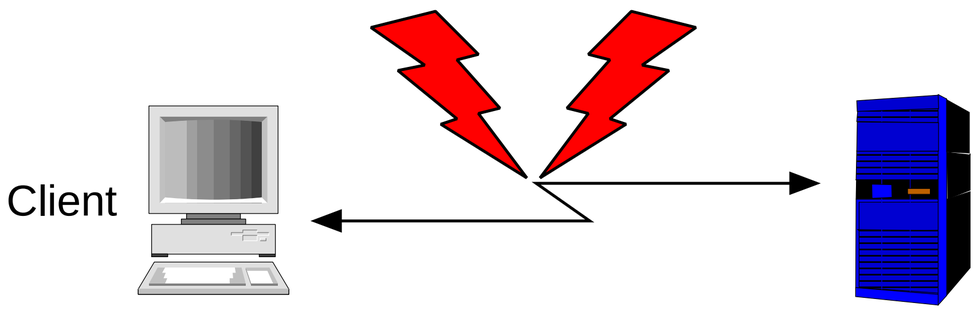
National Institute of Advanced Industrial Science and Technology
AIST

Cyrille Artho, Etsuya Shibayama, RCIS/AIST, Tokyo, Japan (c.artho@aist.go.jp)
Armin Biere, Johannes Kepler University, Linz, Austria
Shinichi Honiden, National Institute of Informatics, Tokyo, Japan

Research Center for Information Security

Enforcer: Fault injection for Java/JUnit

Problem

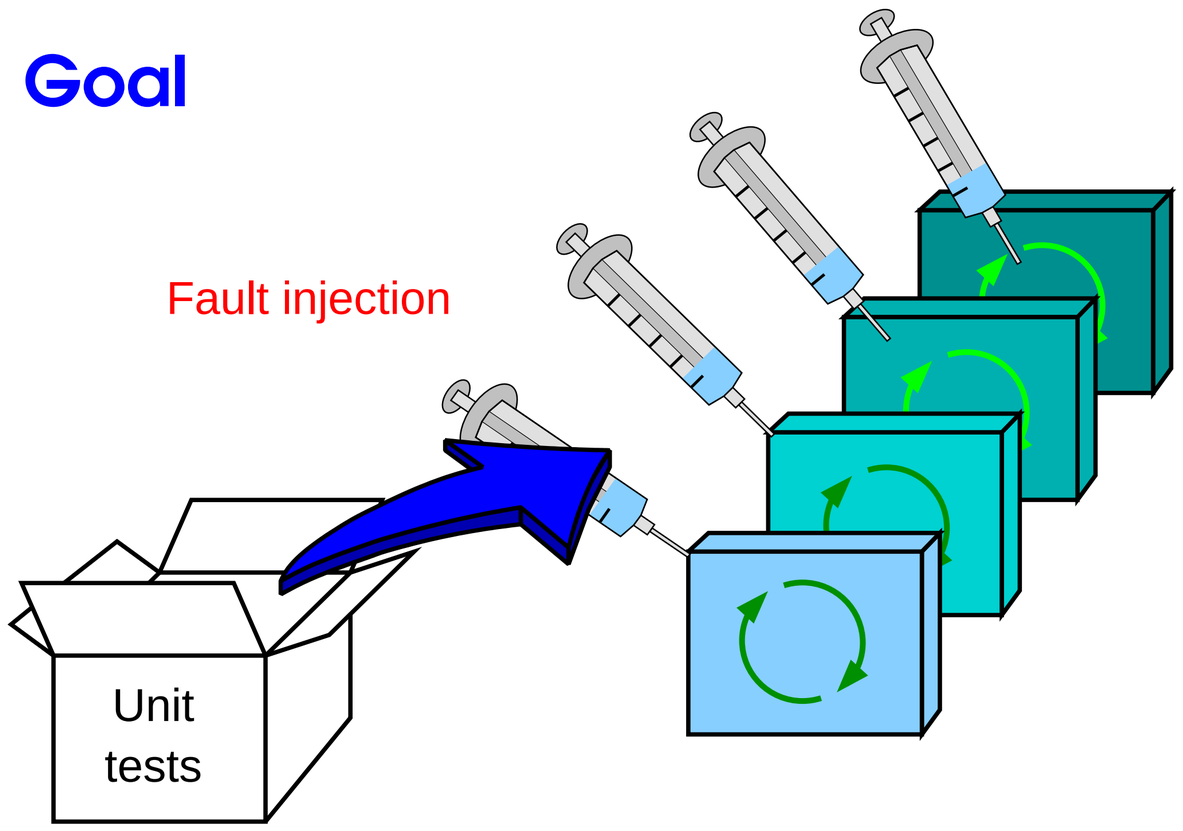


* Any I/O operation may fail.
* Failure results in an exception.
* How to test exceptions systematically?

Fault injection

* Simulates hardware failures in software.
* Can simulate network problems.
* Too expensive to execute large test suite against each possible failure.

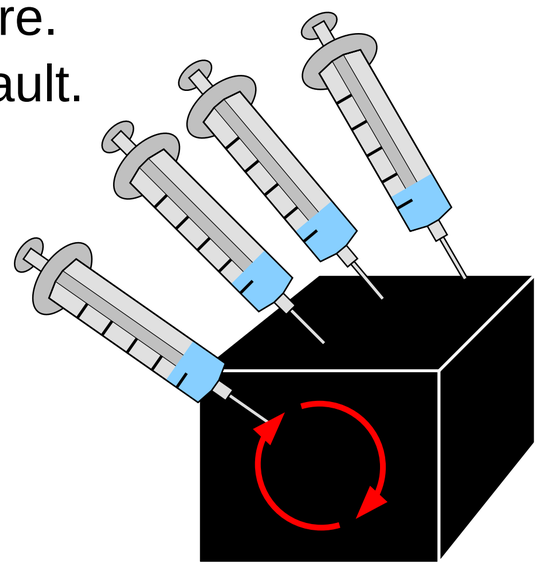
Goal



* Use existing unit tests.
* Pick unit tests affected by exceptions.
* Inject faults into these tests.

Conventional fault injection

* Independent of code structure.
* All tests repeated for each fault.
* Probabilistic fault injection.

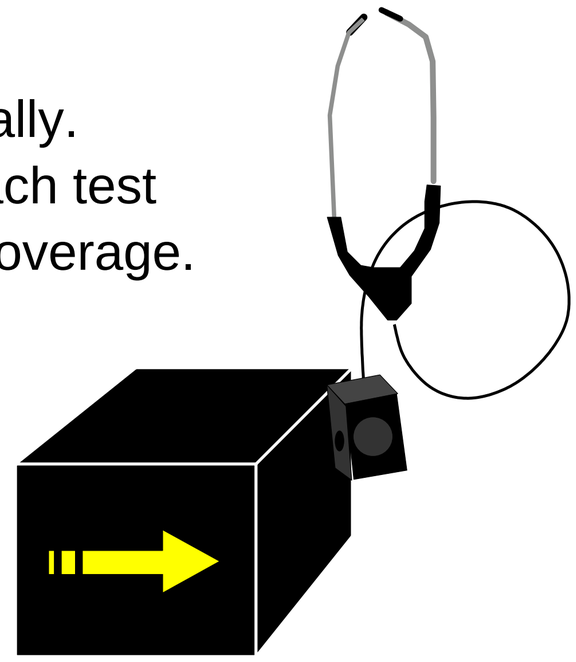


Black box

Solution

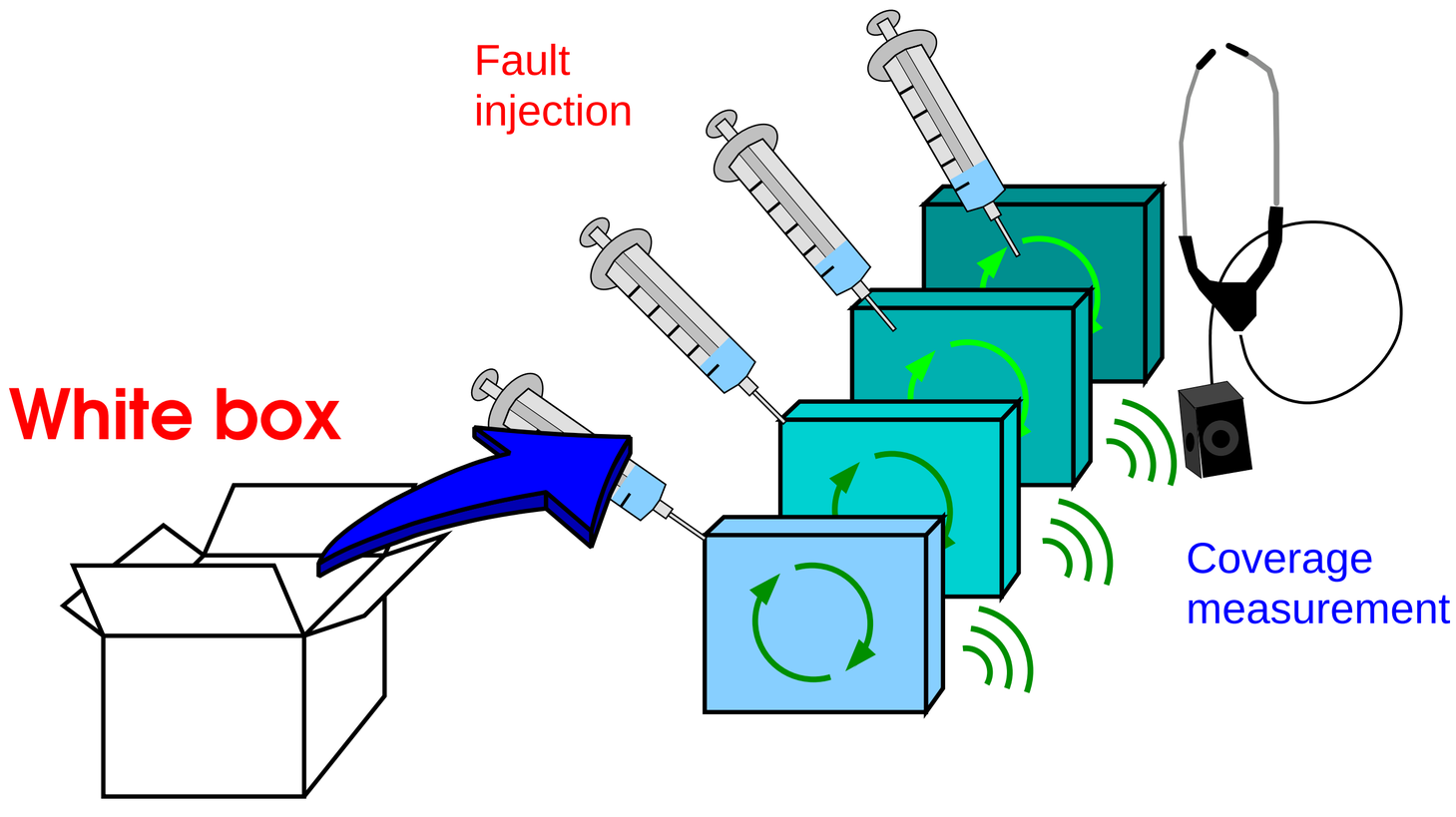
Coverage measurement

(1) Execute test suite normally.
(2) Measure coverage of each test
Result: per−test block coverage.



Test execution

(3) Choose appropriate unit tests,
re−execute one test per fault.



Summary

Use unit test coverage data
to select test case of interest
for each possible fault,
test each fault one by one.

* Faster than black−box approach!
* Also works for nested faults.

URL: http://staff.aist.go.jp/c.artho/enforcer/

Contact: c.artho@aist.go.jp