Model Based Testing in Linux Standard Base Compliance Program

A.V.Khoroshilov, A.K.Petrenko
{ khoroshilov, petrenko } @ ispras.ru

MBT Users Conference 2011,
Berlin, October 2011

About the Program

- Started in September 2005
- Contractor: Institute for System Programming
- of the Russian Academy of Sciences (ISP RAS), Moscow, Russia
- Sponsors:
  - – Ministry of education and Science of Russian Federation (2005- present)
  - – The Linux Foundation (2007-2010)

ISP RAS

MINISTRY OF EDUCATION AND SCIENCE
OF THE RUSSIAN FEDERATION

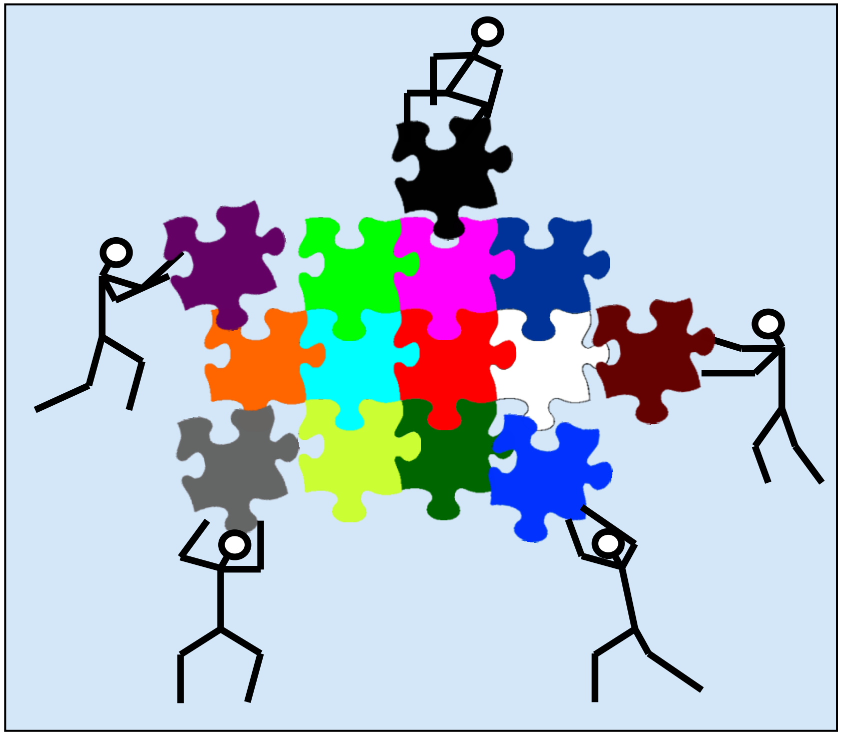
THE LINUX FOUNDATION

LSB Compliance Works

1. Helping industry companies and consortiums to build and evolve **industry-wide API/ABI platforms and open standards** (specifications). Examples of our active projects:
   - Linux Standard Base (LSB)
   - Intel Moblin (Intel/Nokia MeeGo)
   - Google Android
   - Motorola MOTOMAGX
2. Development of **new advanced testing technologies and tools** (including MBT tools: CTESK/UniTESK, API Sanity Autotest)

Constructing Linux Distributions

- There are hundreds (!) of various public Linux distributions.
- A Linux distribution is a collection of various “upstream” components (~500 system ones).
- Components are versioned and can be built in different ways.



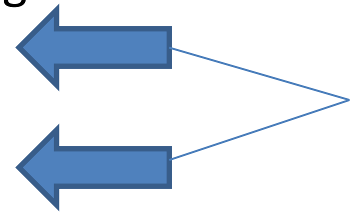
Linux Distribution

Number of Linux Distributions > 500

xandros
LINPUS
freespire
BOSS
turbolinux
redhat
suse
Linspire
sw
SUNWAH
ubuntu
Mandriva
debian
Asianux

No any Universal Solution

- Such a large ecosystem needs several testing approaches and corresponding tools
- 3 grades of test quality
  - Deep
  - Ordinary (“normal”)
  - “Shallow”
- 3 approaches and corresponding tools
  - CTESK / UniTESK for C interfaces
  - T2C
  - API Sanity Autotest



points of MBT interest

LSB Tests Developed by ISPRAS(*)

Testing Grade	# Test Cases	# Interfaces under Test
Deep tests (UniTESK)	> 120 000	~ 1 500 (LSB Core)
Ordinary tests (T2C)	21 000	4 000
“Shallow” test (API Sanity Autotest)	~ 20 700	20 700
Total	> 160 000	25 900

Number of LSB 4.0 Interfaces	~ 40 000
Number of interfaces in a typical Linux distribution	0.5 ÷ 1 million

(*) – Statistics dated at March 2011

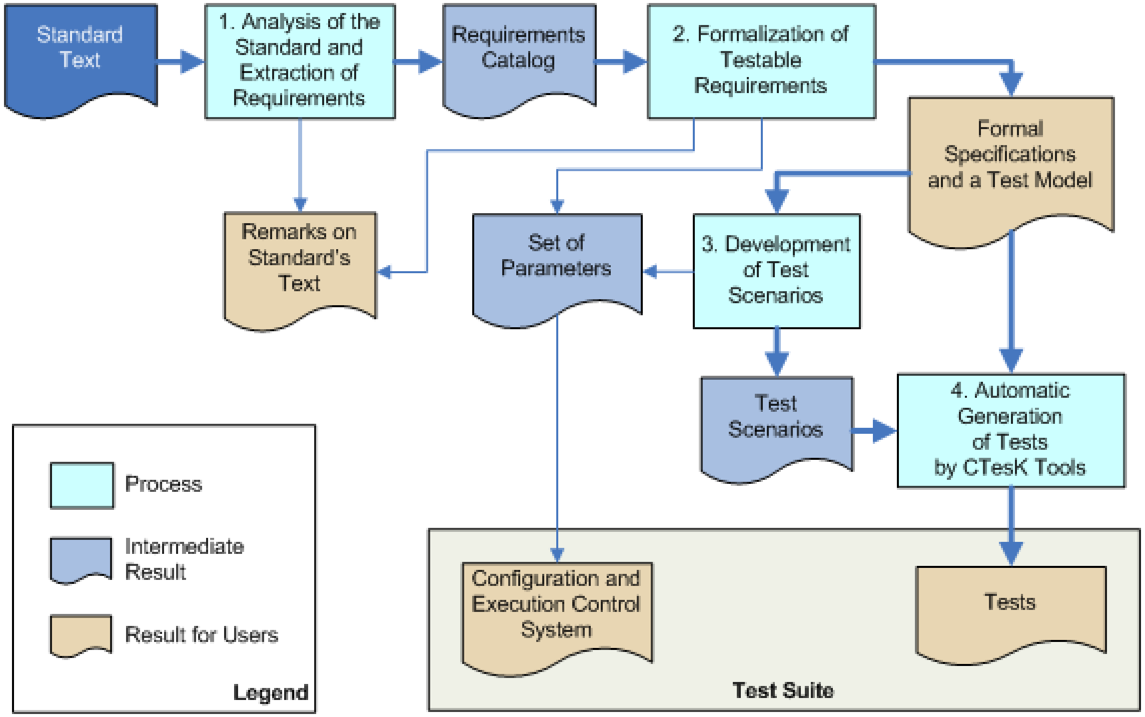
Deep Testing Domain

- LSB Core (extended subset of POSIX)
- ~ 1500 interfaces
- Libraries: libc, C++ run-time
- Groups of functions: memory management, process management, maths, sockets, etc.
- Documentation exists
- Total effort is about 12-15 man*year

CTESK Tool: UniTESK for C Interfaces

- Software contracts specifications (pre/post, invariants)
- Specification Language: SEC (Specification Extension of C)
- Test scenarios describe an input for FSM exploration
- First version of CTESK had been developed in 2000 for IPv6 testing (supported by Microsoft Research)

UniTESK Process


Standard Text
1. Analysis of the Standard and Extraction of Requirements
Requirements Catalog
2. Formalization of Testable Requirements
Formal Specifications and a Test Model
Remarks on Standard's Text
Set of Parameters
3. Development of Test Scenarios
Test Scenarios
4. Automatic Generation of Tests by CTesK Tools
Configuration and Execution Control System
Tests
Test Suite
Process
Intermediate Result
Result for Users
Legend

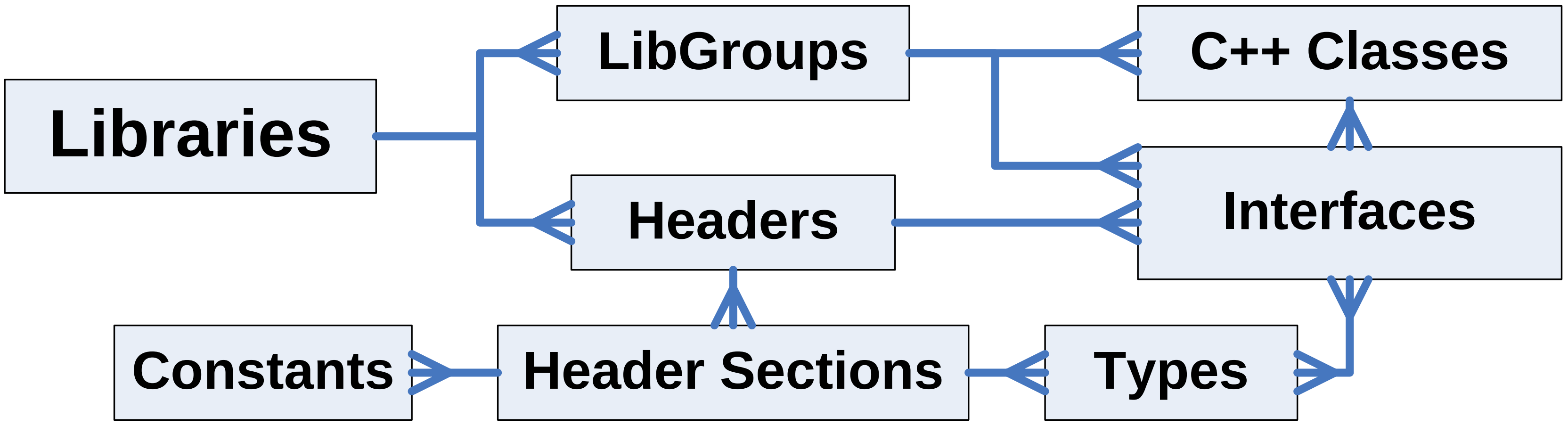
“Shallow” Testing Domain

- Qt3, Qt4 (basic graphics), libxml
- ~ 20 000 interfaces
- Lack of documentation
- Total effort is about 1-2 man*year including effort for tool development

LSB Central Database

- **81** tables with about **25** million records.
- MySQL engine.
- **3** main parts:
  - **Standardized Elements** – structured info about LSB entities such as modules, libraries, headers, classes, interfaces, types, etc.
  - **Linux Ecosystem** – structure of real-world distributions and applications.
  - **Certification Info** – certified product registry and certification workflow support.

LSB DB: A Base for the Model



Libraries

LibGroups

Headers

C++ Classes

Interfaces

Constants

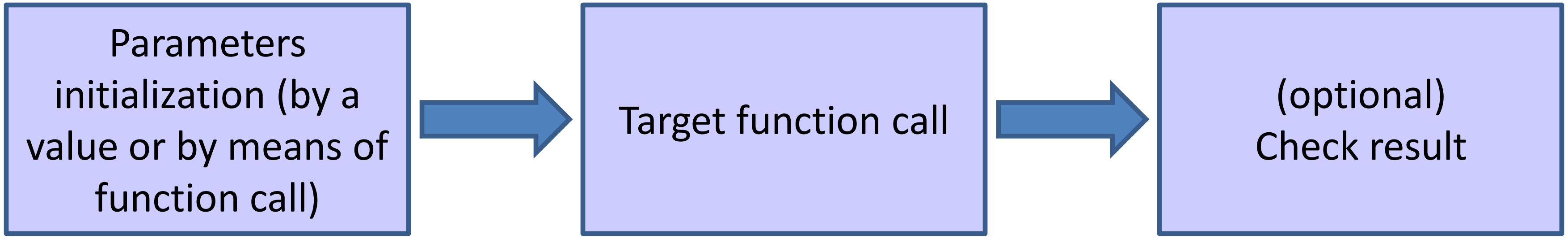
Header Sections

Types

Libraries	44
LibGroups	221
Headers	465
Header Sections	882

Classes	969
Interfaces	36102
Types	4533
Constants	6088

Recursive Scheme:
Test Call Generation



Parameters initialization (by a value or by means of function call)

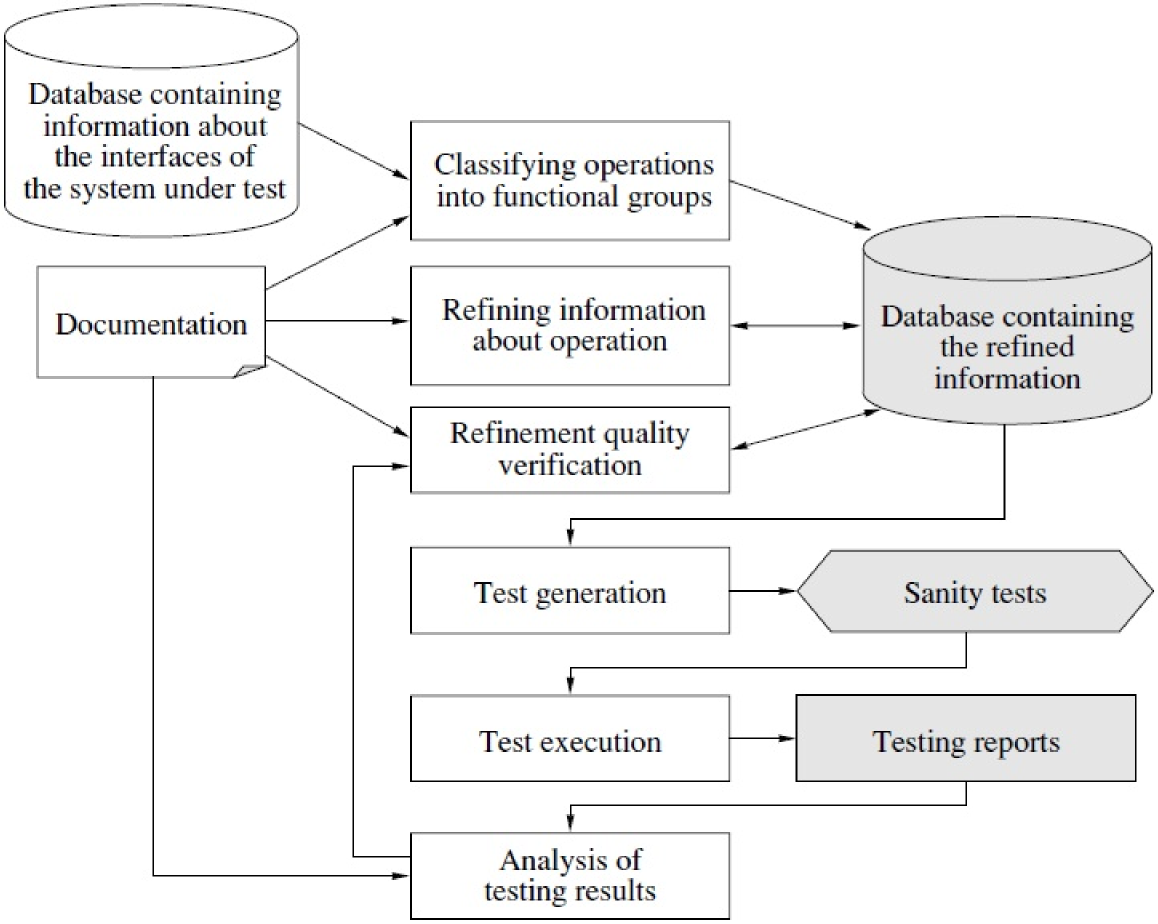
Target function call

(optional)
Check result

Dependencies between parameters types and result types.
Source for dependencies analysis:

- Original types
- Semantically specified types
- Sample values

API Sanity Autotest Process



Database containing information about the interfaces of the system under test
Documentation
Classifying operations into functional groups
Refining information about operation
Refinement quality verification
Database containing the refined information
Test generation
Sanity tests
Test execution
Testing reports
Analysis of testing results

API Sanity Autotesk Usage Statistics

	Number	Original Types
Maximal number of a specialized type uses	513	bool (specialized type for parameters taking the value true)
Number of specialized types used		
> 400 times	3	bool, int
200-399	5	Bool, int, char*, Qwidget*
100-199	16	
10-99	225	
2-9	556	
Total number of specialized types	1665	
Number of uses of specialized types as types of parameters or object calls	11503	
Number of uses of all types as types of parameters or object calls	22757	

ISPRAS: Others MBT Use Cases

- Hardware design unit testing (C++TESK)
- Hardware design system (core) testing (MicroTESK)
- Compiler testing (SynTESK, OTK, Pinery)

See: http://forge.ispras.ru
http://unitesk.com
http://hardware.ispras.ru

Thank you!

http://ispras.ru/~petrenko

http://linuxtesting.org

http://ispras.linuxfoundation.org