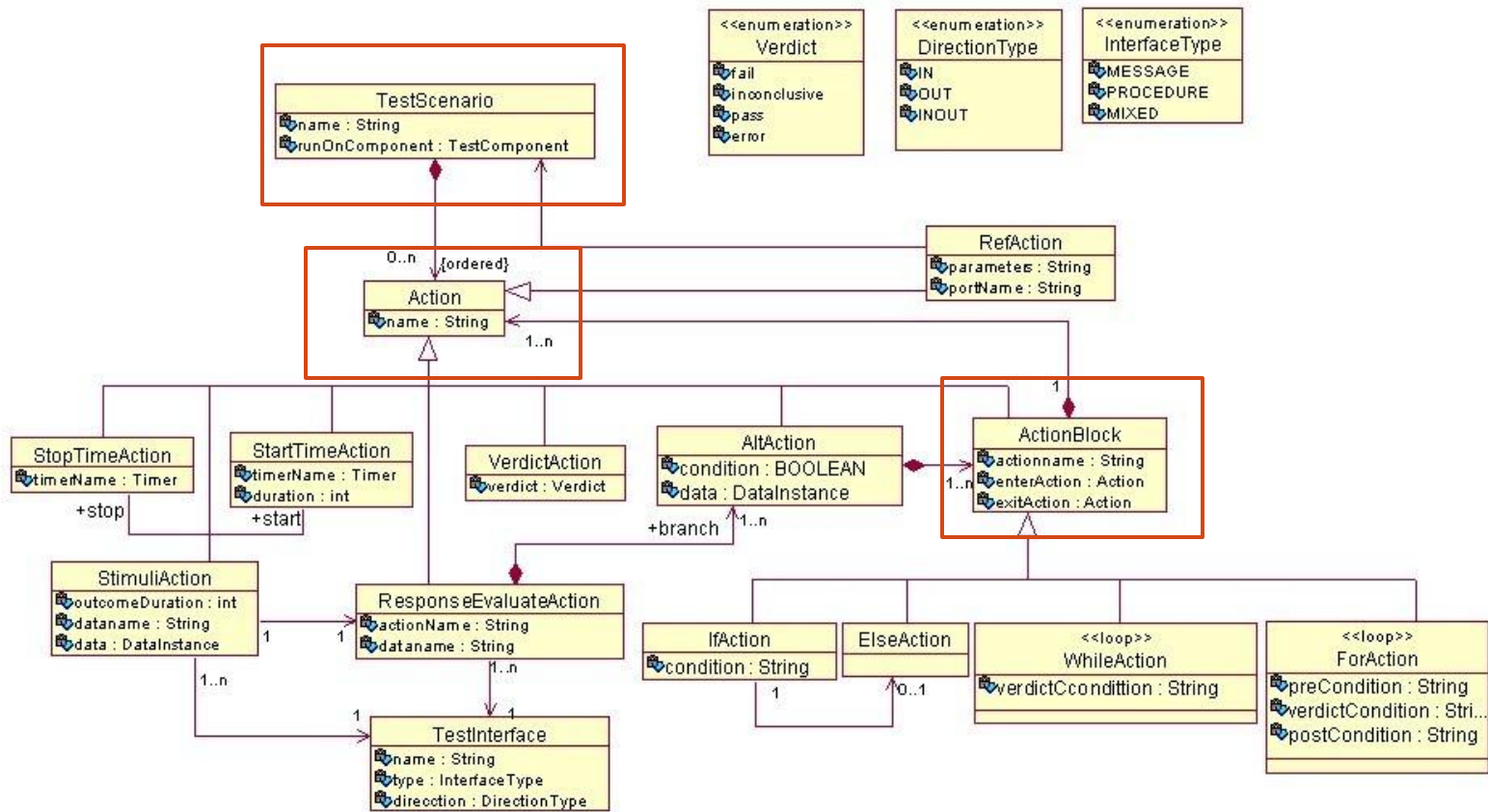
Test System Meta-model: Test Configure



Test System Meta-model: Test Data

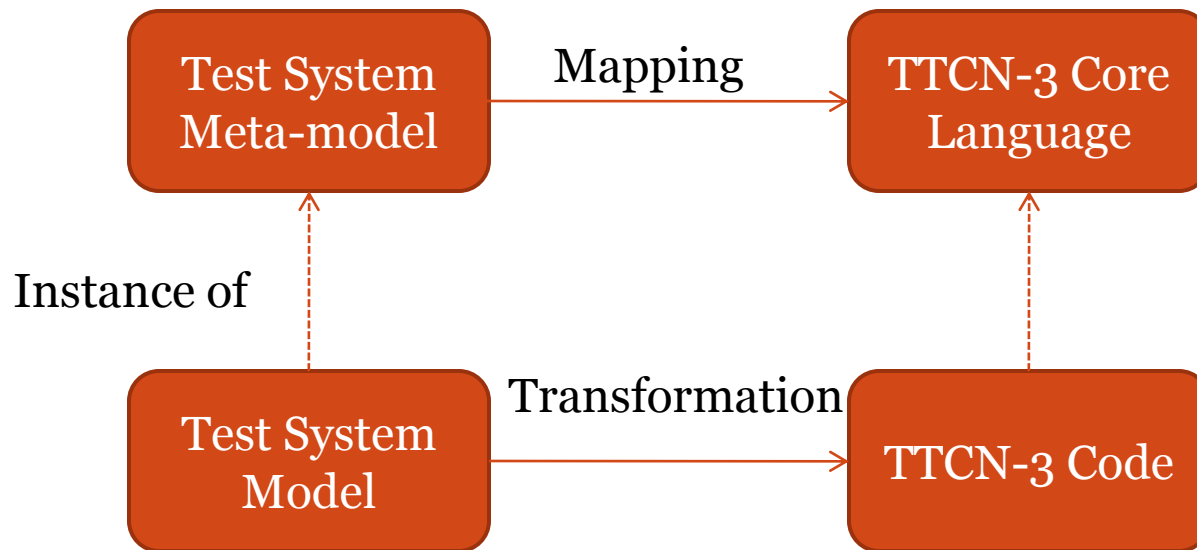


Test System Meta-model: Test Behavior



Aspect	U2TP	Test System Meta-model	TTCN-3 Core Language	
Test Behavior	Test Objective	TestObjective		
	Test Case	TestCase	testcase	
		TestScenario	function	
	Defaults	Default	function	
	Behavior	ExciteAction	send/call/reply	
		ResponseJudgeAction	receive/getcall/getreply	
		DecisionAction	if(...) {...}	
		LoopAction	while(...) {...}	
		RefAction	function call	
		ActionBlock	{...}	
		Verdict	ValidationAction	setverdict
		Arbiter		
Time	Scheduler	TestScheduler		
	Test Control	TestBehavior		
	Timer	StartTimerAction	timer.start	
		StopTimerAction	timer.stop	
Test Configure	Time Zone			
	Package	TestSystem	module	
	SUT	SUT	(test system interface)	
	Test Context	TestArchitecture		
	Test Configuration			
	Test Component	TestComponent	component	
Test Data	Interface	TestInterface	port	
	Data Pool	DataPool	group	
	Data Partition	DataPartition	group	
	Data Selector	DataSelector	function	
	Wildcards		? *	
	Coding Rules	CodingRule		

TTCN-3 Code Generation



Tools

- **Eclipse** plug-in based on **EMF**(Eclipse Modelling Framework) and **GMF**(Graphical Modelling Framework)
- Test System Modelling Tools
 - Test Data
 - Test Configure
 - Test Behavior
- TTCN-3 Code Generation Tools

Java - mode/dns.testaction_diagram - Eclipse Platform

File Edit Diagram Navigate Search Project Run changeZOrder Sample Menu Window Help

Tahoma 9 B I A 100%

abc

- default.dbdesigner
- default.dbdesigner_diagr
- default.testaction
- default.testaction_diagra
- default.typesystem
- default.typesystem_diag
- default2.dbdesigner
- default2.dbdesigner_diagr
- default2.testaction
- default2.testaction_diagr
- default2.typesystem
- default2.typesystem_dia
- MyWSDLFile.wsdl
- gfgh
- model
- p1

*dns.testaction_diagram

```

sequenceDiagram
    participant client as client:MTC
    participant dns as dns:SUT
    client->>dns: record requestA
    dns-->>client: record responseA
    dns-->>client: record other
  
```

client:MTC dns:SUT

record requestA

Alt

record responseA

PASS

record other

FAIL

Palette

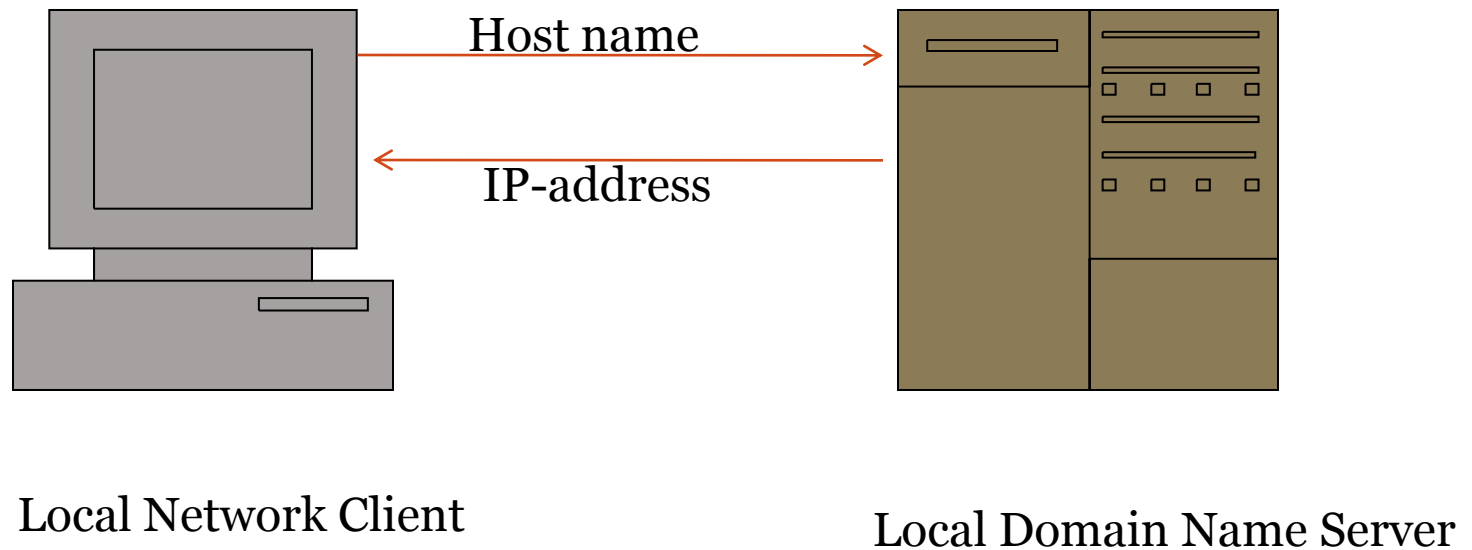
- TestScenario
- TestComponent
- Port
- ExciteAction
- ResponseJudgeAction
- StructuredAction
- RefAction
- WhileAction
- ForAction
- IfAction
- Local Specification
- LocalVariable
- Timer
- Timer
- TerminateTimerAction
- StartTimerAction
- StopTimerAction

Problems Javadoc Declaration Properties InterfaceType Setting TypeSetting view

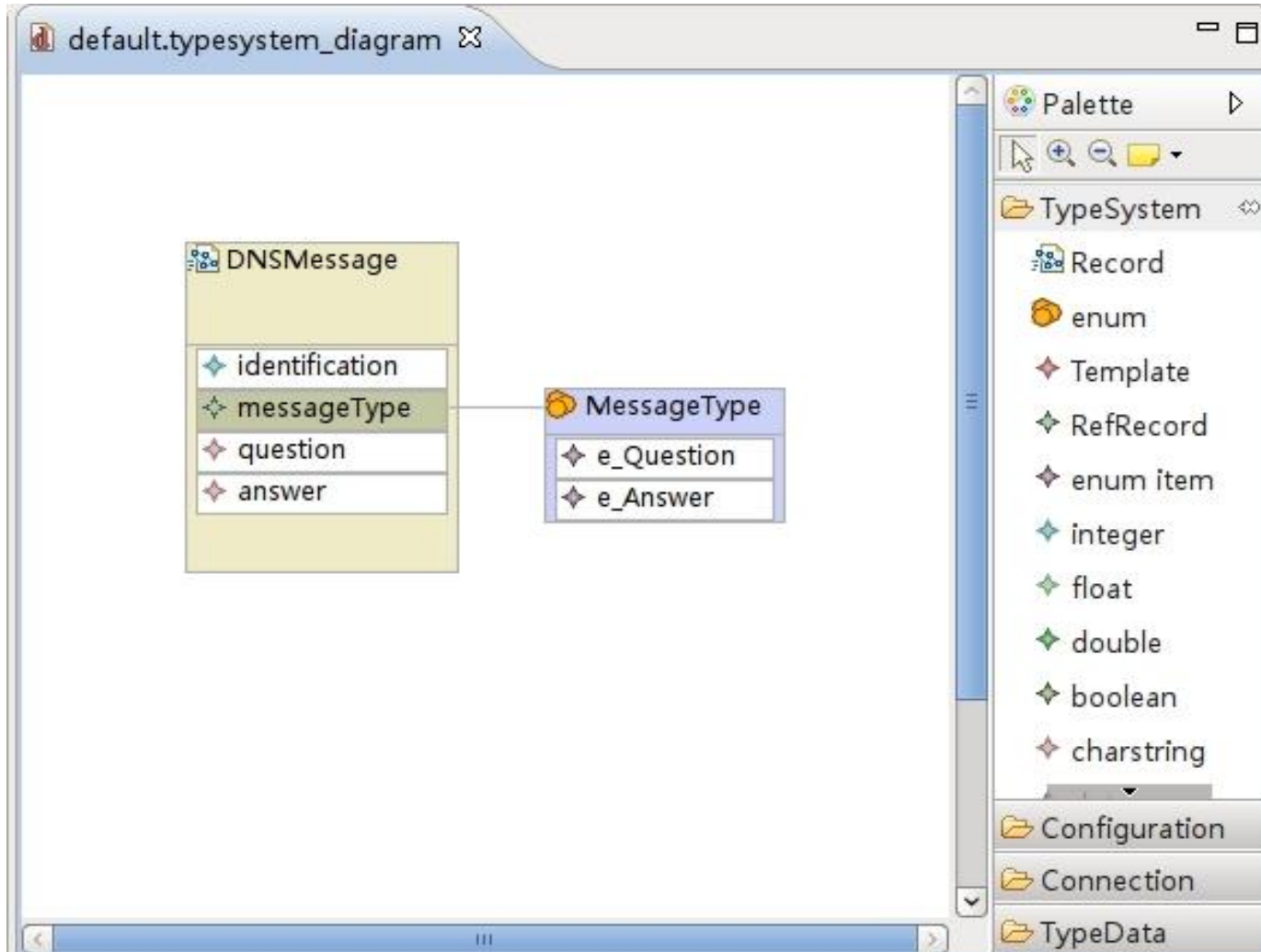
TestComponent

Core	Property	Value
Appearance	Name	dns
	Port Name	
	Timer Name	
	Type	SUT

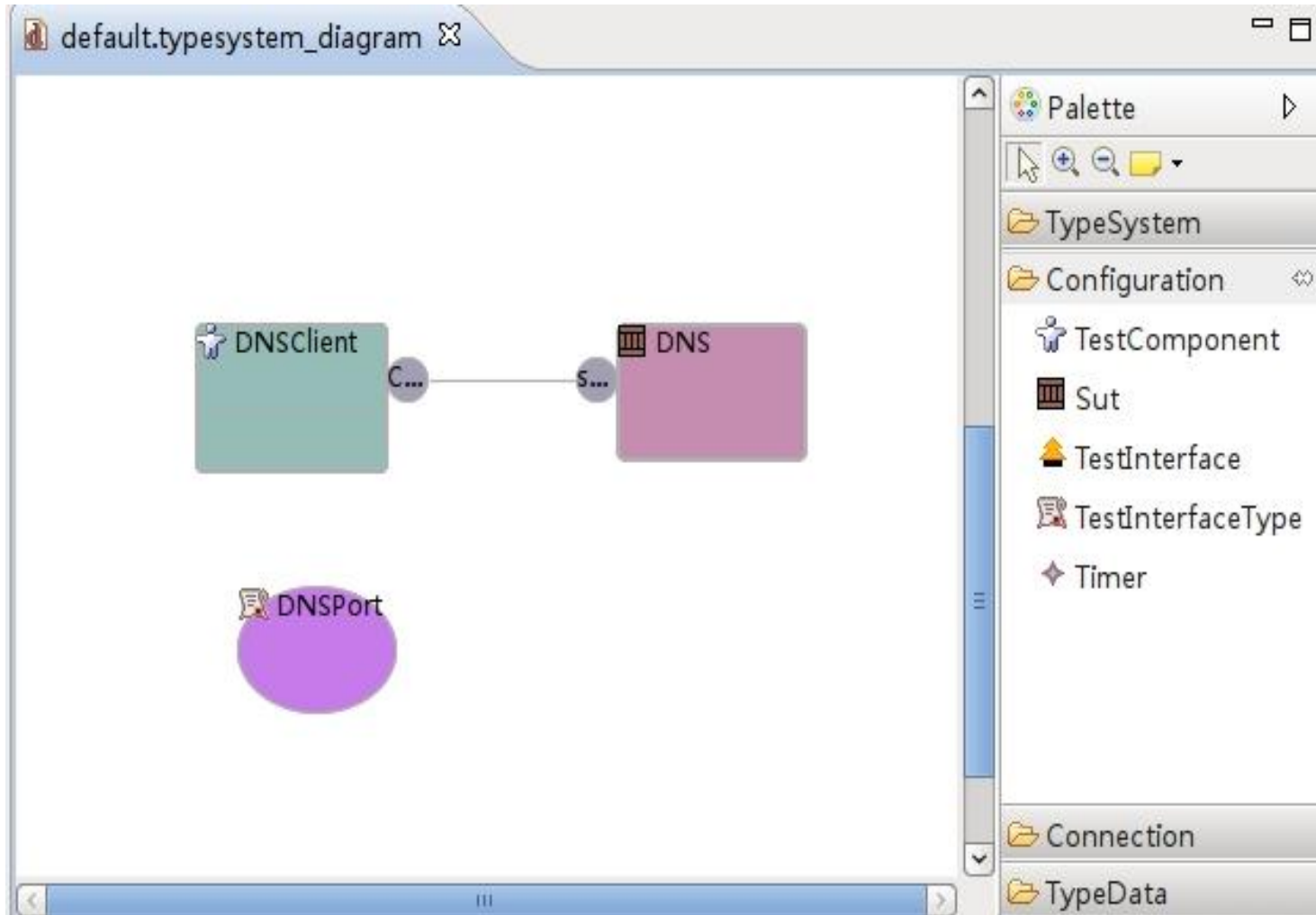
Example: Local DNS Server



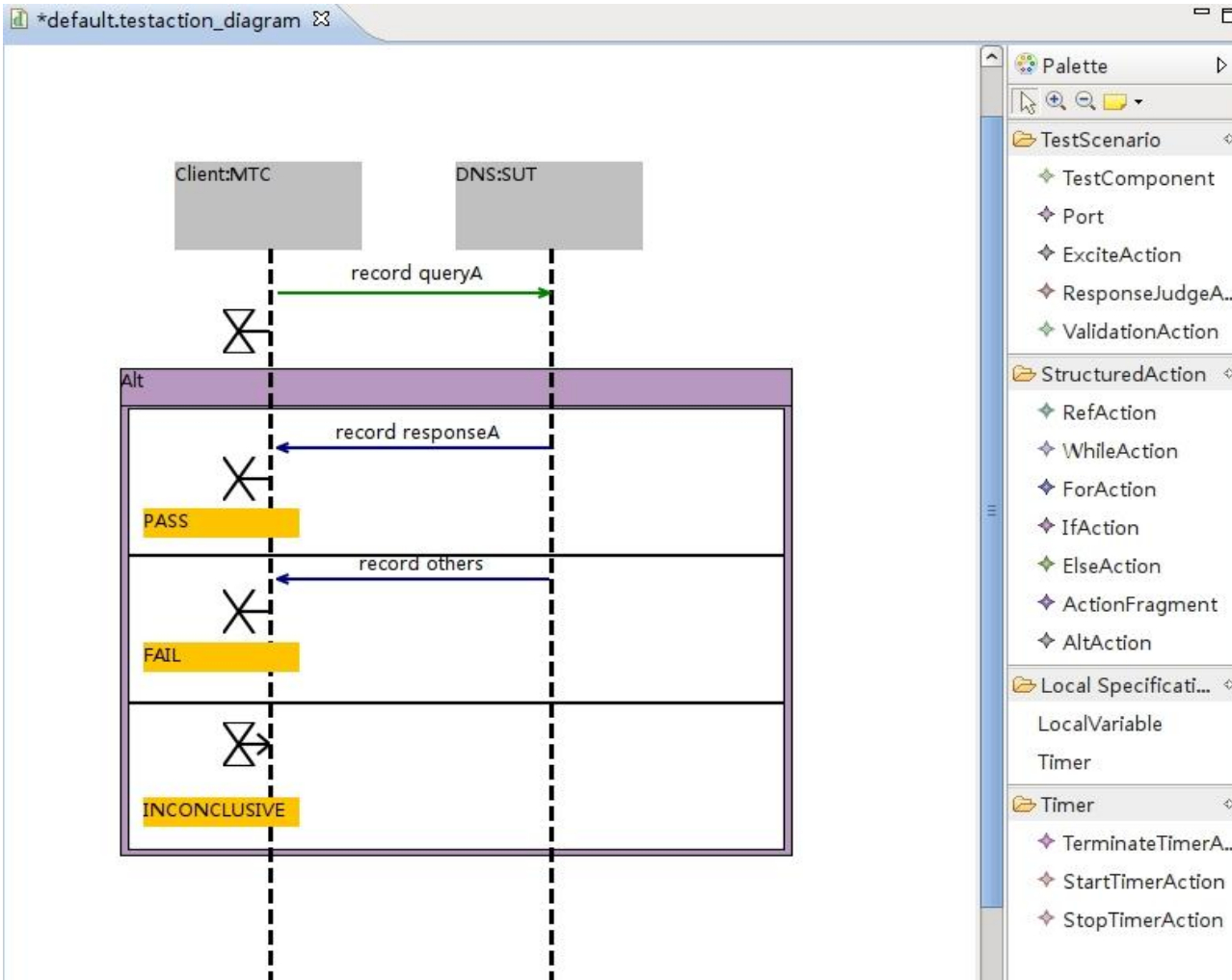
Data Type



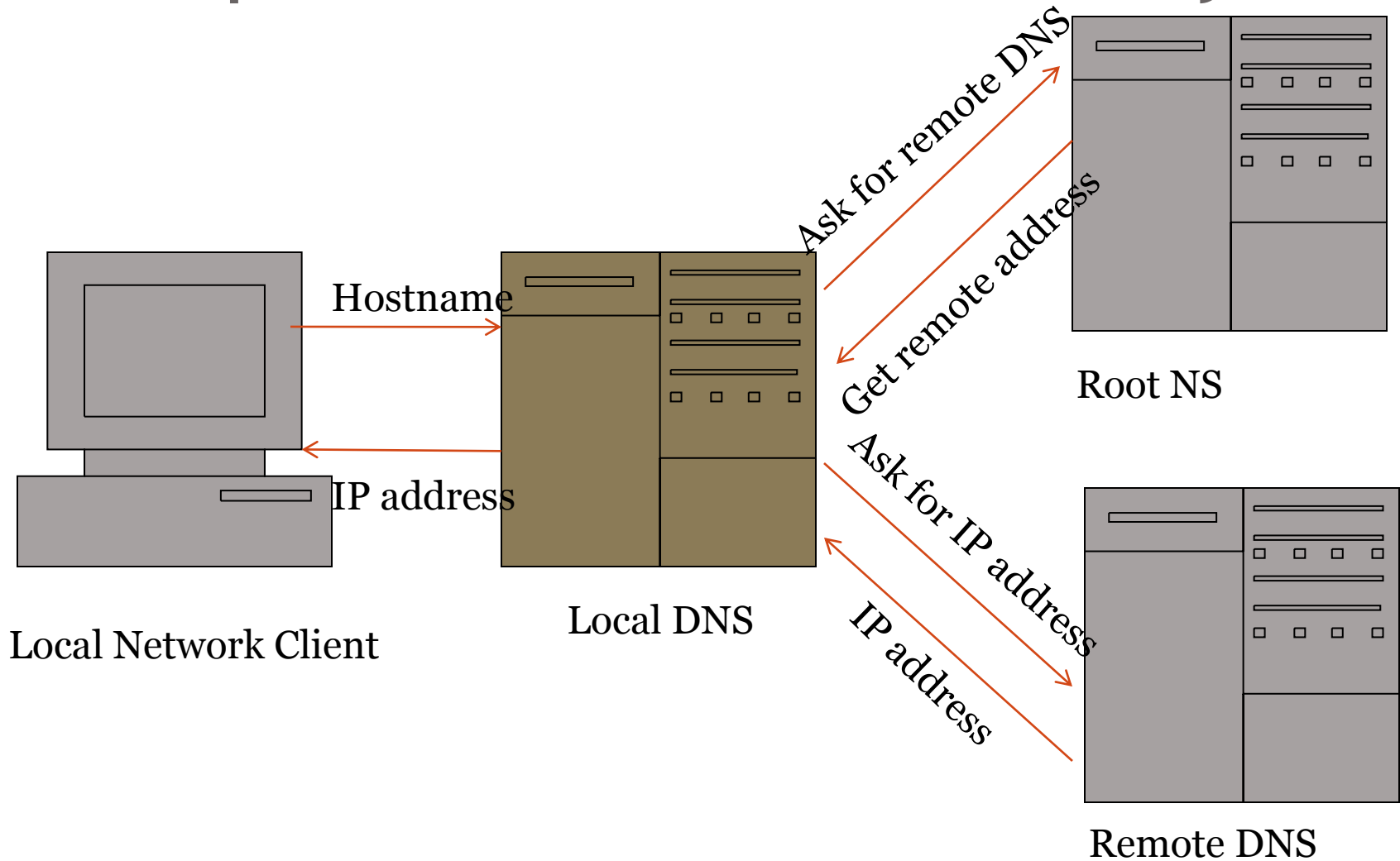
Configure



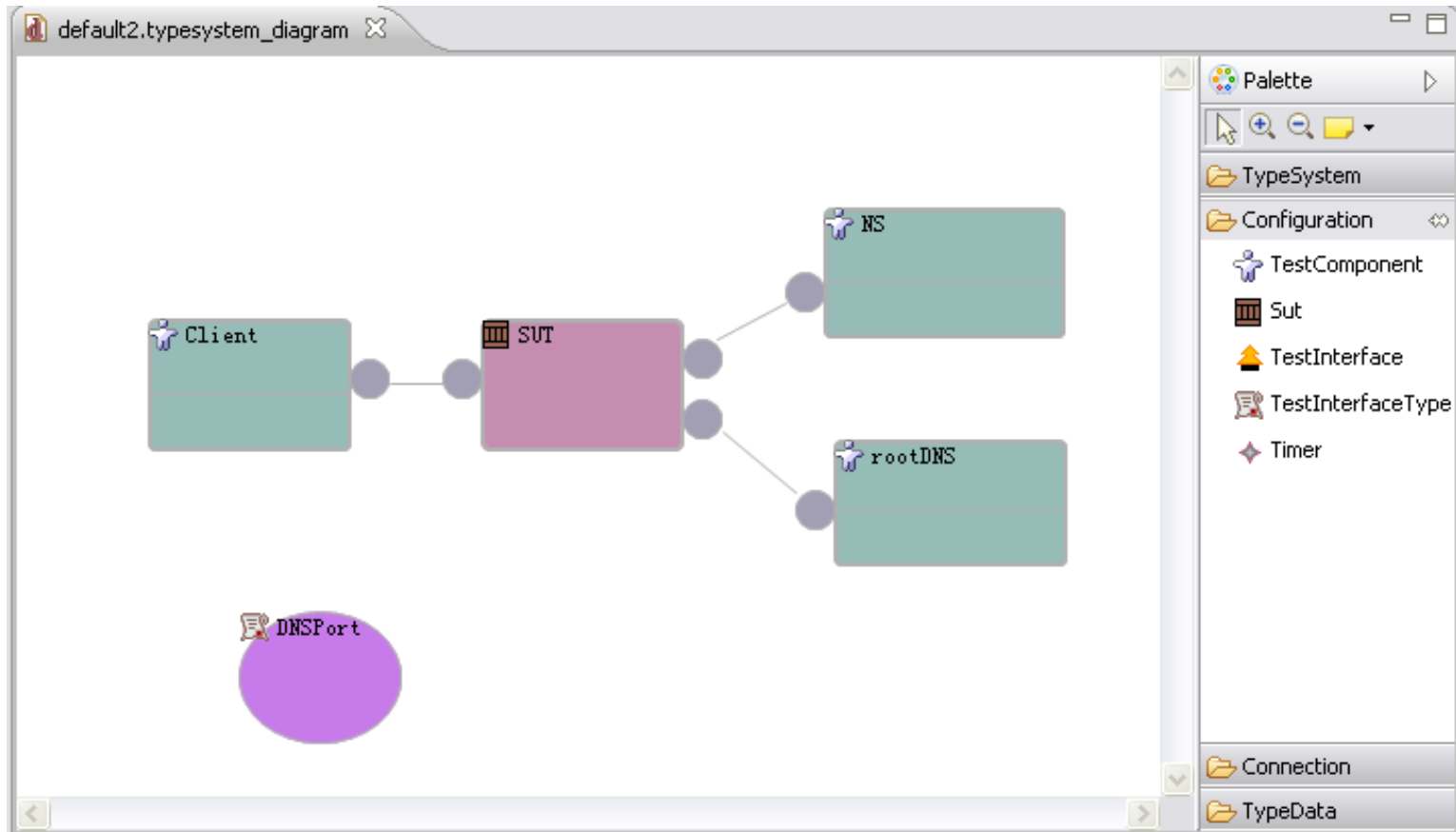
Behavior



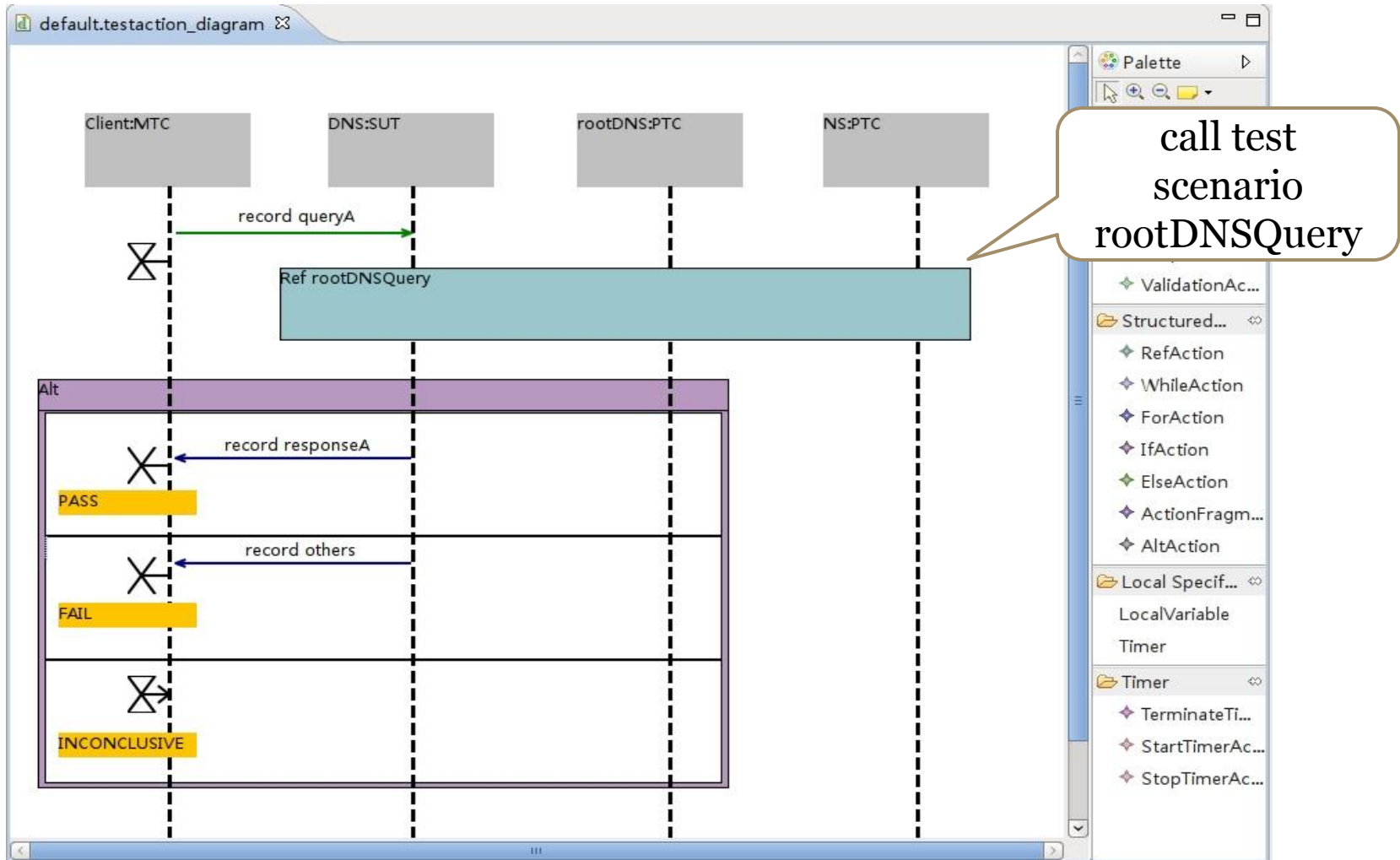
Example: Non-Local DNS Query



Configure

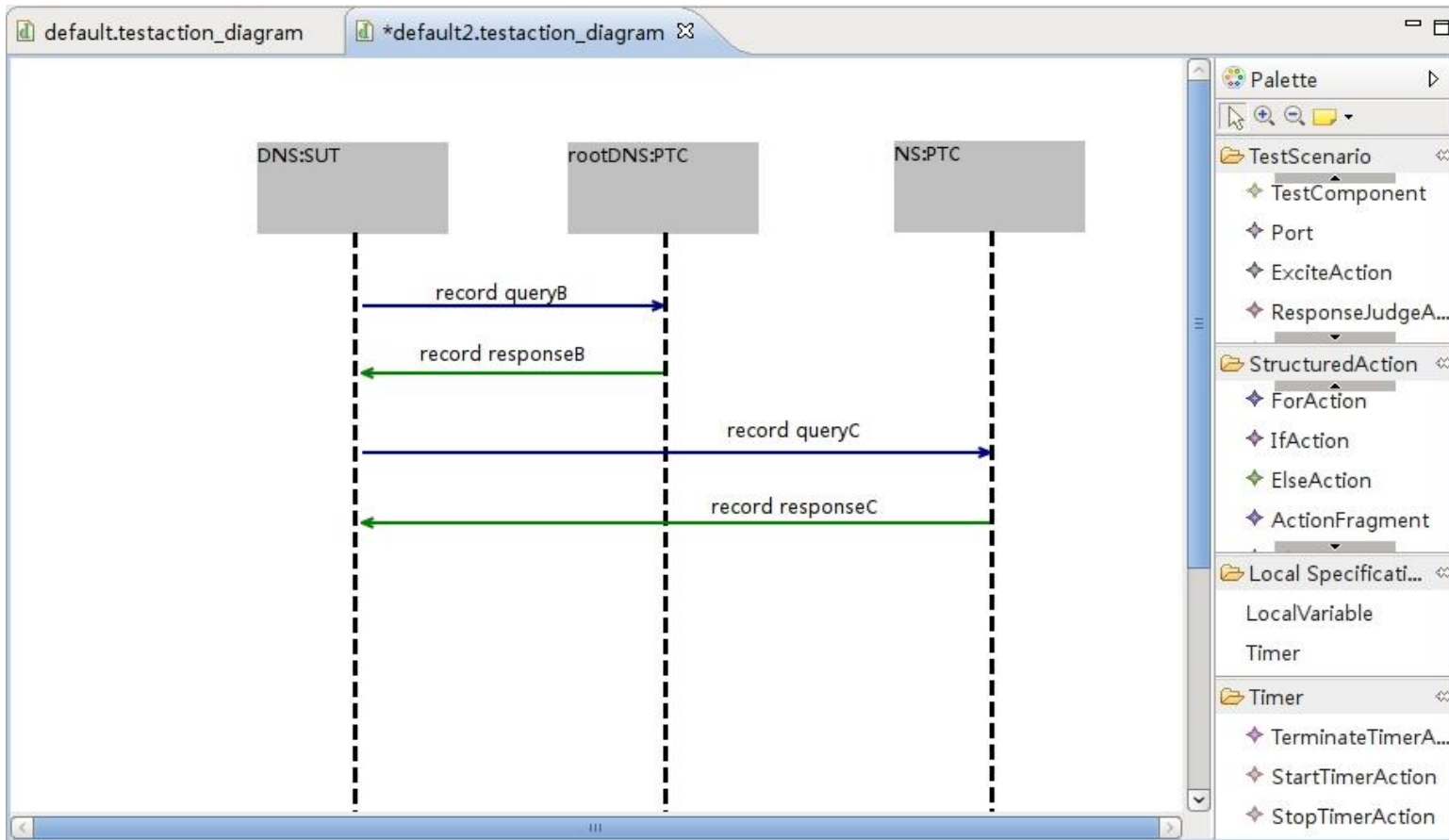


Behavior

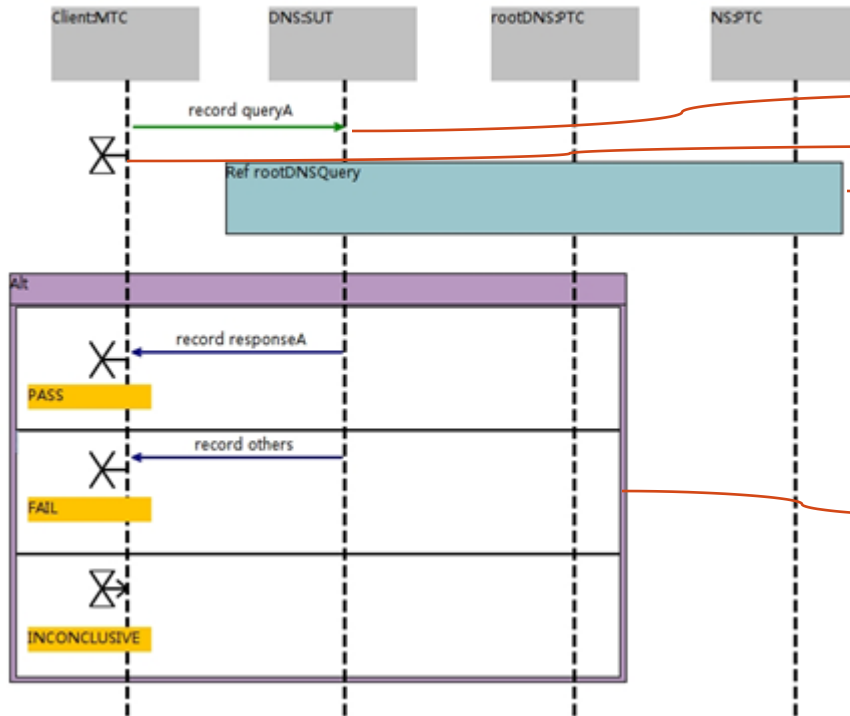


Behavior(2)

TestScenario:rootDNSQuery

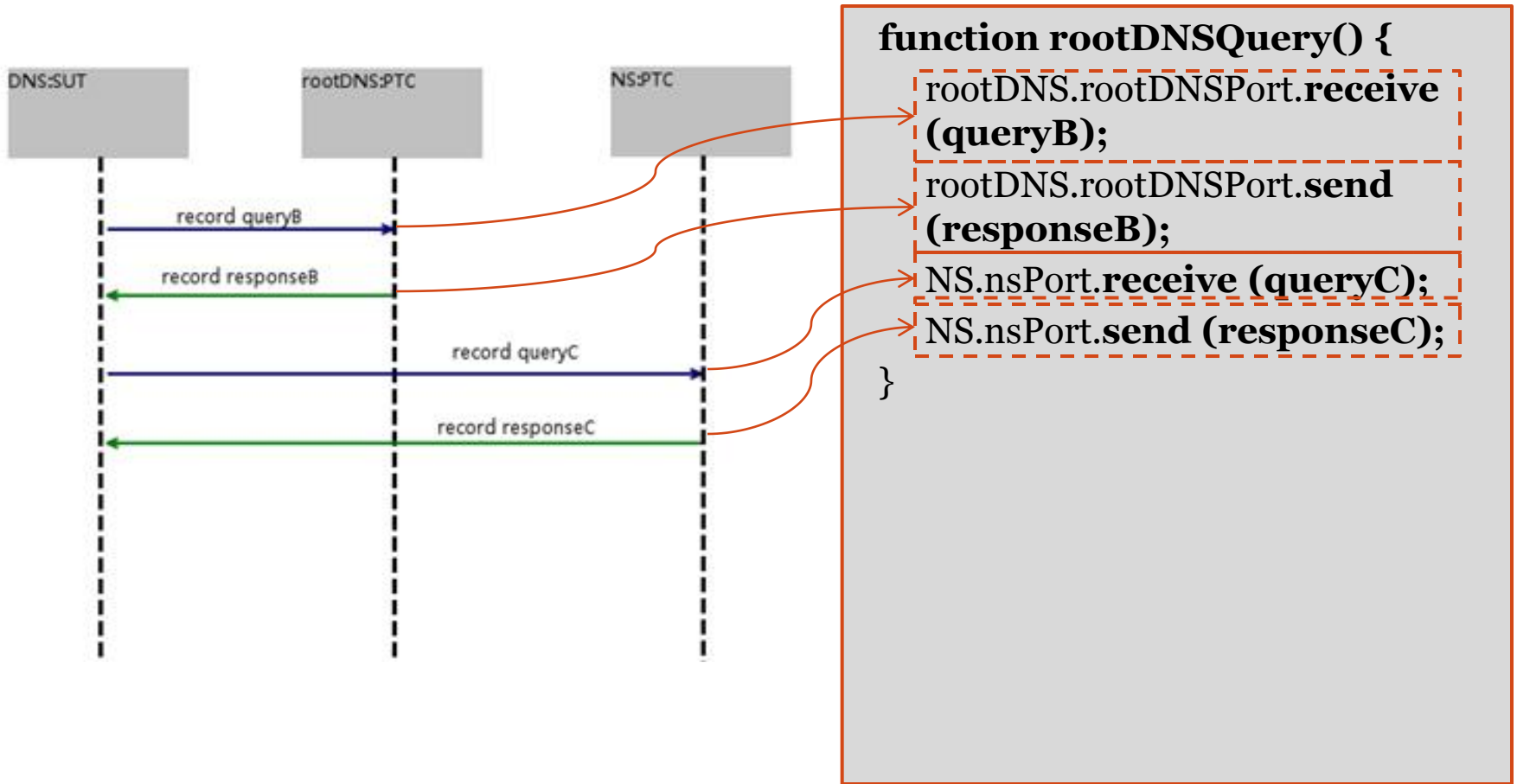


TTCN-3 Code Generation

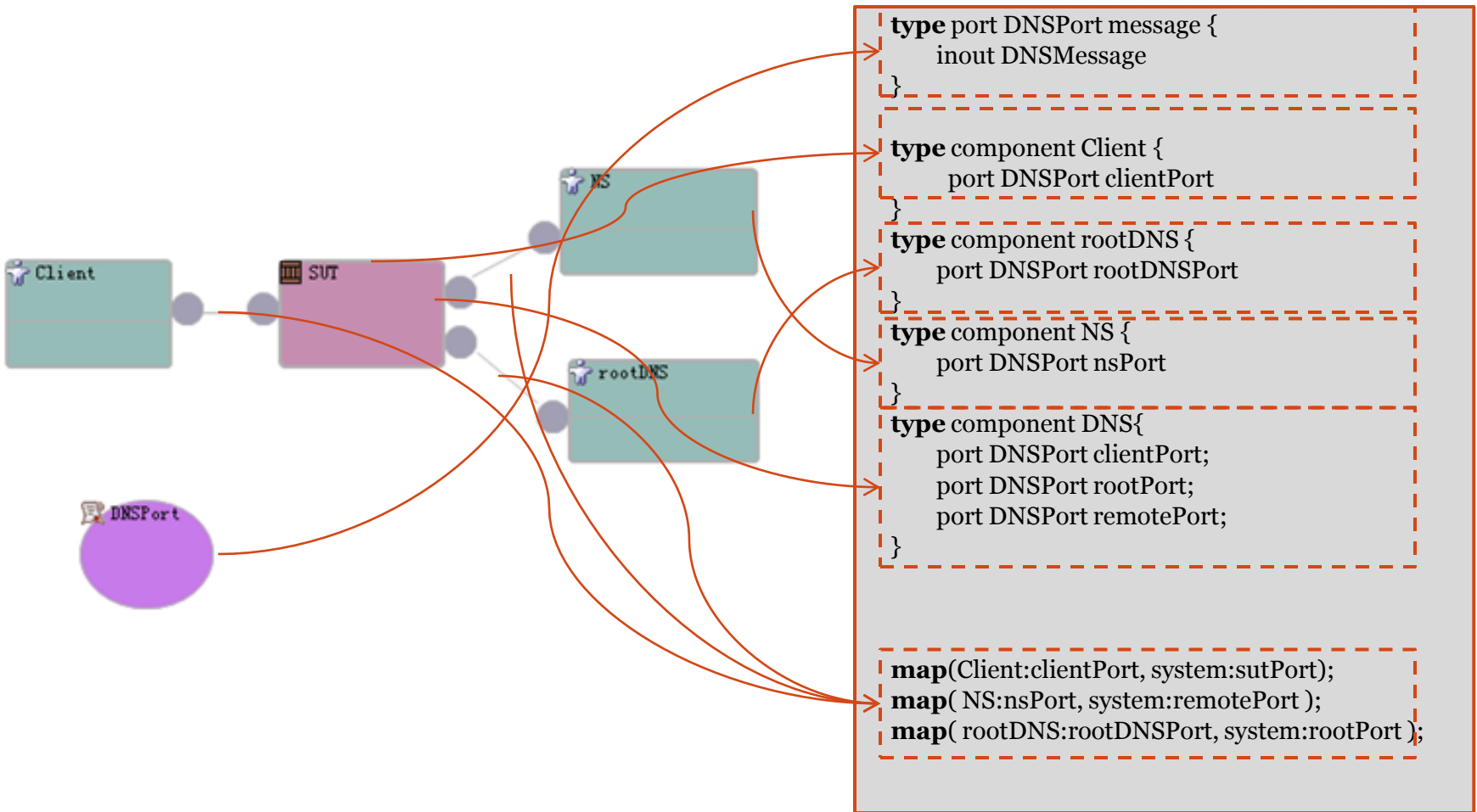


```
testcase testcase1() runs on NS system
DNS {
    clientPort.send (queryA);
    t1.start (10.0);
    rootDNSQuery();
}
alt {
    [] clientPort.receive (responseA) {
        t1.stop;
        setverdict (pass);
    }
    [] client.receive (others) {
        t1.stop;
        setverdict (fail);
    }
    [] t1.timeout {
        setverdict (inconc);
    }
}
```

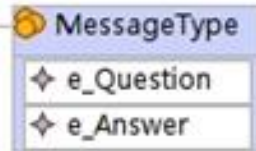
TTCN-3 Code Generation(2)



TTCN-3 Code Generation(3)



TTCN-3 Code Generation(4)



```
type enumerated MessageType  
{e_Question, e_Answer};
```

```
type record DNSMessage {  
integer identification,  
MessageType messageType,  
charstring question,  
charstring answer optional  
}
```

Conclusion

- propose a model driven testing approach
- test system model is defined to describe the test system in three aspects: test data, test configure and test behavior
- a set of tools are developed to visualize the test system model in the period of test design
- these tools can be used to model the static and dynamic characteristics of test system
- automatic generation of executable TTCN-3 from this test system model

Thank you

Contact:

Ji Wu: wuji@buaa.edu.cn

Xinyi Chen: era@sei.buaa.edu.cn

