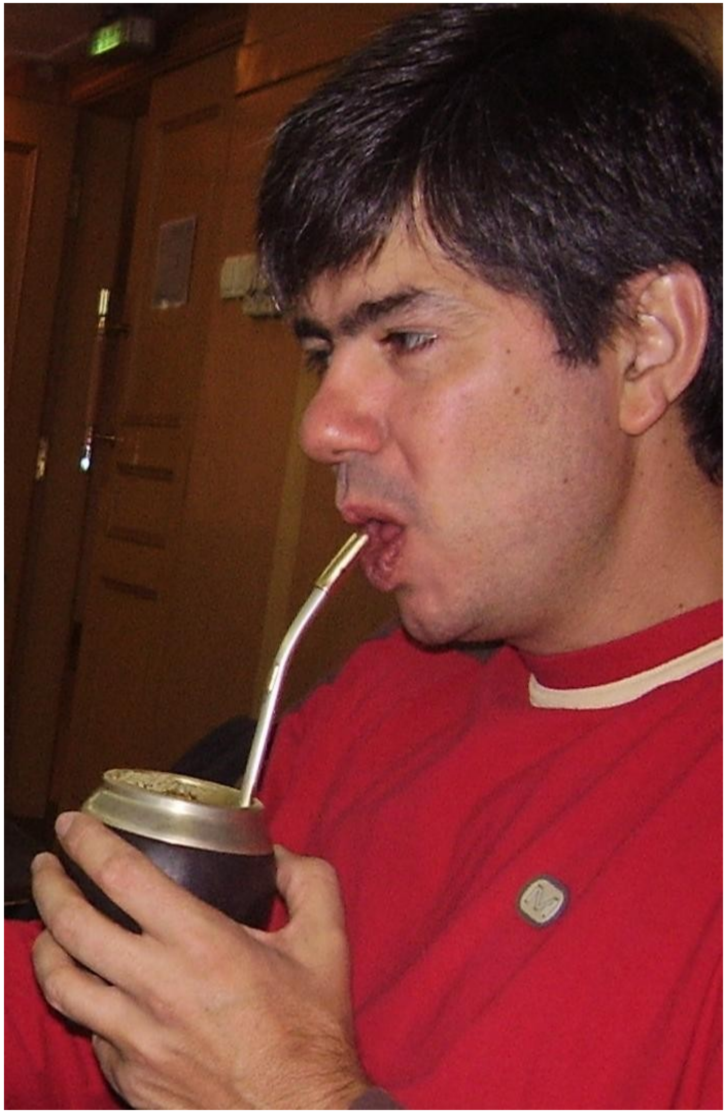
An open compiler for TTCN-3

picoTTCN-3

Ricardo Rezzano (1), Ariel Sabiguero (1), Frank Le Gall (2), Xiaohong Huang(3), Nikolay Pakulin(4), Xianrong Wang (5), Anthony Baire(6)

Partners

France (jTest, INRIA) [6]- Europe (ETSI)
Germany (inno, Fokus) [2]- Spain (Cetecom)
China (BII, CATR, BUPT[3], *IMU*[5])
Russia (Ispras) [4]- Brazil (IPT)
Uruguay (InCo/*UdelaR* [1]*)*



- Ricardo Rezzano

InCo/*Udela,* Uruguay

- Ariel Sabiguero

InCo/*Udela,* Uruguay

- Franck Le Gall
  Inno, Germany

- Nikolay Pakulin
  ISPRAS, Russia

- Xianrong Wang

  Inner Mongolia University

- Anthony Baire
  IRISA, France

Motivation

- TTCN-3 Background
  - broad use of language
  - limitations on use
- Language promotion needs and issues
  - Start level
    - complexity
    - high price
  - Research needs open source implementations

Motivation

- How the free/open compiler helps to promote the use of TTCN-3
  - cheap entry level
  - source code availability
  - tool availability
  - extensibility-standardization
  - ...

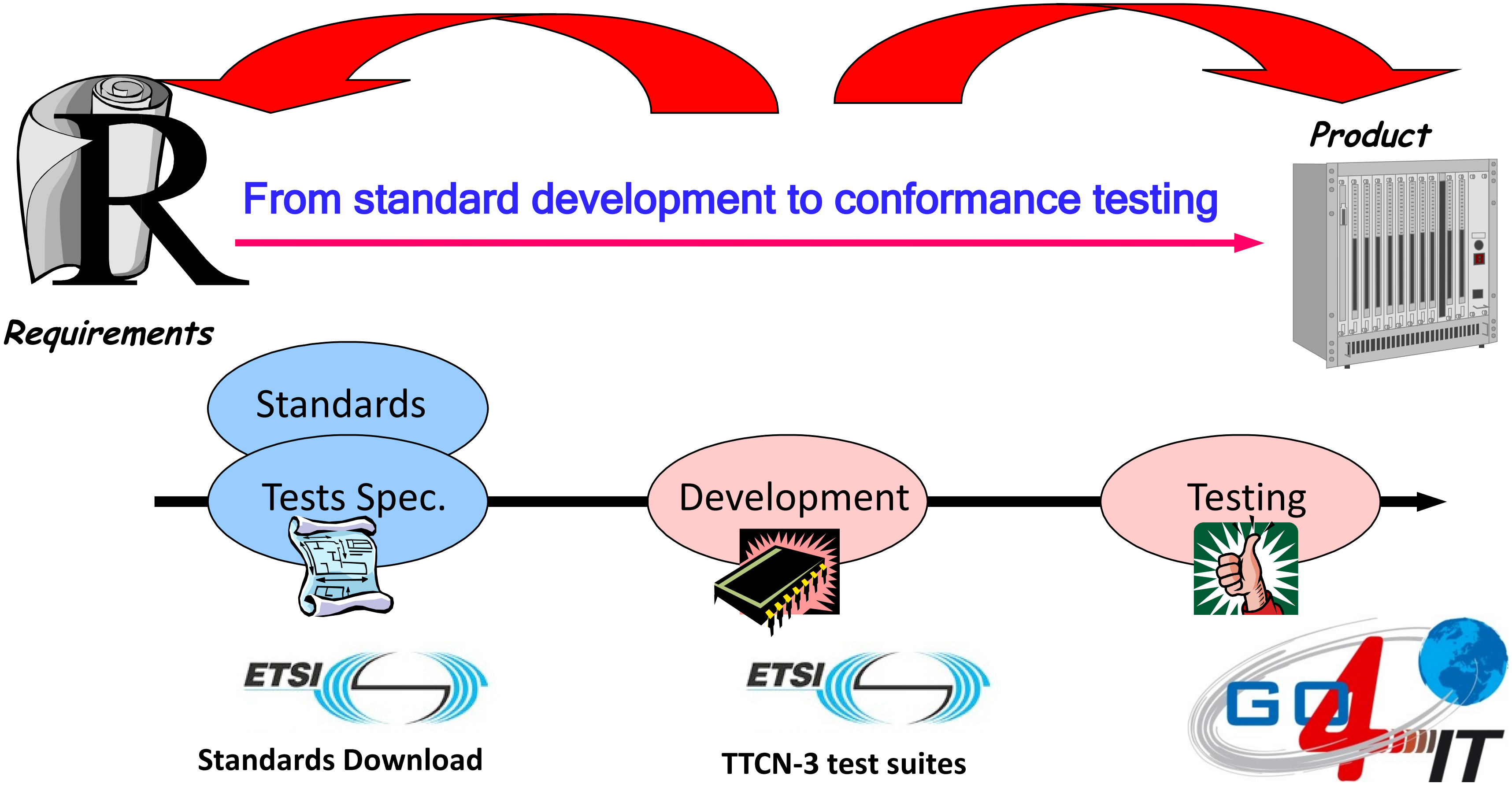
Project identity card

- **Integrated Infrastructure Initiative**
  - Started in Nov 2005
  - 30 month FP6 project
- **Partners:**
  - France (jTest, INRIA)
  - Europe (ETSI)
  - Germany (inno, Fokus)
  - Spain (Cetecom)
  - China (BII, CATR, BUPT, *IMU*)
  - Russia (Ispras)
  - Brazil (IPT)
  - Uruguay (Inco – fing - UdelaR)
- **Go4IT project provided Research Infrastructure users with free *TTCN-3* based *IPv6* testing environment including *test tools, test suites* and the related *services*.**

inno
ETSI
INRIA
RENNES
AT4 Wireless
FOKUS
Fraunhofer Institut für Offene Kommunikationssysteme
ISP RAS
jTest
信息产业部电信研究院
China Academy of Telecommunication Research of MII
UNIVERSIDAD DE LA REPUBLICA URUGUAY
BII
天地互连
INNER MONGOLIA UNIVERSITY
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS
北京邮电大学
IPT
Instituto de Pesquisas Tecnológicas

Concept

Fill product to market gap



From standard development to conformance testing

Requirements

Product

Standards

Tests Spec.

Development

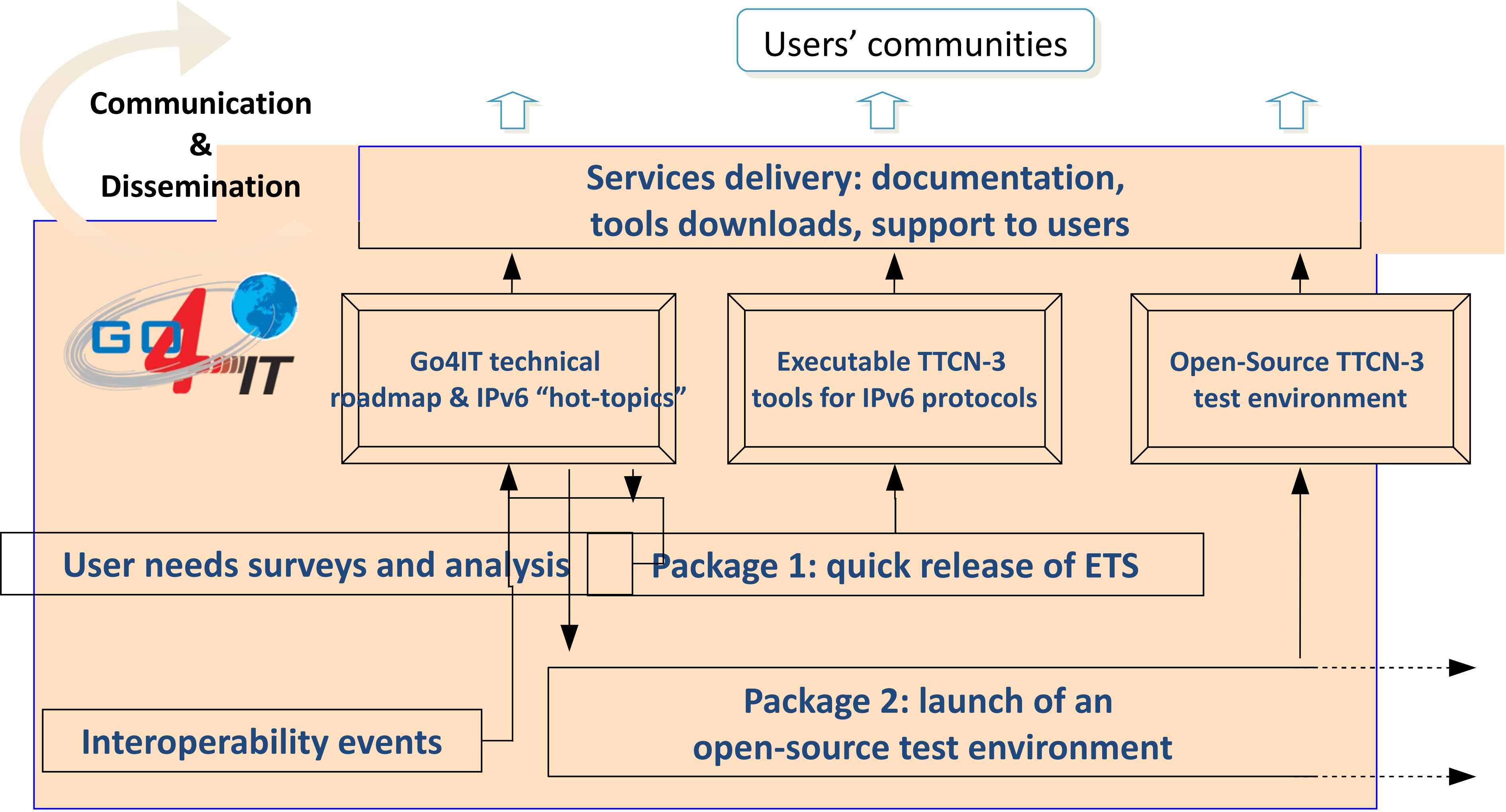
Testing

Standards Download

TTCN-3 test suites

TTCN-3 test platform

Go4IT Approach



Users’ communities

Communication & Dissemination

Services delivery: documentation, tools downloads, support to users

Go4IT technical roadmap & IPv6 “hot-topics”

Executable TTCN-3 tools for IPv6 protocols

Open-Source TTCN-3 test environment

User needs surveys and analysis

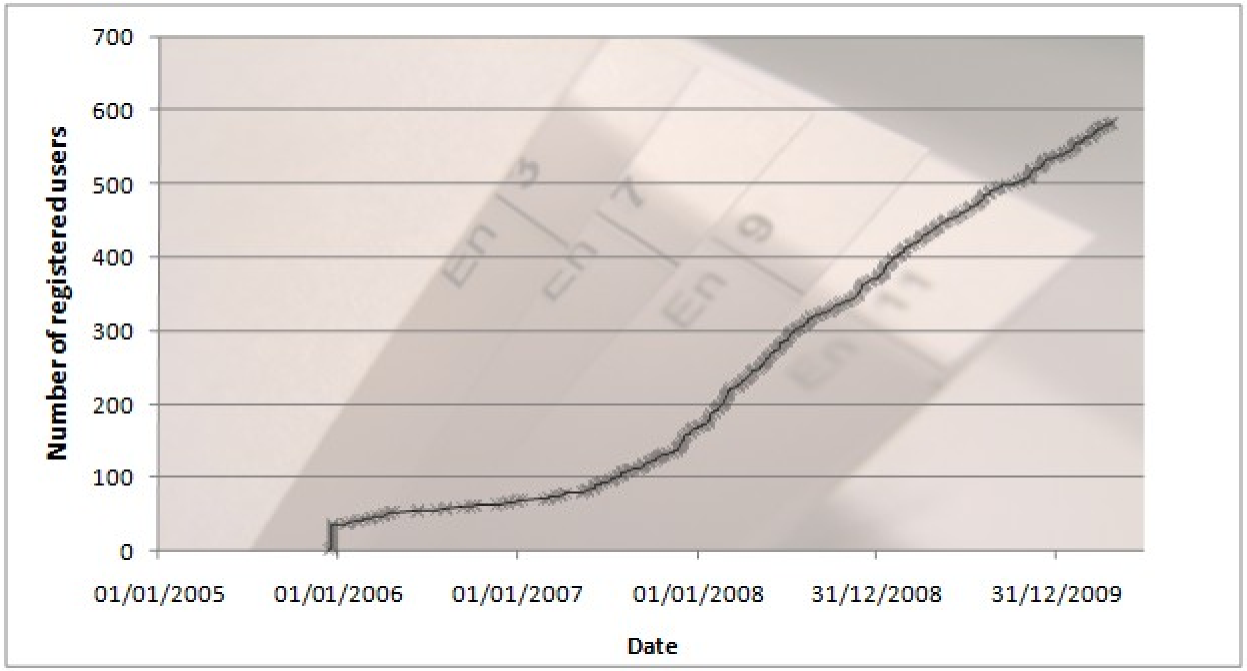
Package 1: quick release of ETS

Interoperability events

Package 2: launch of an open-source test environment

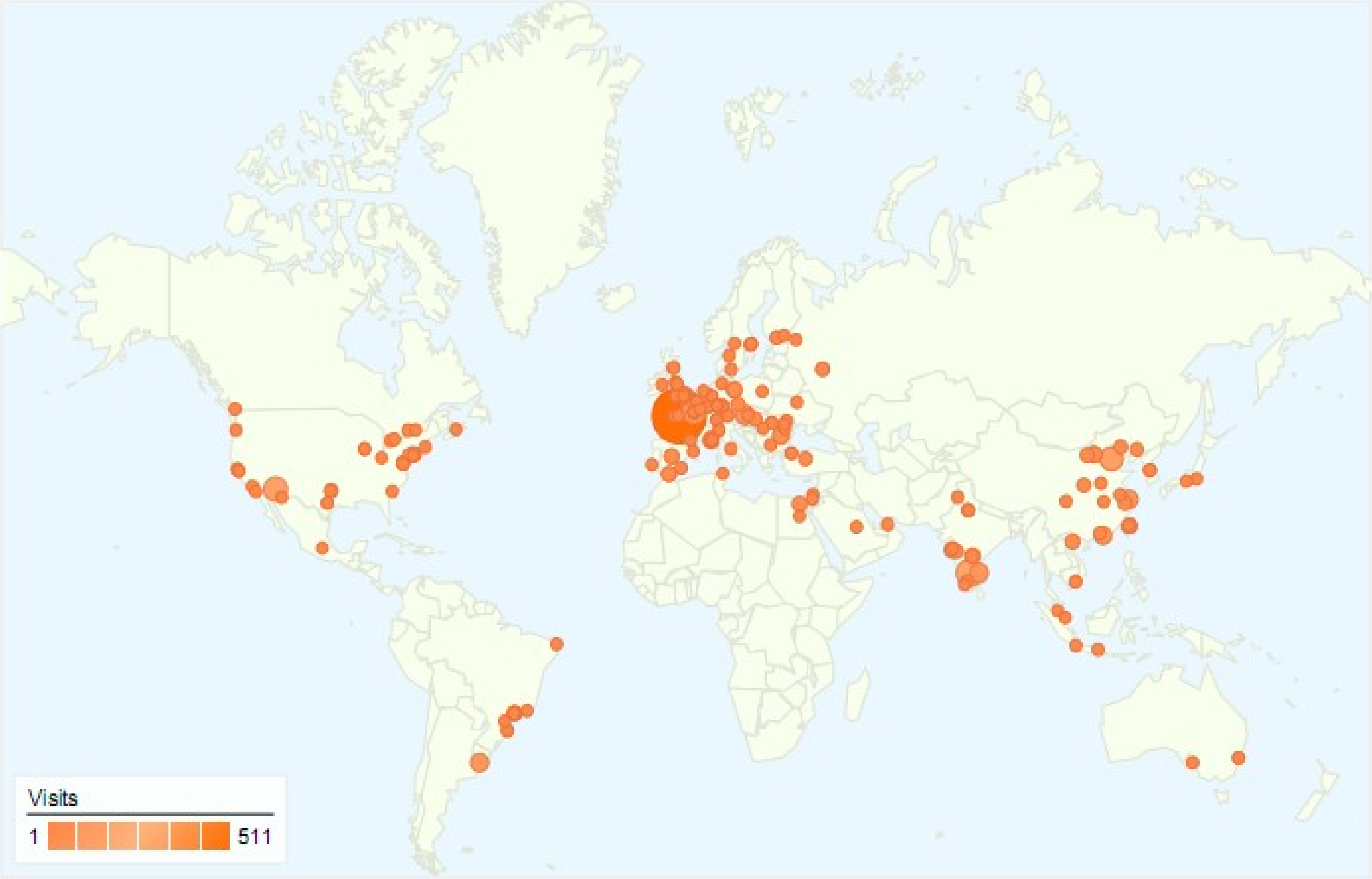
Website Registrations

- http://www.go4-it.eu
- Registration required for software downloads

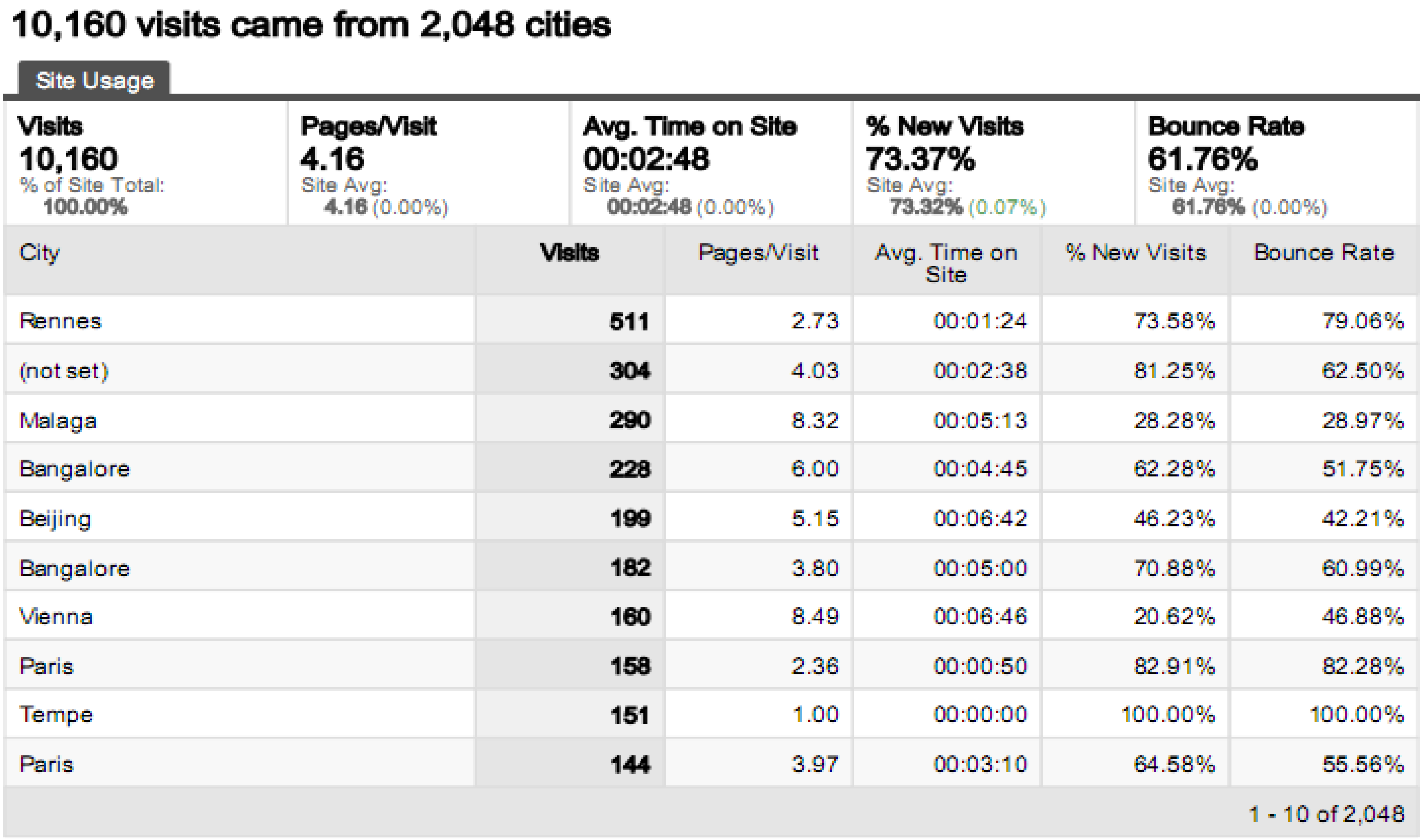


Confirmed interest in solutions provided by Go4IT

Website Visits – Geographical Distribution



Website Visits – City Distribution



10,160 visits came from 2,048 cities

Site Usage

Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
10,160 % of Site Total: **100.00%**	**4.16** Site Avg: **4.16** (0.00%)	**00:02:48** Site Avg: **00:02:48** (0.00%)	**73.37%** Site Avg: **73.32%** (0.07%)	**61.76%** Site Avg: **61.76%** (0.00%)

City	Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
Rennes	**511**	2.73	00:01:24	73.58%	79.06%
(not set)	**304**	4.03	00:02:38	81.25%	62.50%
Malaga	**290**	8.32	00:05:13	28.28%	28.97%
Bangalore	**228**	6.00	00:04:45	62.28%	51.75%
Beijing	**199**	5.15	00:06:42	46.23%	42.21%
Bangalore	**182**	3.80	00:05:00	70.88%	60.99%
Vienna	**160**	8.49	00:06:46	20.62%	46.88%
Paris	**158**	2.36	00:00:50	82.91%	82.28%
Tempe	**151**	1.00	00:00:00	100.00%	100.00%
Paris	**144**	3.97	00:03:10	64.58%	55.56%

1 - 10 of 2,048

Overview & Scope

- free/open compiler
  - GNU GPL & Cecill license
  - gnu technology
- A0 Scope
  - Grammar coverage
  - Technological limitations

Project Organization

- Project Goals
  - Build an open TTCN-3 compiler for Go4IT
  - Integrate the compiler to other Go4IT tools
  - Use the new open compiler for Go4IT specifics needs
- Steps/Modules/Teams
  - How to divide the compiler tasks between distributed teams

Project Organization-cont

- Teams & Tasks
  - Inco/UdelaR, Coordination, Lexical & Syntactical Analysis, Technical Orientation
  - BUPT, Translation
  - IMU, Run Time System
  - INRIA, Coordination & Code QA
  - ISPRAS, Automated Testing
  - Other tasks

Project Organization-cont

- Methodology
  - Modules Isolated development
    - Internal Interface definitions
  - Integration Activities
    - Beijing Integration Activity
    - Close integration remotely
- Coordination
  - InCo/UdelaR and INRIA activities

Parser Step

- InCo/UdelaR
  - Responsible for the Parser Module
  - Build the Analysis Syntactical Tree (AST)
- Tools & Technology
  - Experience & Research
  - Bison/Flex
- Compiler Design
  - Analysis Module
  - Aid tools definition

Parser Step-cont

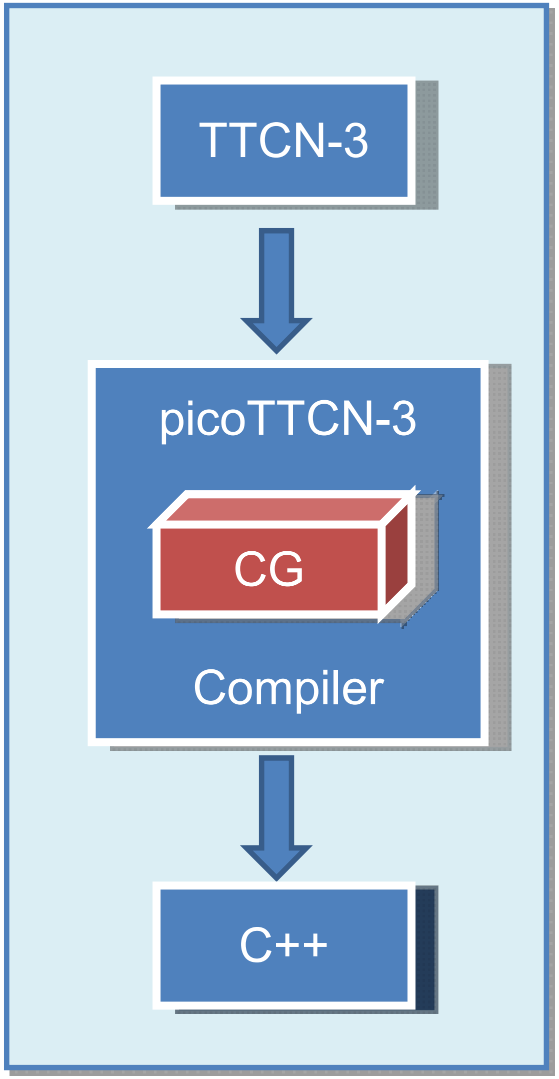
- Implementation Steps
  - Experimenting with a small part of the Language as PoC
    - Selecting the grammar subset
    - Specifying Lexical and Analytical rules
    - Testing the results
    - Build Aid Tools for repetitive tasks
  - The full Parser
    - Scope definition
    - Compiler & Parser Design
    - Methodology, use of Aid Tools built

Parser Step-cont

- Results
  - The Parser ready (ISPRAS Test Cases 816 compiled and 43 rejected)
  - Start Point for the other teams
  - Use for dual compiler research and prototype
- Lessons learned
  - Aid Tools
  - Development Life Cycle

Translation Step (CG)

- Responsible
  - Beijing University of Posts and Telecommunications
  - Code Generator (CG)
- Main task
  - Produces C++ codes based on Abstract Syntactical Tree (AST) and run-time libraries
  - ***Link*** between AST and run-time libraries (T3RTS)
- Tools
  - Developing language: C
  - Debugger: gdb



Translation Step (CG)-cont

First Step

- Write C++ code manually based on T3RTS
- Including data and TTCN-3 specific constructs such as alt, send and receive

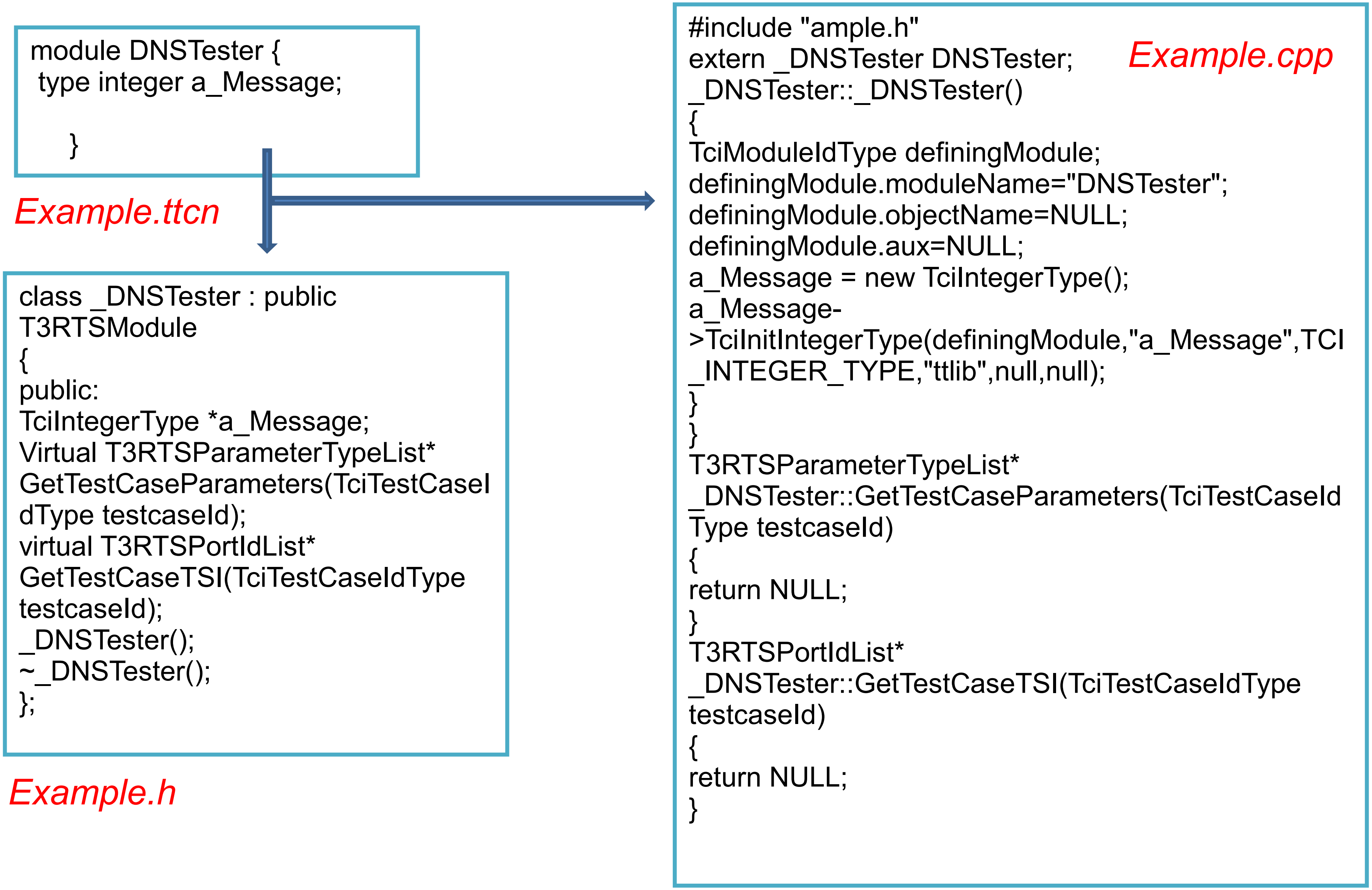
Second Step

- Produce manually designed C++ code through traversing syntax tree
- Passing the information, such as module parameters, to T3RTS

Third Step

- Debug
- Integration with other parts

Translation Step (CG)-cont



```
module DNSTester {
 type integer a_Message;

    }
```

Example.ttcn

```
class _DNSTester : public
T3RTSModule
{
public:
TciIntegerType *a_Message;
Virtual T3RTSParameterTypeList*
GetTestCaseParameters(TciTestCaseI
dType testcaseId);
virtual T3RTSPortIdList*
GetTestCaseTSI(TciTestCaseIdType
testcaseId);
_DNSTester();
~_DNSTester();
};
```

Example.h

Example.cpp

```
#include "ample.h"
extern _DNSTester DNSTester;
_DNSTester::_DNSTester()
{
TciModuleIdType definingModule;
definingModule.moduleName="DNSTester";
definingModule.objectName=NULL;
definingModule.aux=NULL;
a_Message = new TciIntegerType();
a_Message-
>TciInitIntegerType(definingModule,"a_Message",TCI
_INTEGER_TYPE,"ttlib",null,null);
}
}
T3RTSParameterTypeList*
_DNSTester::GetTestCaseParameters(TciTestCaseId
Type testcaseId)
{
return NULL;
}
T3RTSPortIdList*
_DNSTester::GetTestCaseTSI(TciTestCaseIdType
testcaseId)
{
return NULL;
}
```

TTCN-3 Runtime System (RTS)

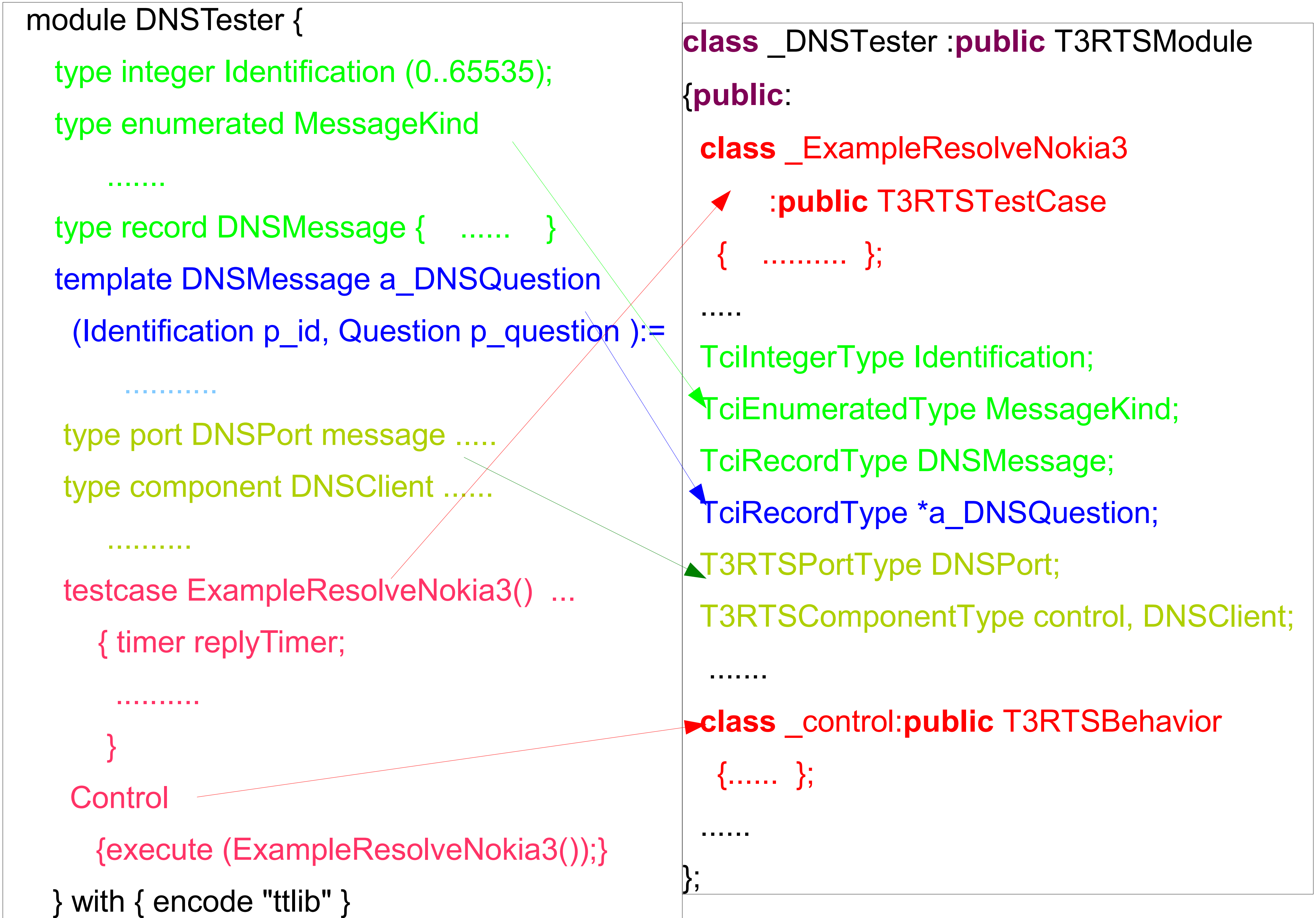
- Responsible
  - IMU
- Main Task
  - Definitions and implementations of internal interfaces in TE
    - Internal interfaces between the T3RTS and ETS
    - Isolate ETS, making the distributing of testing components transparent to ETS
    - Translation step is simplified
  - Definitions and implementations of Type/Value
    - A public service with two functions
      - Unify the components message format
      - Uniform representation for information for the codec
    - Type and value class hierarchy
      - Tci_Type, correspond to ATS types
      - Tci_Value, correspond to ATS values
  - Implementations of TCI/TRI Required

TTCN-3 Runtime System (RTS)-cont

The functions of the main classes in RTS

class	functions
T3RTSModule	Represents a TTCN-3 module. This is an abstract base class which is reimplemented in the generated ETS. The derived classes contain all the TTCN-3 definitions present in the module.
T3RTSComponentType	holding information about a component type
T3RTSPortType	Class holding information about a port type
T3RTSComponent	This class is abstract. Possible implementations are T3RTSSystemComponent and T3RTSLocalComponent.
T3RTSLocalComponent	It is used for the Control Component, MTC and the PTCs run by the local runtime system
T3RTSBehavior	It is a base class and the actual behaviors defined in ATS should be derived from it.
T3RTSTestCase	ETS should define the behavior classes described in ATS by deriving from this class directly or indirectly.
T3RTSPort	This class is abstract. Possible implementations are T3RTSSystemPort and T3RTSLocalPort.
T3RTSTimer	The class defines the timer in ATS
T3RTSMessage	TTCN-3 messages can be represented in two ways: either with an abstract TTCN-3 value or with a binary string .

RTS using the sample



```
module DNSTester {
   type integer Identification (0..65535);
   type enumerated MessageKind
        .......
   type record DNSMessage {    ......    }
   template DNSMessage a_DNSQuestion
     (Identification p_id, Question p_question ):=
          ...........
    type port DNSPort message .....
    type component DNSClient ......
          ..........
    testcase ExampleResolveNokia3()  ...
        { timer replyTimer;
           ..........
          }
     Control
        {execute (ExampleResolveNokia3());}
   } with { encode "ttlib" }
```

```
class _DNSTester :public T3RTSModule
{public:
  class _ExampleResolveNokia3
          :public T3RTSTestCase
   {    ..........  };
  .....
  TciIntegerType Identification;
  TciEnumeratedType MessageKind;
  TciRecordType DNSMessage;
  TciRecordType *a_DNSQuestion;
  T3RTSPortType DNSPort;
  T3RTSComponentType control, DNSClient;
   .......
  class _control:public T3RTSBehavior
   {......  };
  ......
};
```

Automated Testing

- ISPRAS
- Grammar based testing
  - Tool for generate test cases
- Test Cycles
- Testing Results
  - 816 compiled and 43 rejected
  - For the 43 rejected 40 are grammar variations that are not supported in A0, the rest are empty constructions unused.

Today Status of picoTTCN-3

- Final Status of the project
  - A0 operative
  - DNSTester running
- dual API for platform interoperability
  - picoTTCN-3 was used to experiment with the dual platform solution

Main Lessons Learned

- Distributed and Multicultural teams
  - Communication
  - Formality
  - Methodology
  - Standard processes and documents
- Extended tools to build compilers
  - Tools for transform BNF into Bison and Flex specifications

Further Activities

- Search of new partners
  - To finish the open points
  - To use the compiler for research proposes
  - To use the compiler for educational proposes
- Pending activities
  - Finish OO approach
  - Finish grammar rules

Thank you!
rrezzano@fing.edu.uy