Qualips
Quality Platform for Open Source Software

Systematic Technical Interoperability Testing

Dr. Yuri Glickman
Fraunhofer Institute for Open Communication Systems (FOKUS)

November 8, 2010

fOSSa 2010, Grenoble, France

CORDIS

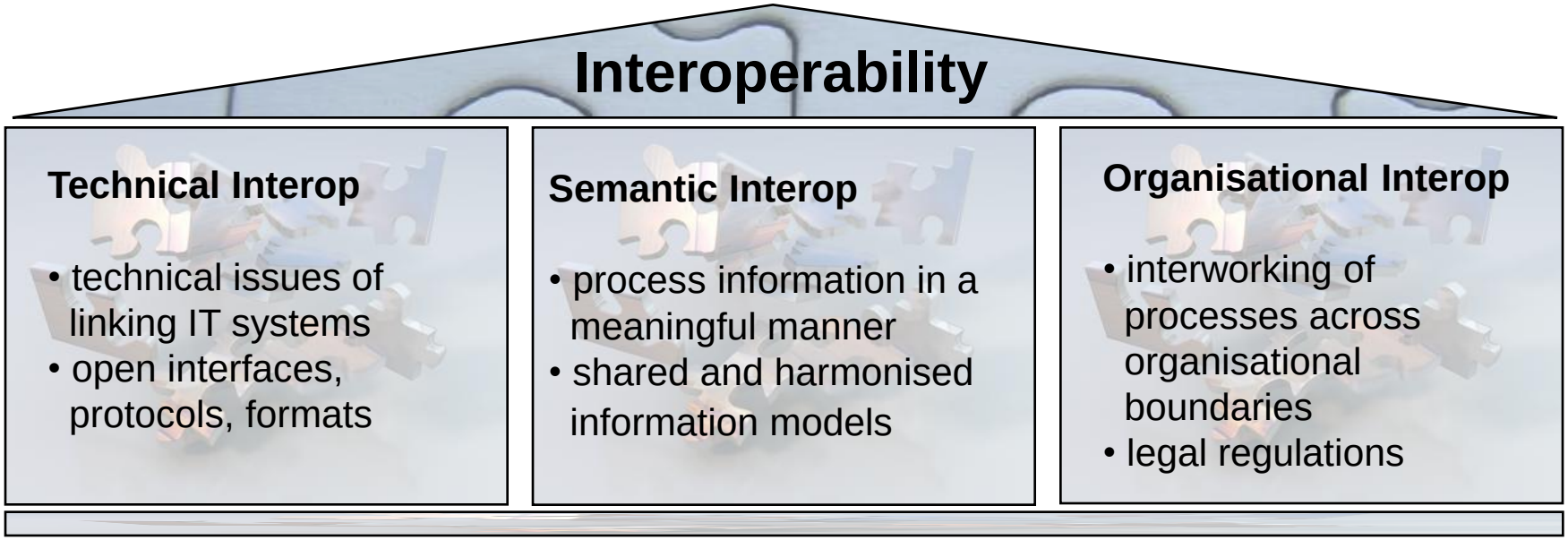
Information Society Technologies

creative commons

Fraunhofer FOKUS

Interoperability in QualiPSo

- QualiPSo – Quality Platform for Open Source Software (IST Integrated Project)
- **One of the objectives:** Foster interoperability of OSS components as well as between open and closed source software



cf. EIF (IDABC)

Motivation – Interoperability and OSS

- High number and diversity of systems, components and services, especially in OSS domain
- Interoperability – a significant quality and trust criterion for Open Source Software
- Achieving interoperability is a significant issue
  - systems are not interoperable by default
  - standards do not assure interoperability as expected
- Interoperability **testing** is the key
- Software interoperability testing is a complex process
- Systematic testing is required to improve the quality

QualiPSo Approach

- A practical “light-weight” approach to software interoperability testing
  - Applicable in OSS communities
  - Addressing non-professional testers and OSS developers

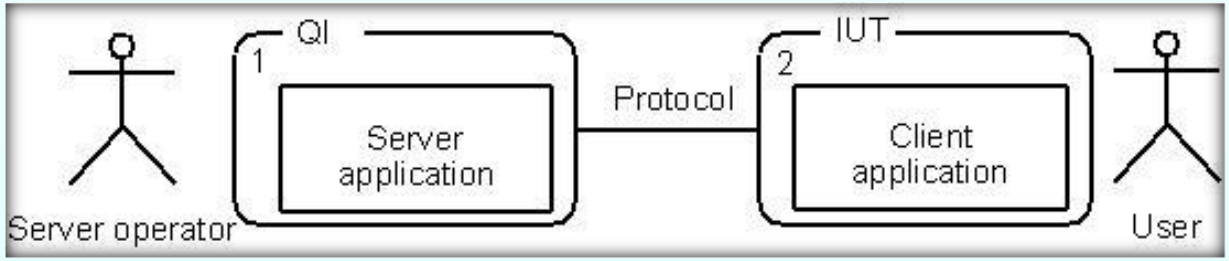
- The approach is based on ETSI’s „Generic Approach to Interoperability Testing”. The approach is simplified and refined:
  - to be more comprehsive for non-proffesional testers
  - to be straightforwardly applicable for testing **software** interoperability

Approach overview

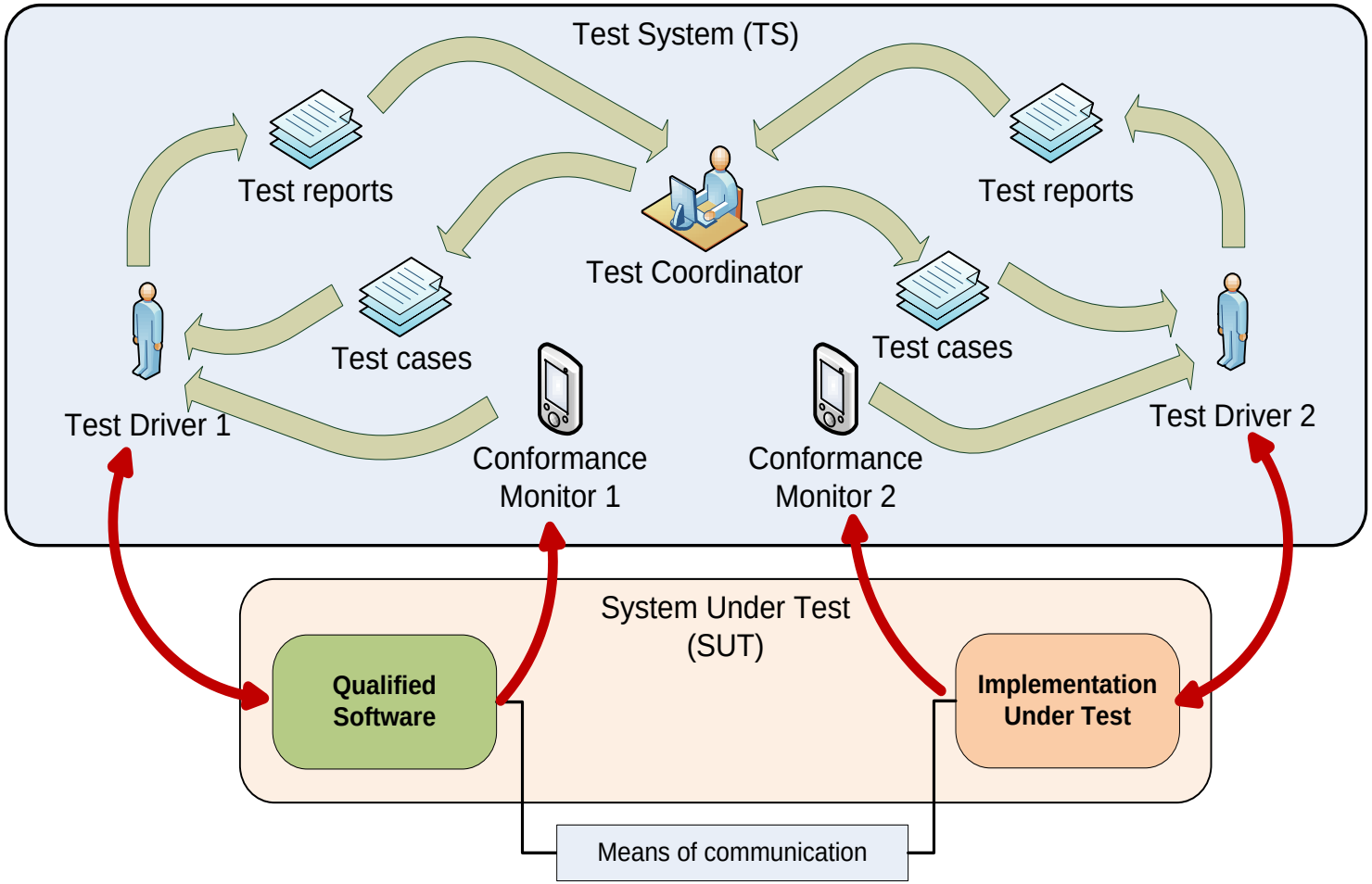
Specify abstract testing architecture

Clearly indicate:

- Implementation(s) under test (IUT) – software systems or components which interoperability has to be verified,
- Qualified implementation(s) (QI) – IUT(s) that have been already tested and can be considered as interoperable,
- Communication channels between components – indicating which components are interconnected,
- Communication protocol or information exchange specification for each channel.



Qualipso abstract interoperability testbed architecture



Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

- Clearly identify the software by providing
  - entity name (used in abstract architecture),
  - full software name,
  - version,
  - supplier,
  - homepage,
  - contact point.
- Gather information for each entity in a separate table containing the listed elements.

Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

- ❑ Meaningful role name
- ❑ Short description
- ❑ Reference to the specification (or standard)
- ❑ Reference to the concrete system implementation in testbed

Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

- functions which must be supported by the IUT(s) to be interoperable with other systems (e.g. “Send a request”)
- ❑ on the basis of the available information exchange specification
- ❑ function name, unique identifier, short description (optionally) and reference to the specification or standard where it is defined
- ❑ is mandatory, optional or conditional (depending on some conditions)?

Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

Develop a test suite structure

- organize the interoperability functions into logical groups:
  - by the role of the entity or
  - by functionality or
  - by test bed configuration or
  - by normal(successful) / exceptiona(unsuccessful) behaviour

Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

Develop a test suite structure

Prepare test purposes

- what must be tested for each identified interoperability function
  - in natural language or
  - in TPLan (for complex test purposes)

Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

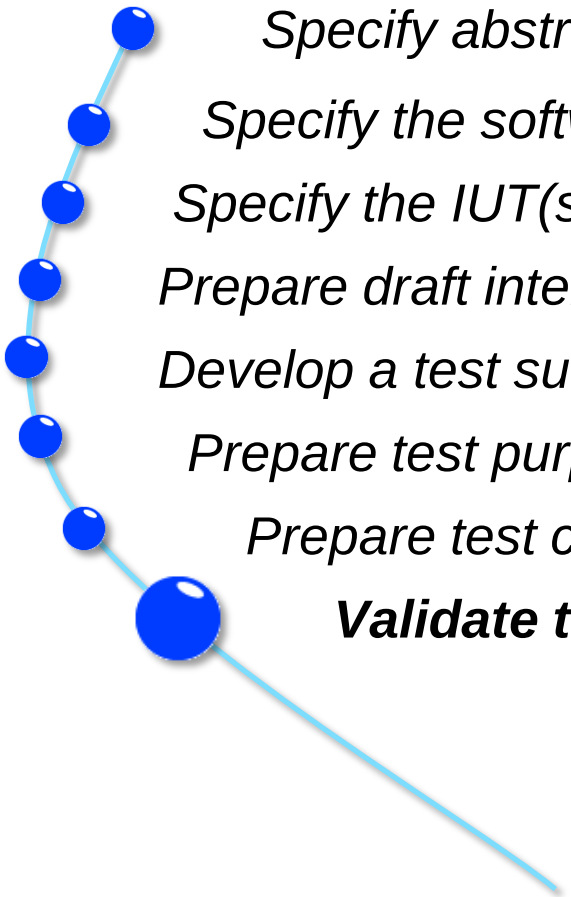
Develop a test suite structure

Prepare test purposes

Prepare test cases

Test case identifier, Summary, Test purpose, Test purpose ID, Testing Architecture ID, Test preconditions, Pre-amble and post-amble, Test steps, Verdict criteria and verdicts

Approach overview



Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

Develop a test suite structure

Prepare test purposes

Prepare test cases

Validate test cases

Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

Develop a test suite structure

Prepare test purposes

Prepare test cases

Validate test cases

Prepare for testing (arrangement and planning)

- ❑ Configure the test bed
- ❑ Prepare the testing plan

Approach overview

Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

Develop a test suite structure

Prepare test purposes

Prepare test cases

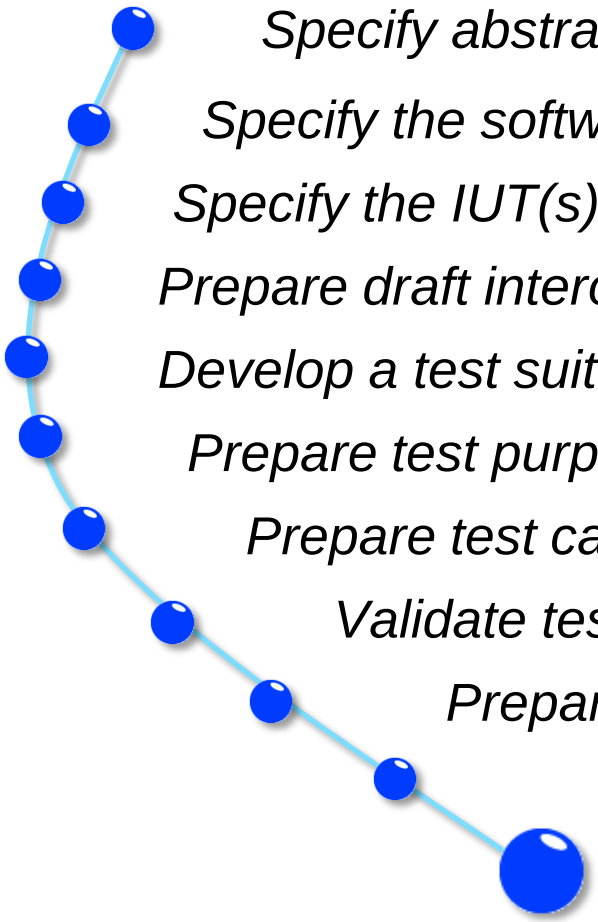
Validate test cases

Prepare for testing (arrangement and planning)

Perform tests

- ❑ Execute test cases
- ❑ Record verdicts

Approach overview



Specify abstract testing architecture

Specify the software systems involved in tests

Specify the IUT(s) roles

Prepare draft interoperability functions

Develop a test suite structure

Prepare test purposes

Prepare test cases

Validate test cases

Prepare for testing (arrangement and planning)

Perform tests

Make a test report

The approach characteristic

- Holistic approach
- Clear terminology
- Practical instructions for execution
- Minimal documentation and clear templates
- All steps illustrated with examples for one scenario
- Recommendations on OSS tools to use
- Possibility for automation

www.qualipso.org

or

German QualiPSo Competence Centre

Yuri Glickman
yuri.glickman@fokus.fraunhofer.de