SCIENCE*SPRING*DAY

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

DEPARTAMENTO DE INFORMÁTICA

Correct and Secure Global Software Infrastructures via Logics and Types

SOFTWARE SYSTEMS / PLASTIC Team

CENTER FOR INFORMATICS AND INFORMATION TECHNOLOGIES

Jorge A. Pérez

(Postdoc)

PhD in Computer Science,
Univ.of Bologna, Italy (2010)

Objectives

How can we detect and eliminate software bugs? Modern computing systems are based on Web services, which rely on **complex communication protocols**. Ensuring that these interacting programs have no bugs is **very challenging**.

Our goal is to produce **tools** that help software developers in verifying that communicating programs are **correct**. In particular, we focus on **type systems**, a technique used to detect errors in programs before they are executed.

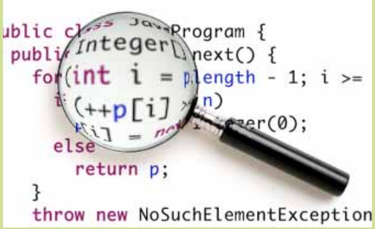
We investigate how **logics** can help us in developing more precise type systems. We will be able to eliminate subtle programming errors, and to enforce enhanced **correctness properties**.

Methodology

We develop our type systems on top of **process calculi**, small programming languages which capture essential aspects of **concurrent, interacting programs**.

Process calculi provide an **adequate framework** to write sophisticated programs and, more importantly, to **formally reason** about their correctness and security properties.

By building upon process calculi foundations, **transferring** our logic-based type systems techniques to conventional programming languages becomes feasible.
In this way, **rigorous mathematical foundations** guide the development of effective, practical techniques for programmers.



```
ublic class JavaProgram {
 public Integer[] next() {
   for(int i = p.length - 1; i >=
     if (++p[i] > n)
       p[i] = new Integer(0);
     else
       return p;
   }
   throw new NoSuchElementException
```

Expected Results

1. New high--level **specification languages** suited to represent communication-based programs. These languages should be sufficiently formal so as to admit tractable analysis (using type systems, for instance) but also sufficiently concrete so as to provide **realistic programming abstractions**.

2. New **type systems** for the specification languages described above. Based on logical principles, such type systems will be able to specify and enforce **security and correctness properties**.

3. Prototype **implementations** of tools allowing the specification, programming and verification of modern distributed systems.

Funding:

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Pest-OE/EEI/UI0527/2011
SFRH/BPD/84067/2012