Testing Challenges for Cyber Physical Systems

Jan Tretmans

TNO - ESI – Embedded Systems Innovation at TNO
Radboud University Nijmegen
Högskolan i Halmstad
jan.tretmans@tno.nl
Jan Tretmans

TNO - ESI
Embedded Systems Innovation
Eindhoven
The Netherlands

Radboud University
Nijmegen
The Netherlands
Jan Tretmans

• Formal Methods
  – Maeslant Kering Rotterdam

• Software Testing

• Model-Based Testing MBT
  – \textit{ioco}-theory for MBT with labelled transition systems
  – MBT tool \textit{TorXakis}

Synopsis
- Foundation ESI started in 2002
- ESI acquired by TNO per January 2013
- ~60 staff members, many with extensive industrial experience
- 7 Part-time Professors
- Working at industry locations

Focus
- Managing complexity of high-tech systems through
  - system architecting,
  - system reasoning and
  - model-driven engineering
delivering
- methodologies validated in cutting-edge industrial practice

Partners
- ASML
- Thermo Fisher Scientific
- Philips
- Vanderlande
- Thales
- Canon
- TNO
- TU Delft
- Radboud University Nijmegen
- University of Twente
Cyber-Physical Systems

Software is brain of system

- software controls, connects, monitors almost any aspect of ES system behaviour
- majority of innovation is in software

Software determines quality and reliability of Cyber-Physical System

- often > 50% of system defects are software bugs
Testing Challenges for Cyber-Physical Systems
Testing Trends & Challenges

- Complexity
- Building Blocks
- Environment
- New Paradigms
- Variability
- Connectivity
- Going Up
- Multi-Disciplinarity
- Non-Functionals
- Agile
- Evolution
- Non-Functionals
Testing Trends & Challenges

- Size & Complexity
- Building Blocks
- Environment
- New Paradigms
- Variability
- Connectivity
- Going Up
- Multi-Disciplinarity
- Non-Functionals
- Agile
- Evolution
Size & Complexity

Car

- 100,000,000 LoC
- 40,000 parts
- 4,000 manufactured components

Machine with 300 parameters

- \(2^{300} = 10^{90}\) different configurations
- #atoms on earth = \(10^{50}\), #atoms in known universe = \(10^{80}\)

Completely testing ‘+’ for 32-bit Int

- \(2^{32} \times 2^{32} = 10^{19}\) test cases
- 1 nsec / test = 585 years of testing
Size & Complexity

Testing effort grows exponentially with system size

Testing cannot keep pace with development

$x : [0..9] \rightarrow \text{10 ways that it can go wrong}$

$10 \text{ combinations of inputs to check}$

$x : [0..9] \rightarrow \text{100 ways that it can go wrong}$

$100 \text{ combinations of inputs to check}$

$y : [0..9] \rightarrow \text{1000 ways that it can go wrong}$

$1000 \text{ combinations of inputs to check}$
Size & Complexity

Testing effort grows exponentially with system size

Testing cannot keep pace with development

→ combinatorial explosion of required testing effort
Testing Trends & Challenges

- Complexity
- Building Blocks
- New Paradigms
- Environment
- Connectivity
- Going Up
- Non-Functionals
- Agile
- Variability
- Multi-Disciplinarity
- Evolution
Variability & Product Lines

or: How to Select your Sandwich
Variability

- Highly configurable sandwich: \textit{exponential number of choices, a different sandwich for everybody on the planet!}

- Not all combinations make sense: \textit{dependencies}

- \textbf{How to taste / test all of them?}

- Sandwich product line = family of sandwiches

- Also for high-tech systems
  Linux, cars, . . .
• Customization & reuse by developing families of 'similar' products
  → identify variation points
  → instantiate to different configurations = products

• Aim: instantiate as late as possible,
  to perform design, analysis, …, on the product family
  and not on each individual product

• But: testing is always on an individual product
  → how to select configurations for testing?
Testing Trends & Challenges

- Complexity
- Building Blocks
- Environment
- New Paradigms
- Variability
- Connectivity
- Going Up
- Multi-Disciplinarity
- Non-Functionals
- Agile
- Evolution
Evolution: Change over Time

- system never comes alone: *variability*
- systems continuously change: *evolution*

yet another source of Test Explosion

variability

product lines

mass customization

series of one

evolution
Evolution, Change: Fading Boundaries

- Requirements
- Design
- Build
- Test
- Operation
Testing Trends & Challenges

- Complexity
- Building Blocks
- Environment
- New Paradigms
- Variability
- Connectivity
- Going Up
- Non-Functionals
- Agile
- Multi-Disciplinarity
- Evolution
Cyber-Physical Systems

- Semiconductor manufacturing equipment
- Medical systems
- Food processing
- Agricultural robots
- Traffic management
- Electron microscopes
- Building control
- Robotized warehousing
- Combat management systems
- Industrial printers
- Automotive
- Dike
Software is Different

Software is different from hardware:

- non-continuous
- any bug is a design error
- adopting redundancy is useless
- no wear and tear
- no MTBF
- what is software reliability?
Multi-disciplinarity

• Cyber-Physical Systems

• Combination of physics/mechanics/electronics … with computer/software

• Requires various expertises

• Testing such combinations requires

  stubs, simulations, virtualization, digital twin
Multi-disciplinarity

• Virtualization
  – models to simulate/emulate physical and environment in tests
  – intelligent stub, in-the-loop testing
  – because real system is: expensive, infeasible, dangerous,
    too slow, too fast, cannot produce error scenarios, …

• Modeling
  – system ↔ physical part ↔ software ↔ environment
  – models for virtualization ↔ models for testing
Multi-disciplinarity: Different Views on Systems
Models for Multi-disciplinary Testing
Building Blocks: Components

Components come from anywhere

- reuse
- platform
- integration challenges
- dependencies
- when to test
- where to diagnose, repair

TO REUSE, OR NOT TO REUSE

IN PARTICULAR, IN TIMES OF CONTINUOUS CHANGES
Components and Failures

Ariane V rocket
- Design defects in control software

• Design
  – Exception handler assumed hardware errors only
  – Reuse of Ariane IV component in Ariane V without proper system testing

• Error
  – Software exception

• Failure
  – Mis-interpretation of diagnostic information
Testing Trends & Challenges

- Complexity
- Building Blocks
- Connectivity
- New Paradigms
- Environment
- Going Up
- Agile
- Multi-Disciplinarity
- Variability
- Evolution
- Non-Functionals
Connectivity

- Blurring boundaries of systems ➔ everything connected
- Systems-Of-Systems
  - Dynamically connected systems
  - Not under own control
- Software is glue
  - with internal and external world
- Testing:
  - what is SUT ?
- Virtualization
  - which systems are available for testing ?
  - which systems must be virtualized?
- Dynamics
  - run-time testing and integration
Fading Boundaries

SYSTEM

ENVIRONMENT

SYSTEM

OF

SYSTEMS
Testing Trends & Challenges

- Complexity
- Building Blocks
- Variability
- Connectivity
- New Paradigms
- Multi-Disciplinarity
- Environment
- Evolution
- Non-Functionals
- Going Up
- Agile
- Agile
Environment

tester

CPS

software

simulation model
virtualization
digital twin
Environment

tester

CPS

software

simulation model
virtualization
digital twin

environment
Environment

functional

state-based

calculation: \( I \rightarrow O \)

reactive: \( I, S \rightarrow O, S' \)

proactive: \( I, S, E \rightarrow O, S', E' \)

tests over \( I \)

tests over \( I, S \)

tests over \( I, S, E \)

for safety, trustworthiness, dependability, the environment must be taken into account
Environment

• Autonomous
  → take part in environment

• Safety of autonomous cars
  → test in all possible environments

• Environment
  → not, or limited, under (test) control

• Environment
  → can change
  → new testing ?

Testing everything before release is an illusion
→ continue quality control after release
Environment

tester

CPS

software

simulation model
virtualization
digital twin

environment model
virtualization
Testing Trends & Challenges

- Complexity
- Building Blocks
- Connectivity
- Variability
- Non-Functionals
- Evolution
- Multi-Disciplinarity
- Environment
- Going Up
- Agile
- New Paradigms
Quality Characteristics

Reliability
- maturity
- fault tolerance
- recoverability
- availability
- degradability

Efficiency
- time behaviour
- resource behaviour

Portability
- adaptability
- installability
- conformance
- replaceability

Functionality
- suitability
- accuracy
- interoperability
- compliance
- security
- traceability

Usability
- understandability
- learnability
- operability
- explicitness
- customisability
- attractiveness
- clarity
- helpfulness
- user-friendliness

Maintainability
- analysability
- changeability
- stability
- testability
- manageability
- reusability
Non/More Emphasis on Quality Function ("attributes")

Trend: More Emphasis on Quality Characteristics

(from ISO 25010)
For large and complex systems:
Some Quality Attributes are Compositional, others are not → emerging qualities
Testing Trends & Challenges

- Complexity
- Building Blocks
- New Paradigms
- Environment
- Going Up
- Variability
- Connectivity
- Non-Functionals
- Agile
- Multi-Disciplinarity
- Evolution
- Agile
Going Up

• In the Value Chain
  – new business models
  – testing quality-of-service

• In the Test Pyramid
  – everybody does unit tests
  – bugs are on the higher levels
Going Up

• In Coding
  – from software to meta-software: build tools, build scripts, configuration setting, ...
Going Up Consequence: Uncertainty & Non-Determinism

- Sometimes you don’t know …..
  - testing a search engine,
    weather forecast, …
  - systems-of-systems, big data, ...

- Sometimes you don’t want to know …..
  - no details
  - abstraction
  - particular view

Uncertainty of test outcomes & oracles
  - non-determinism
  - probabilities

What is the weather like?
Testing Trends & Challenges

- Complexity
- Building Blocks
- Environment
- Connectivity
- Going Up
- New Paradigms
- Agile
- Variability
- Multi-Disciplinarity
- Evolution
- Non-Functionals
Agile?

We're going to try something called Agile programming.

That means no more planning and no more documentation. Just start writing code and complaining.

I'm glad it has a name.

That was your training.
Agile

• Agile \(\rightarrow\) test automation
  – test execution automation
  – test fast and often
→ Large repositories of scripted tests
  – the night is too short
  – traceability to requirements?
  – maintainability?
  – pesticide paradox: *how to increase variation in tests?*

Agile - fallacy of complete specification:
*We finally have the guts to admit that we don't know precisely what the system should do when we start coding.*
Testing Trends & Challenges

- Complexity
- Building Blocks
- Environment
- Connectivity
- Going Up
- Non-Functionals
- Agile
- New Paradigms
- Variability
- Multi-Disciplinarity
- Evolution
New Paradigms and Technologies

• AI, Machine Learning
• Self-adaptive systems
• Quantum Computing
• Cloud
• Ethics, sustainability, ...
• .......
Testing Trends & Challenges

- Complexity
- Building Blocks
- Connectivity
- Non-Functionals
- Environment
- New Paradigms
- Evolution
- Variability
- Multi-Disciplinarity
- Going Up
- New Paradigms
- Challenges Ahead