




# UniTestK: Testing Technology Based on Formal Specification

ISP RAS

December 14<sup>th</sup>, 2005



# Testing of Operating Systems

- n Conformance testing
  - n Functional testing
  - n Interoperability testing
- n Load testing
  - n Stress testing
- n Performance testing
- 

# UniTesK: Recent Projects


- n Real-time OS testing - 1994-2000
- n Testing of IPv6 implementation - 2001-2003
  - Microsoft Research IPv6
  - Mobile IPv6 (Windows CE 4.1)
  - Oktet (St. Petersburg )
- n Intel's optimizing compiler testing - 2001-2004
- n Standards IPMP (MPEG-2, MPEG-21) - 2004
- n Java platform - 2005
- n Billing system (VimpelCom) - 2005
- n POSIX compatible RT operating system - 2005
- n Simulink optimizer (Daimler Chrysler) - 2005
- n Pilot projects - 2003 - 3005
  - A part of real-time system (GosNIIAS)
  - Banking software (Luxoft)
  - Tiny OS (sensor network) (Intel/Berkeley Univ.)






# UniTesK Domain

- n Interfaces are stable enough
- n There is necessity in high reliability

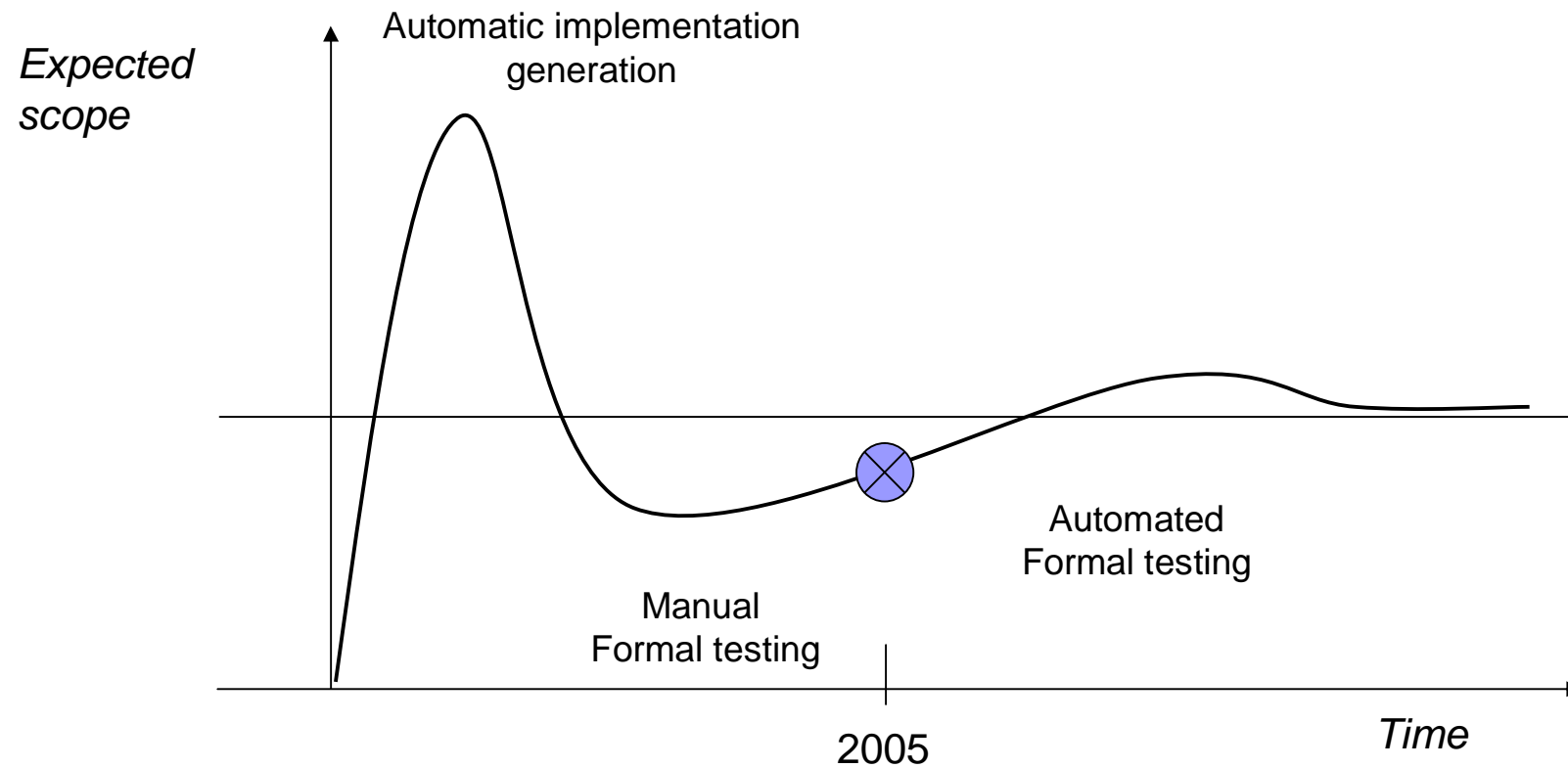
- n Evolution of the interfaces is expected
  - n Appearance of several clones and modifications is expected
- 



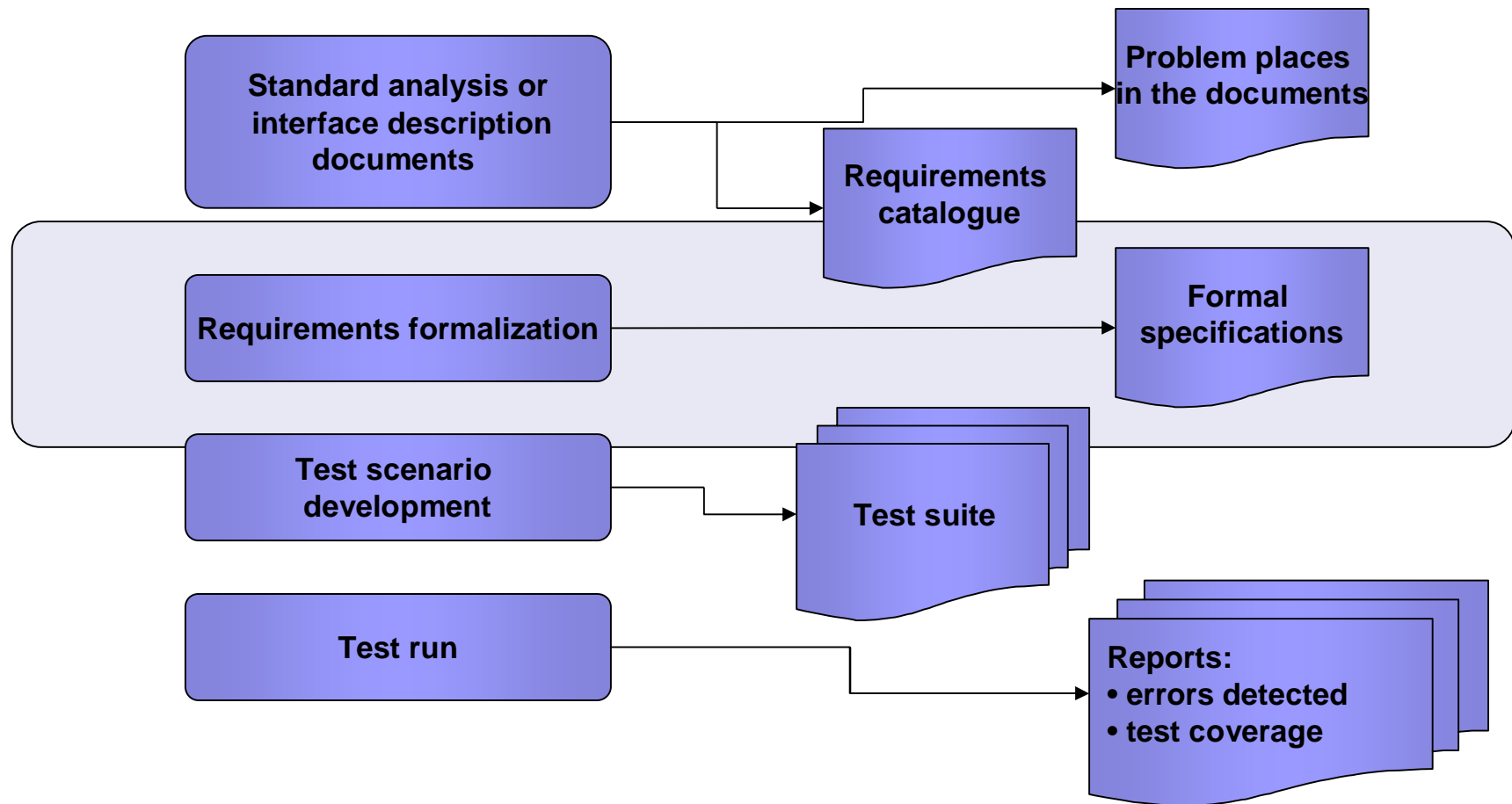
# Evolution of Formal Specs Treatment

- n Accurate, unambiguous requirements definition (possibly, machine readable)
  - n Source for implementation generation
  - n Means for proof of correctness
  - n Source for automatic test generation
- 

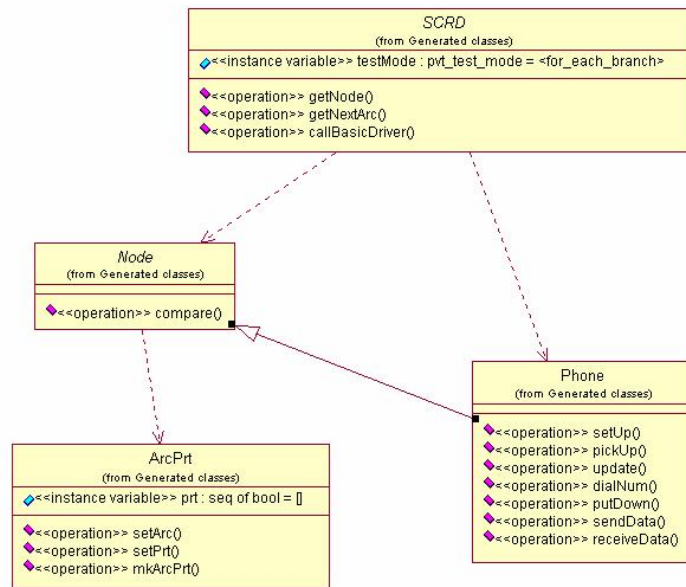
# Evolution ... (2)



# UniTestK Process

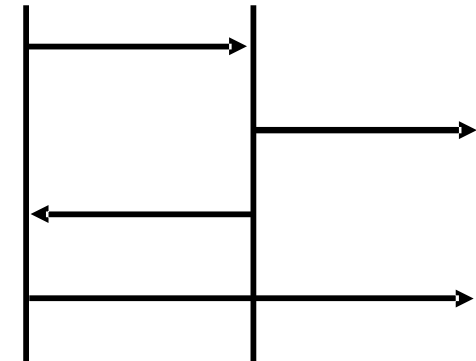


# Kinds of Specifications

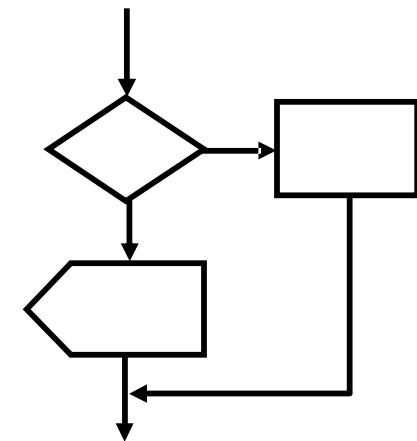



**UML**

**MSC**



**SDL**






# An Example: C Function `day_of_week`

```
int
day_of_week (int tday, int tyear, rc * rc) {

    if( tyear < 0 || tday <= 0 || tday > 366 ||
        ( tday == 366 && is_leap( tyear ) ) ) {
        *rc = nok;
        return 0;
    } else {
        *rc = ok;
        return
            ( days_after_initial_year( tyear, tday )
              + initial_day_of_week )
            % days_in_week;
    }
}
```



# «Semiformal» Specification

## Requirements to input parameters

tyear > 0  
**and**  
tday > 0  
**and**  
tday ≤ 366  
**and**  
tday ≈ 366  
**or**  
tyear is a leap year

## In case of incorrect input (**BRANCH** "Bad parameters"):

Result must be 0  
**and**  
Return code rc = NOK

## In case of correct input (**BRANCH** "OK"):


Result must be equal to residue of sum of function «number of days after initial year» value and «weekday of the initial year» divided by number of day «tday»

**and**  
Return code rc = OK



# Specification in RAISE Language

```
DAY_OF_WEEK : INT >< INT ---> RC >< WEEKDAY
DAY_OF_WEEK( tday, tyear ) as ( post_rc, post_Answer )
post
  if    tyear <= 0 V tday <= 0 V
        tday > 366 V tday = 366
         $\wedge$   $\sim$ a_IS_LEAP( tyear )
  then
    BRANCH( bad_param, "Bad parameters" );
    post_Answer = 0  $\wedge$  post_rc = NOK
  else
    BRANCH( ok, "OK" );
    post_Answer = (a_DAYS_AFTER_INITIAL_YEAR(tyear, tday ) +
                  a_INITIAL_DAY_OF_WEEK ) \
    a_DAYS_IN_WEEK  $\wedge$  post_rc = OK
  end
```





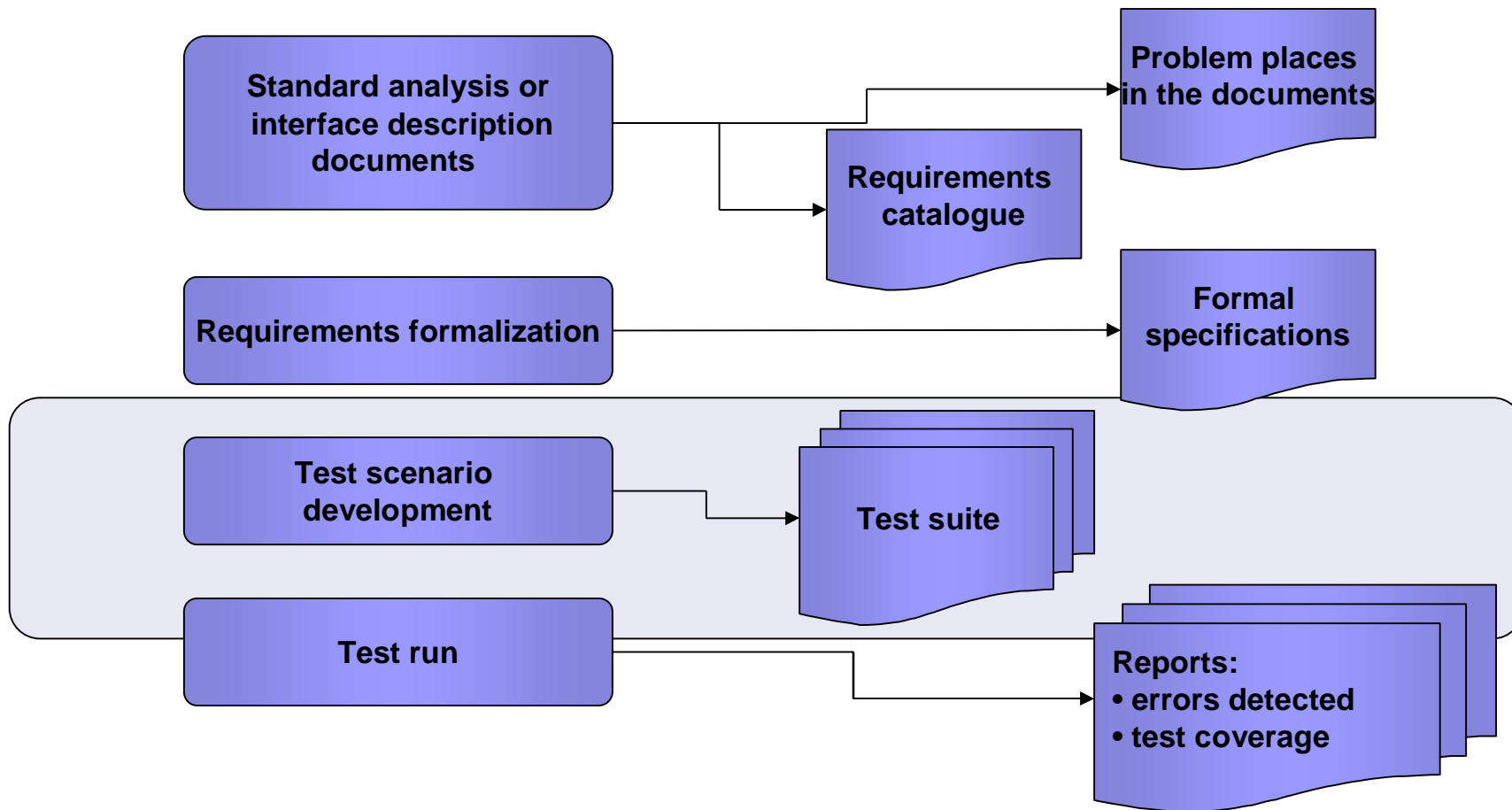
# Example from POSIX. Specification in C extension (SeC)

```
specification void* realloc_spec( void *ptr, size_t size) {  
post {  
    if (ptr != NULL) {  
        if (size > 0) {  
            if (realloc_spec != NULL) {  
                // The contents of the object shall remain unchanged up to the lesser of  
                // the new and old sizes.  
                return is_unchanged( old_value, realloc_spec, min(old_size,size) )  
                // Each such allocation shall yield a pointer to an object disjoint from  
                // any other object.  
                && ( (ptr != realloc_spec) => is_disjoint_object(  
remove_Chunk(@memory,ptr), realloc_spec, size ) );  
            } else /* realloc == NULL */ {  
                // If the space cannot be allocated, the object shall remain unchanged.  
            }  
        }  
    }  
}
```

• • •

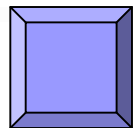
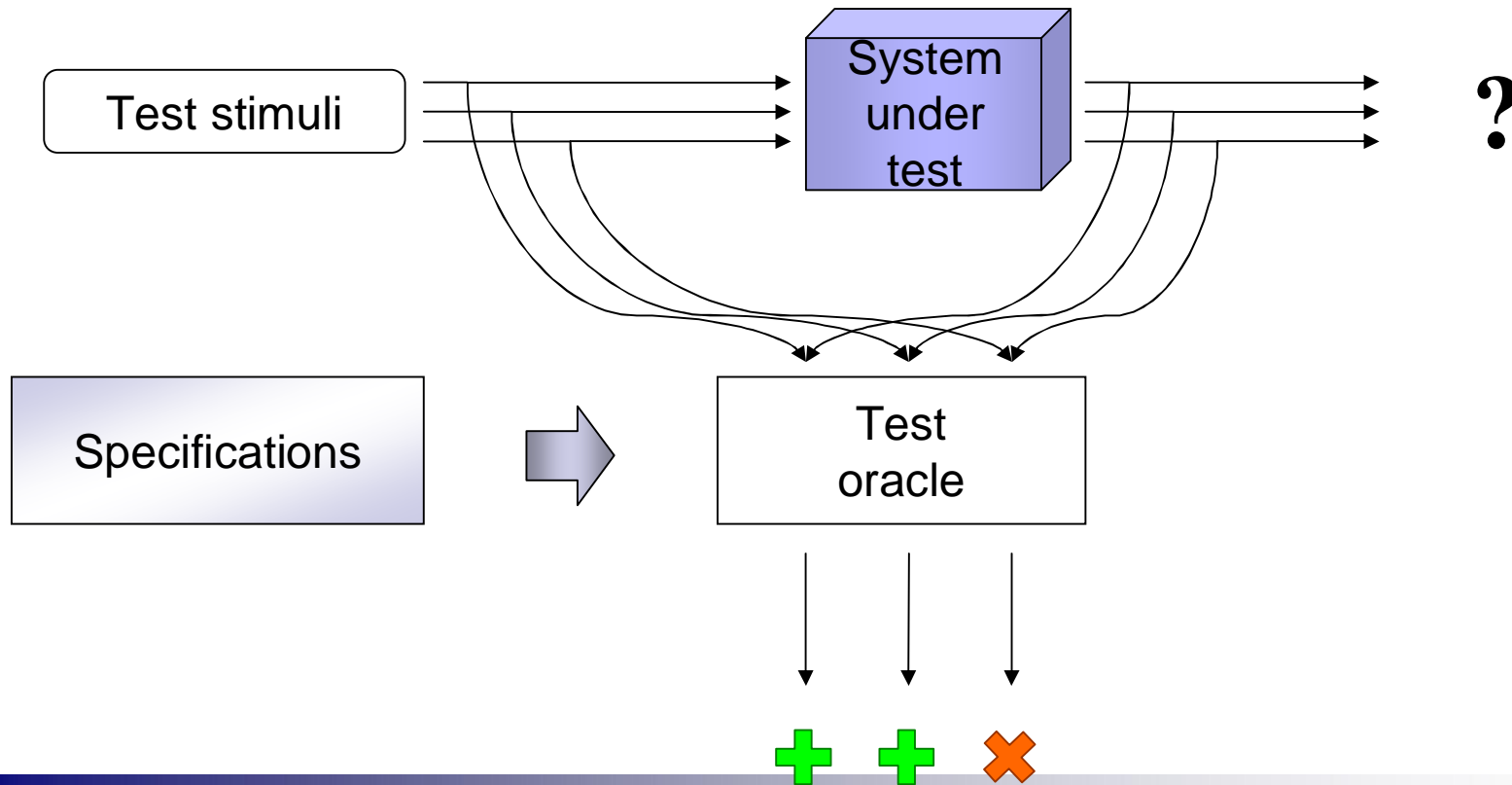


# UniTestK Process



# Test Oracles

Automatic checking of output correctness



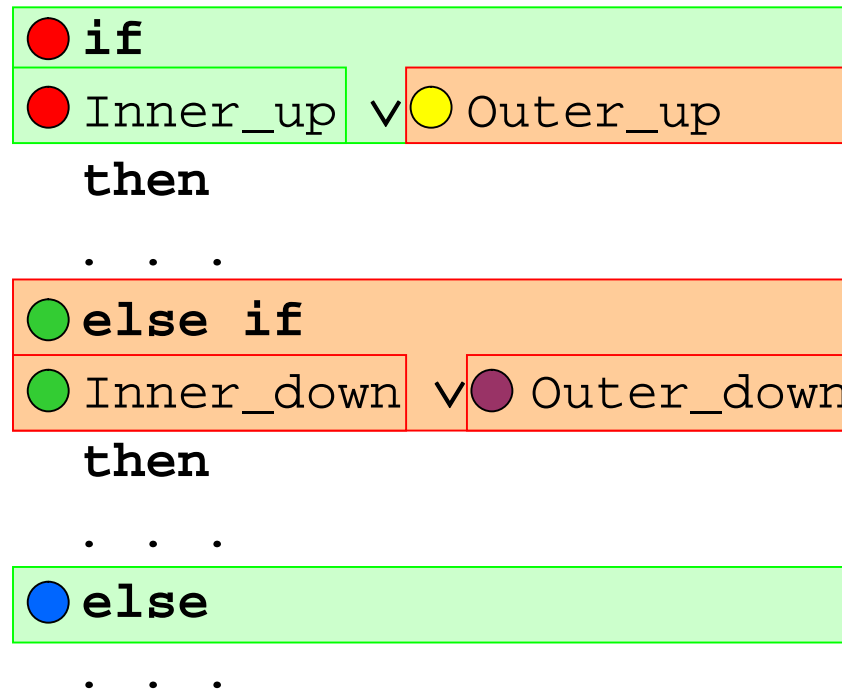
# Test Coverage Criteria

Testing quality measures:

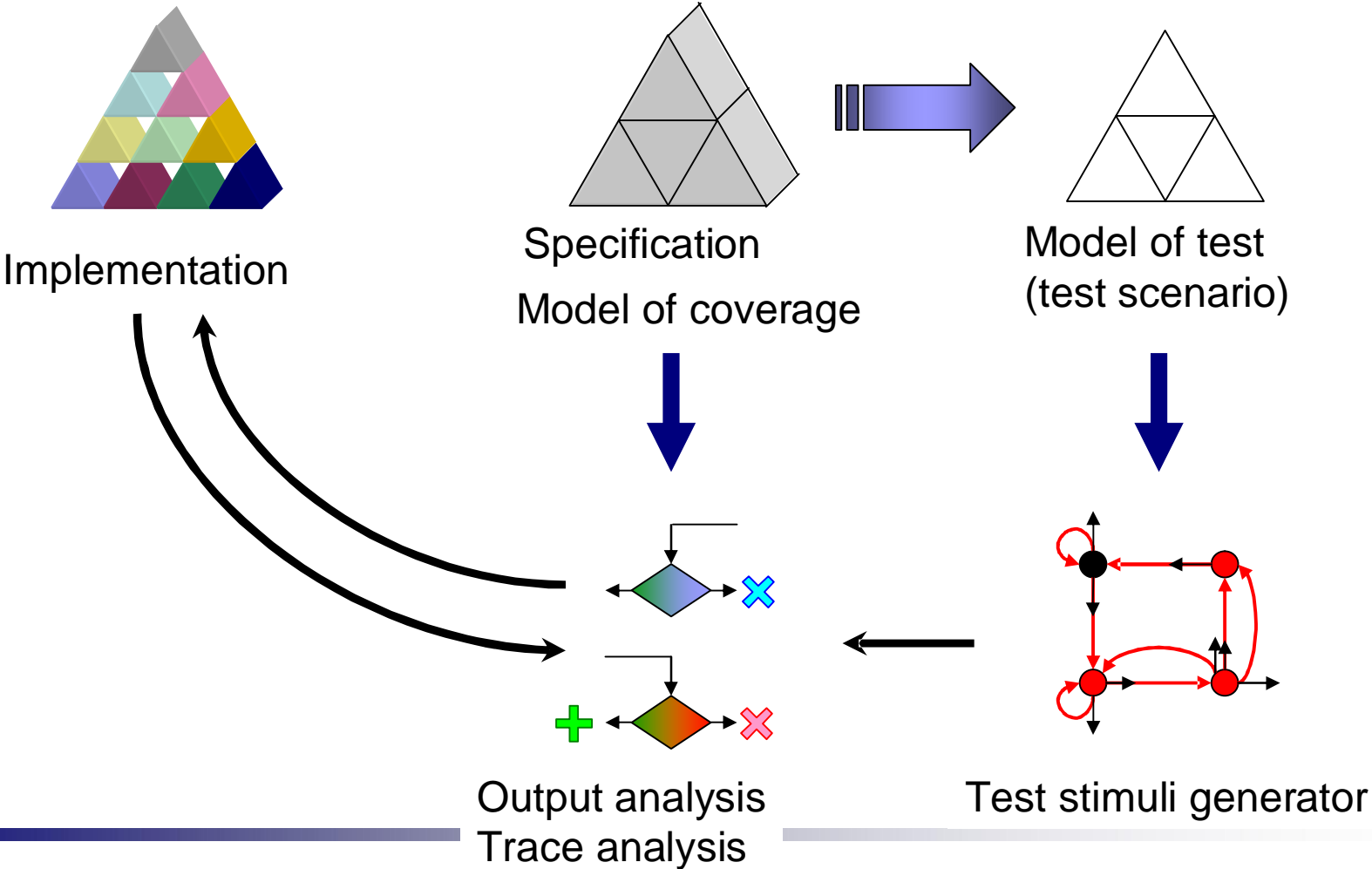
n Coverage of implementation

n Coverage of logical branches in specification

post



# UniTestK Workflow





Forté for Java 4, Community Edition [Project] | J@T Report - D:\JAT\examples\tests\ru\ispras\redverst\se\java\examples\account\model\AccountTestScenario.1062068489519.jatrac...

File Edit View Project Build Debug Versio

Editing GUI Editing Running Debugging

Explorer [Filesystems]

- D:\JAT\examples\tests
- ru
- ispras
- redverst
- se
- java
- examples

J@T Report - D:\JAT\examples\tests\ru\

Trace View Play Window

XML Structure MSC FSM Model

Speed: 0 | Frame: 16 of 2447

ru.ispras.redverst.se.java.examples.account.model.AccountMediator@1479feb : Model ru.ispras.redverst

ru.ispras.redverst.se.java.examples.account.model.AccountTestScenario : Scenario

Start Time: Thu Aug 28 15:01:34 MSD 2003  
Host: aurora/195.208.53.136  
JVM: Sun Microsystems Inc. Java Hot Spot(TM) Client VM 1.4.1-b21  
OS: Windows 2000 x86 5.0

deposit

ru.ispras.redverst.se.java.examples.account.model.AccountSpecification.deposit(int ) (int sum = 1)

prime\_formula: 0 < sum = true

prime\_formula: !( (java.lang.Integer.MAX\_VALUE - sum ) < balance ) = true

precondition\_end:

prime\_formula: 0 < balance = false

prime\_formula: balance == 0 = true

mark: Deposit on empty account

branch: Single branch

deposit( int ) (int sum = 1)

Success

prime\_formula: balance == ( @balance + sum ) = true

prime\_formula: reads java.lang.Integer.MAX\_VALUE = true

Success

Success



# Contacts

n Institute for System Programming of  
Russian Academy of Sciences

<http://www.ispras.ru>

n UniTesK

<http://unitesk.ispras.ru>

n Alexander K. Petrenko

[petrenko@ispras.ru](mailto:petrenko@ispras.ru)

+7-495-912-5317 ext. 4404





Thank You!