

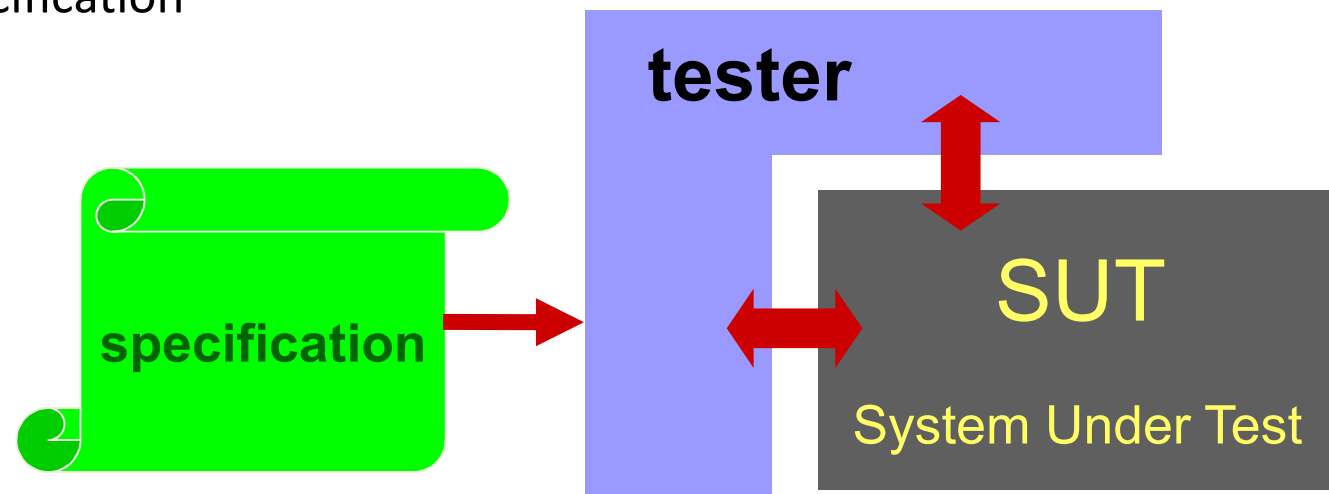


Model-Based Testing

TestNet Werkgroep MBT

Software Testing

Checking or measuring
some quality characteristics
of an executing software object
by performing experiments
in a controlled way
w.r.t. a specification





Testing : Why Something New ?

- Quest for improved quality, with less time and cost
 - *detect more bugs faster and cheaper*
- Changing processes:
 - *scrum, agile*
 - *model-driven*
- Increasing system size, complexity
- Blurring boundaries
 - *time: continuous change, continuous delivery*
 - deliver fast, deliver often → **test fast, test often**
 - *space: complex interactions, systems-of-systems*
 - end-to-end testing → **where is the end ?**

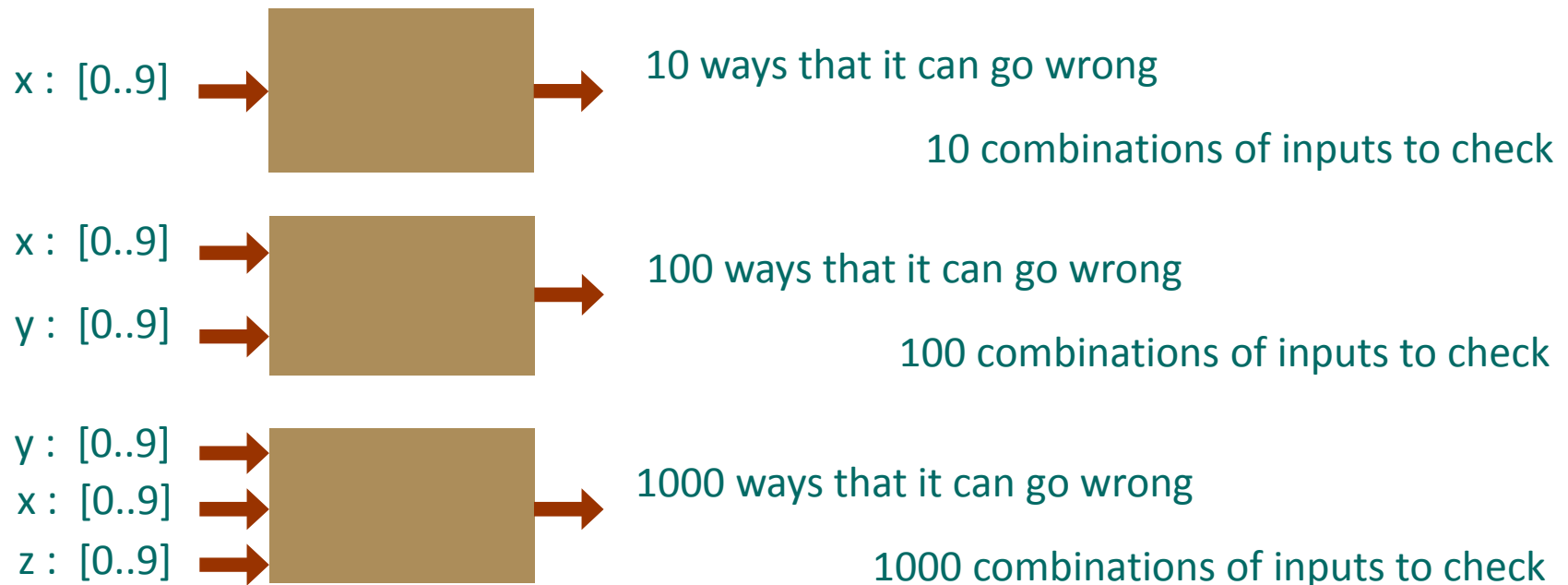
Software bugs cost US economy
\$ 59.500.000.000 (www.nist.gov)



Testing Complexity

Testing effort grows exponentially with system size

Testing cannot keep pace with development



→ combinatorial explosion of required testing effort

Testing Complexity

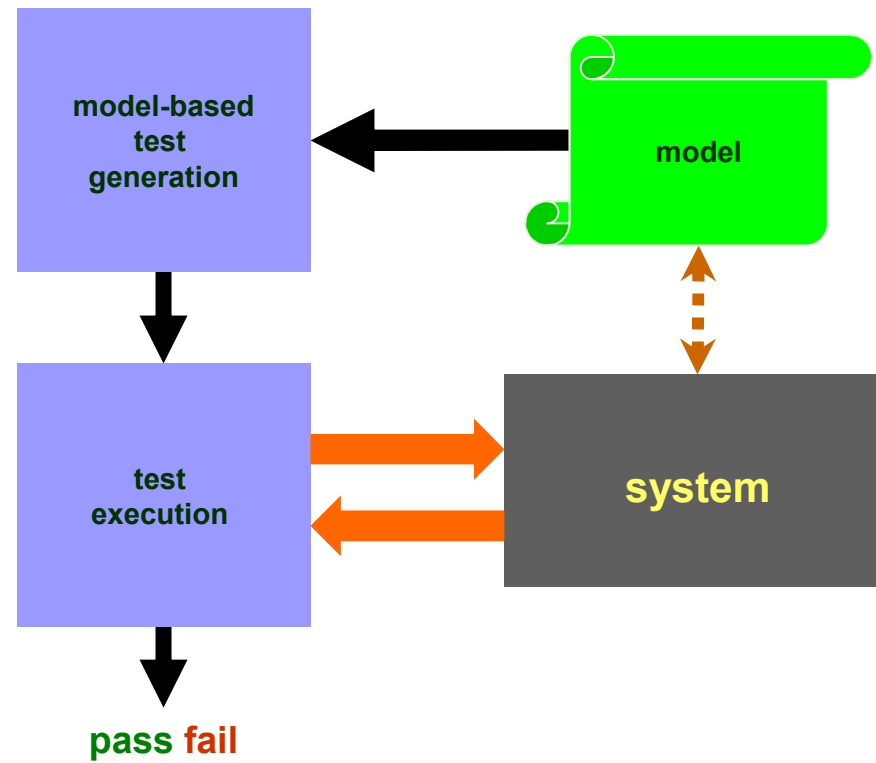
Testing effort grows exponentially with system size

Testing cannot keep pace with development



→ combinatorial explosion of required testing effort

MBT

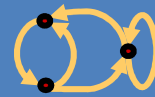




What is MBT ?

- **Model-Based Testing**
- Next step in test automation
 - + **automatic test generation and result analysis**
- New way of testing to detect more bugs faster and cheaper
- Testing with powerful tools
- Testing based on an abstract system **model**
that specifies the behaviour of (an aspect of) the SUT

MBT : Next Step Test Automation



Model-Based Testing

keyword	parameters
customer	Jan Pieters
price	€ 20.45
number	3

Keyword-Driven



Scripted



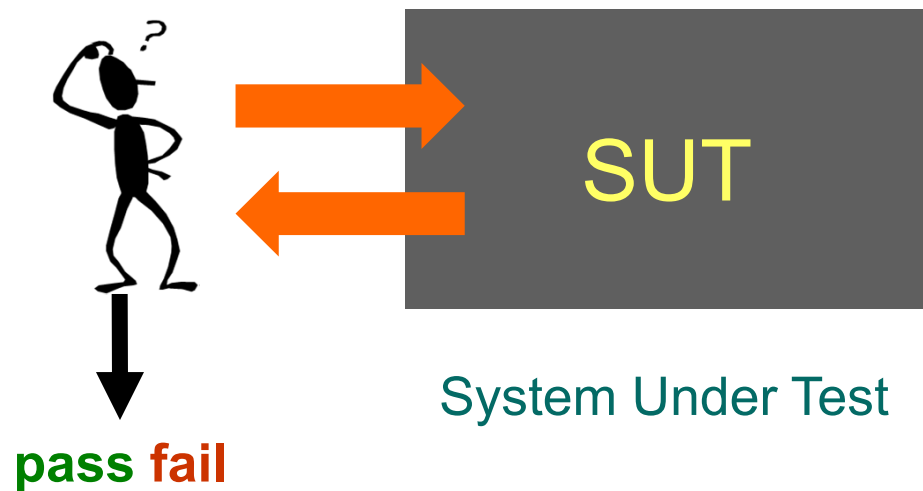
Record & Playback



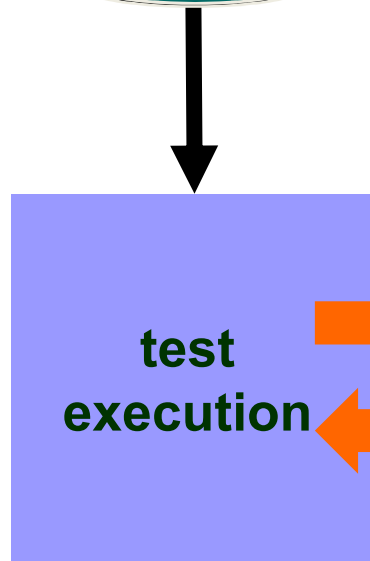
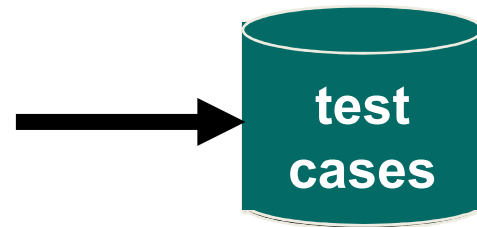
Manual Testing

Testing 1 : Manual Testing

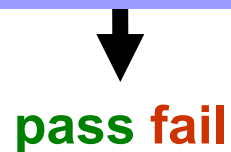
1. Manual testing



Testing 2 : Scripted Testing



1. Manual testing
2. Scripted testing



A black arrow points from the "test execution" box to the text "pass fail".

pass fail

Testing 3 : Keyword-Driven Testing

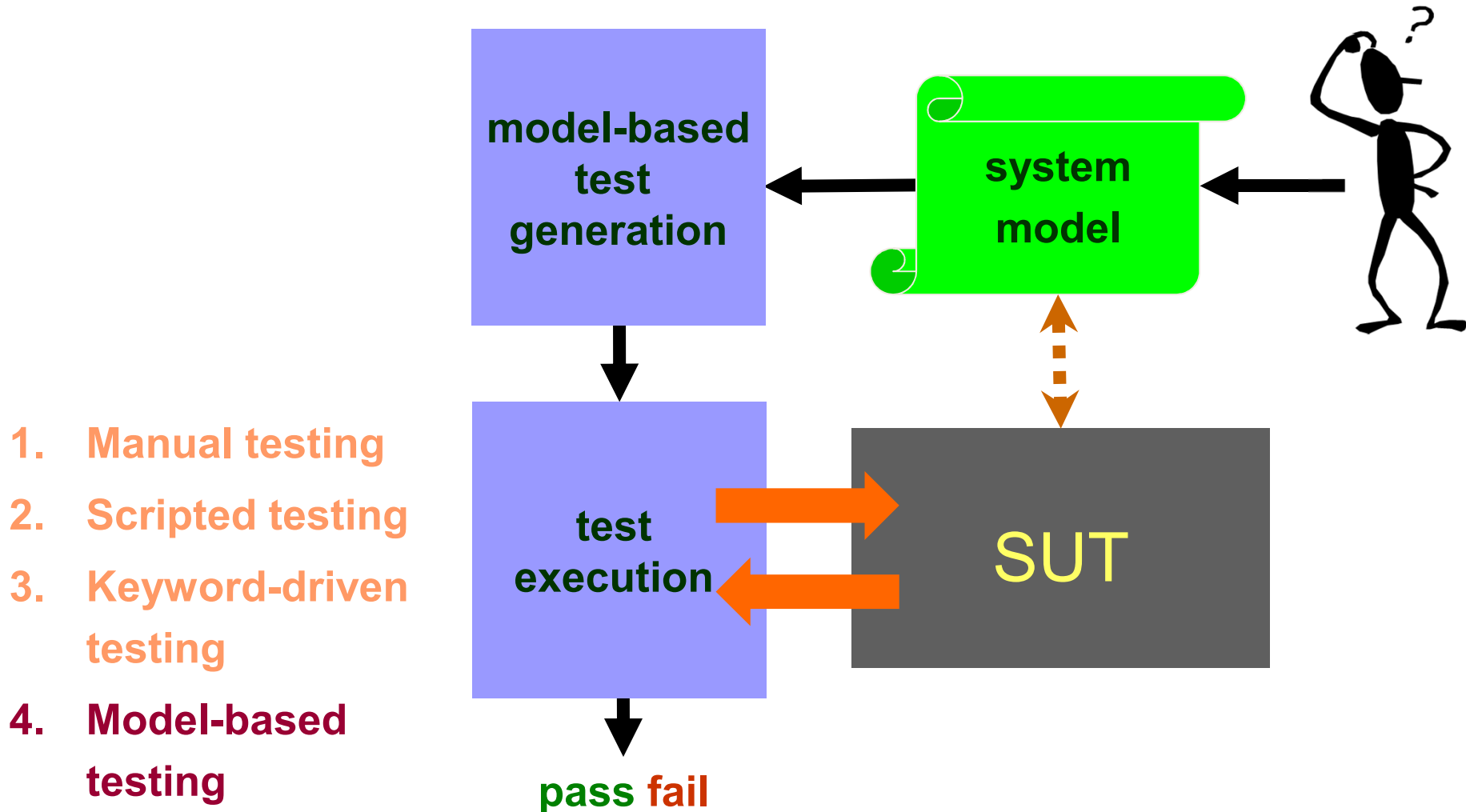


- 1. Manual testing
- 2. Scripted testing
- 3. Keyword-driven testing



pass fail

Testing 4 : Model-Based Testing



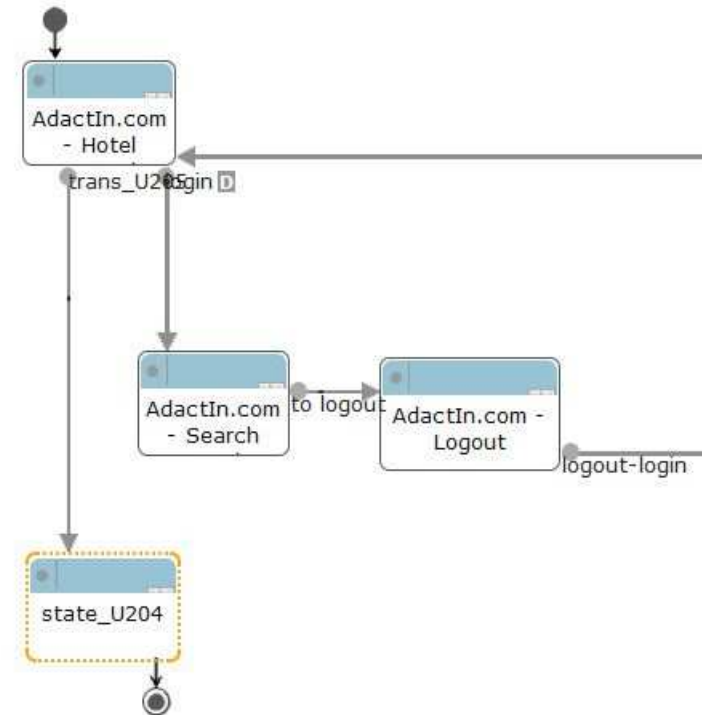
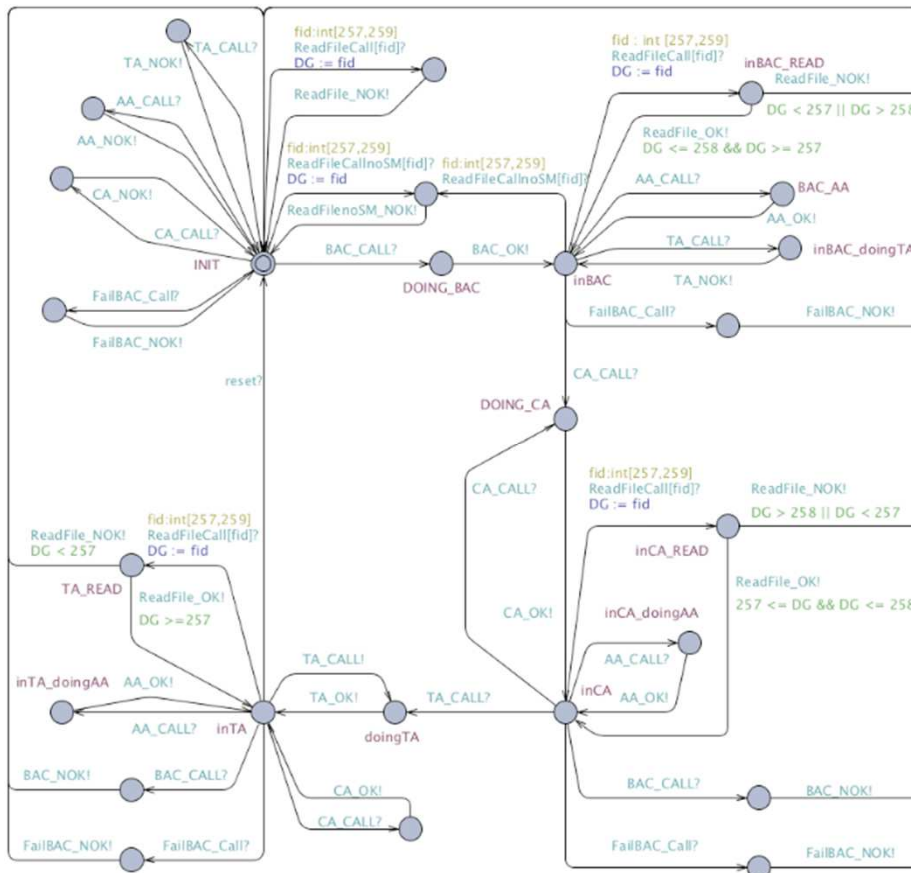
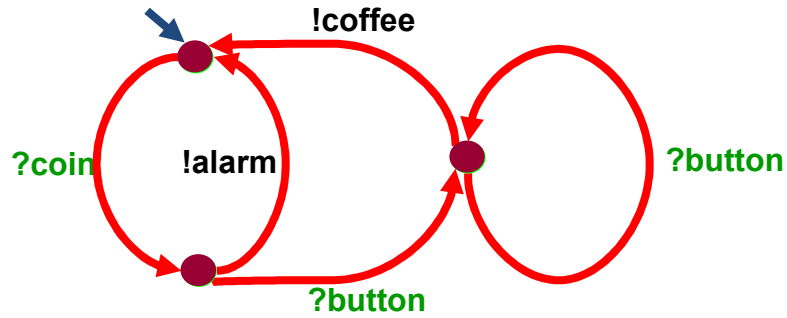
1. Manual testing
2. Scripted testing
3. Keyword-driven testing
4. **Model-based testing**



MBT Improves the Testing Process

- Automatic testing
 - *test generation + execution + analysis*
- Detecting more bugs faster and cheaper
 - *more, longer, and more diversified tests*
- Model is precise and consistent test basis
 - *unambiguous analysis of test results*
- Early error detection during model construction and analysis
 - *improved understanding of system requirements*
- Test maintenance by maintaining models
 - *improved regression testing*

Models





And Moreover . . . MBT . . .

- Enabling **flexibility** in modelling alternative, concurrent, distributed, exceptional, uncertain, . . . behaviours
- **Compositionality** : building large models from simple ones
- Expressing **test coverage** by model coverage
- **Diagnosis** through model analysis
- Linking to **model-based system development**
- Promoting **uniformity** of tests and test processes

MBT : Abstraction + Composition

- Four tasks in parallel, in any order

task(start?, ready!)



taskA := task (startA?, readyA!)

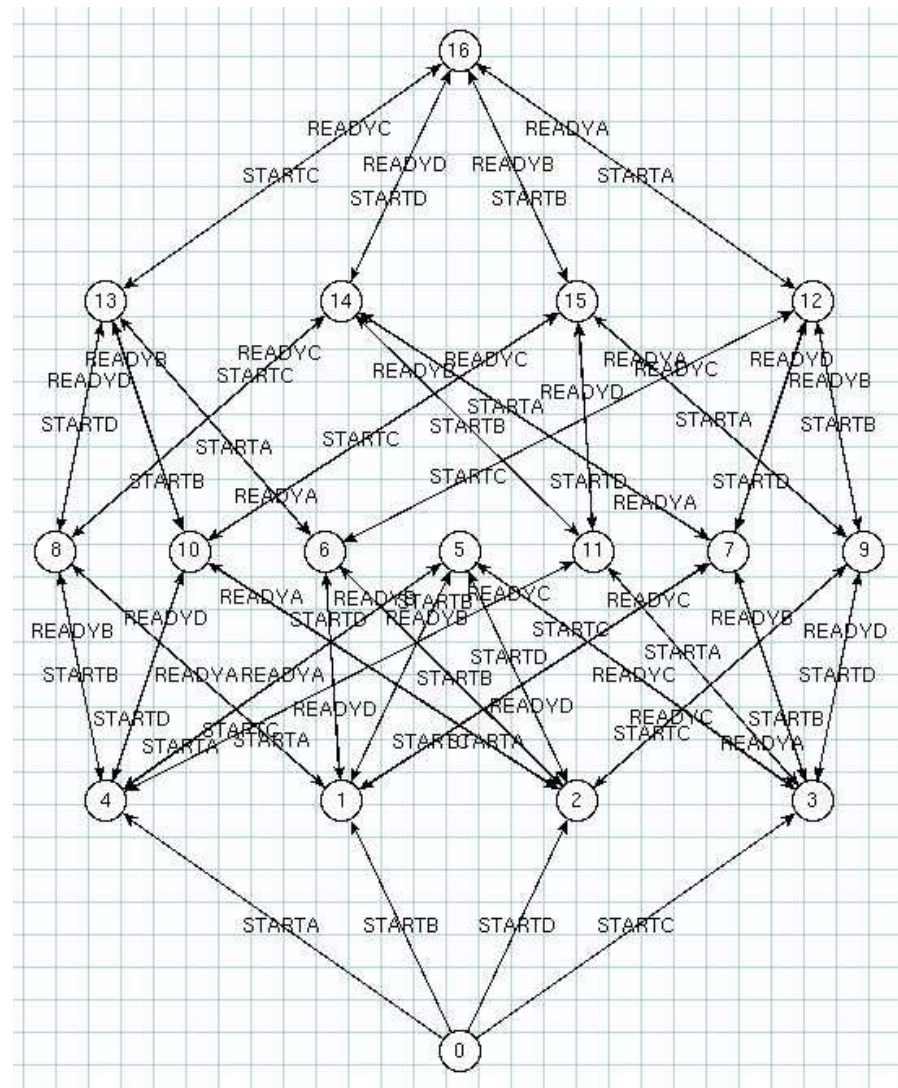
taskB := task (startB?, readyB!)

taskC := task (startC?, readyC!)

taskD := task (startD?, readyD!)

model := taskA ||| taskB ||| taskC ||| taskD

MBT : Abstraction + Composition

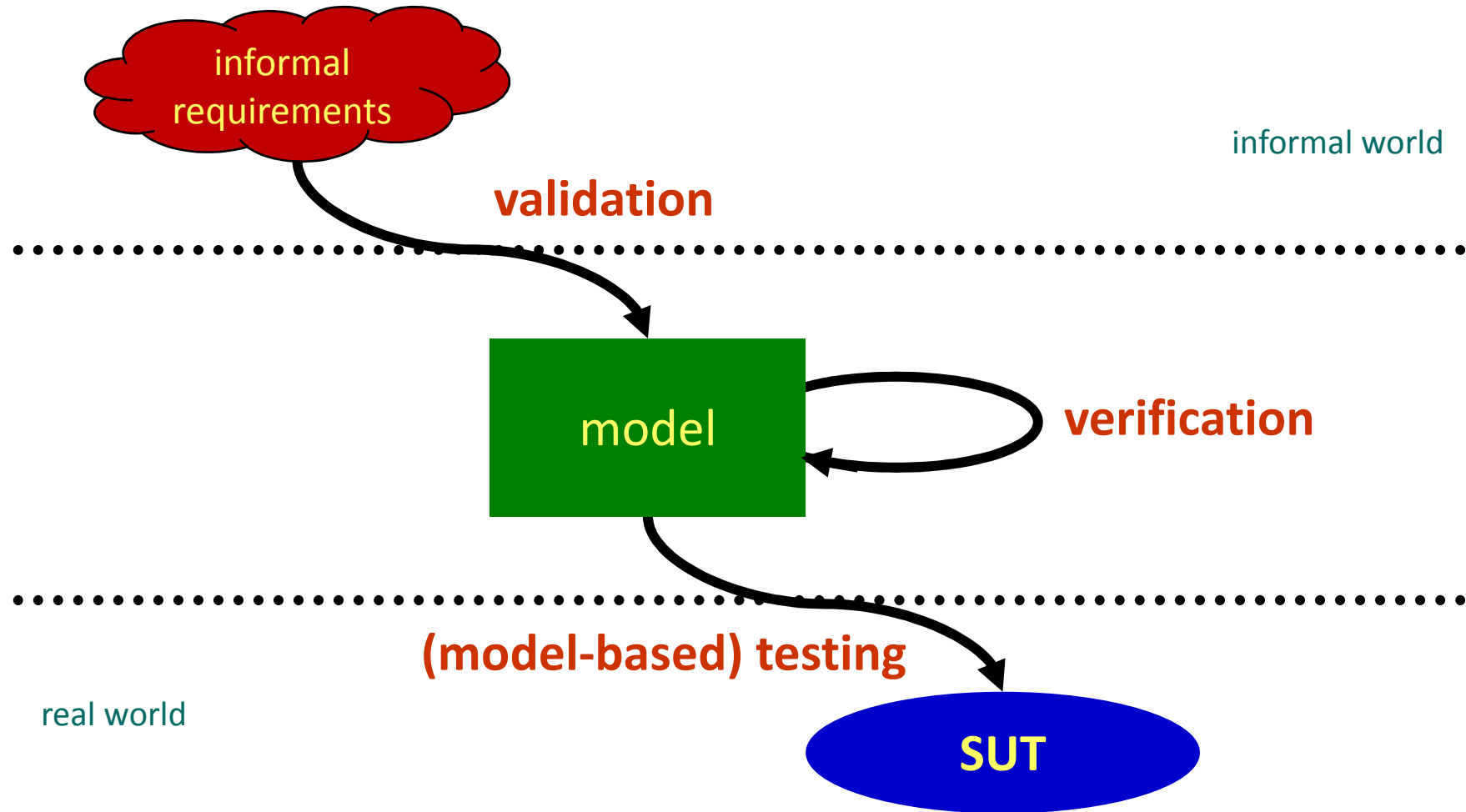




Doing Something with Models

- **Modelling** making a model reveals errors
- **Simulation** go step-by-step through the model
- **Model checking** go through all states of the model
- **Theorem proving** prove theorems about the model
- **Code generation** executable code from the model
- **Testing** test an implementation for compliance
- **Model learning** generate a model from observation

Validation, Verification, and Testing



Code Generation from a Model



men van selectie, configuratie dus; de modelgedreven gemeenschap denkt typisch in termen van creatie, customization dus. Er zou echter geen verschil moeten zijn. Wat ik wil duidelijk maken, is dat beide werelden

'Modellerings zonder codegeneratie is zinloos'

heel goed zijn te combineren. Sommige variabiliteit is te vatten in configuratie, andere variabiliteit in customization. Sommige dingen zijn het best uit te drukken met feature-modellering, andere zijn het best te representeren met domeinspecifieke talen.'

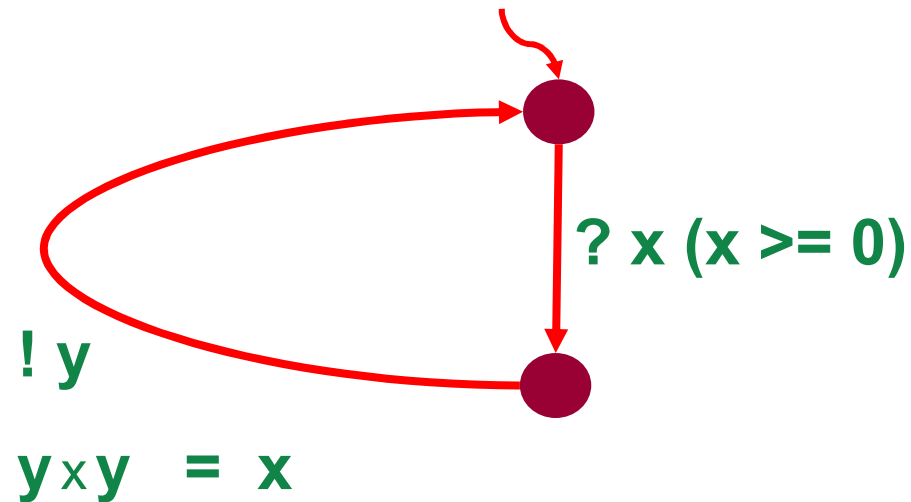
De combinatie is nog verder door te voeren. 'Configuratie is niet alleen te gebruiken om parameters in te stellen, maar ook om modellen te veranderen', licht Völter toe. 'Voor elke feature die je niet selecteert, vervalt er een aantal toestanden in je toestandsdiagram. Zo komen configuratie en customization samen, wat alles een stuk simpeler maakt. Een domeinspecifieke taal is beknopt, exact en high-level.'

A model is more (*less*)

than code generation:

- views
- abstraction
- testing of aspects
- verification and validation of aspects

Code Generation from a Model ?



model of \sqrt{x}

- specification of **properties** rather than construction
- **under-specification**
- **non-determinism**

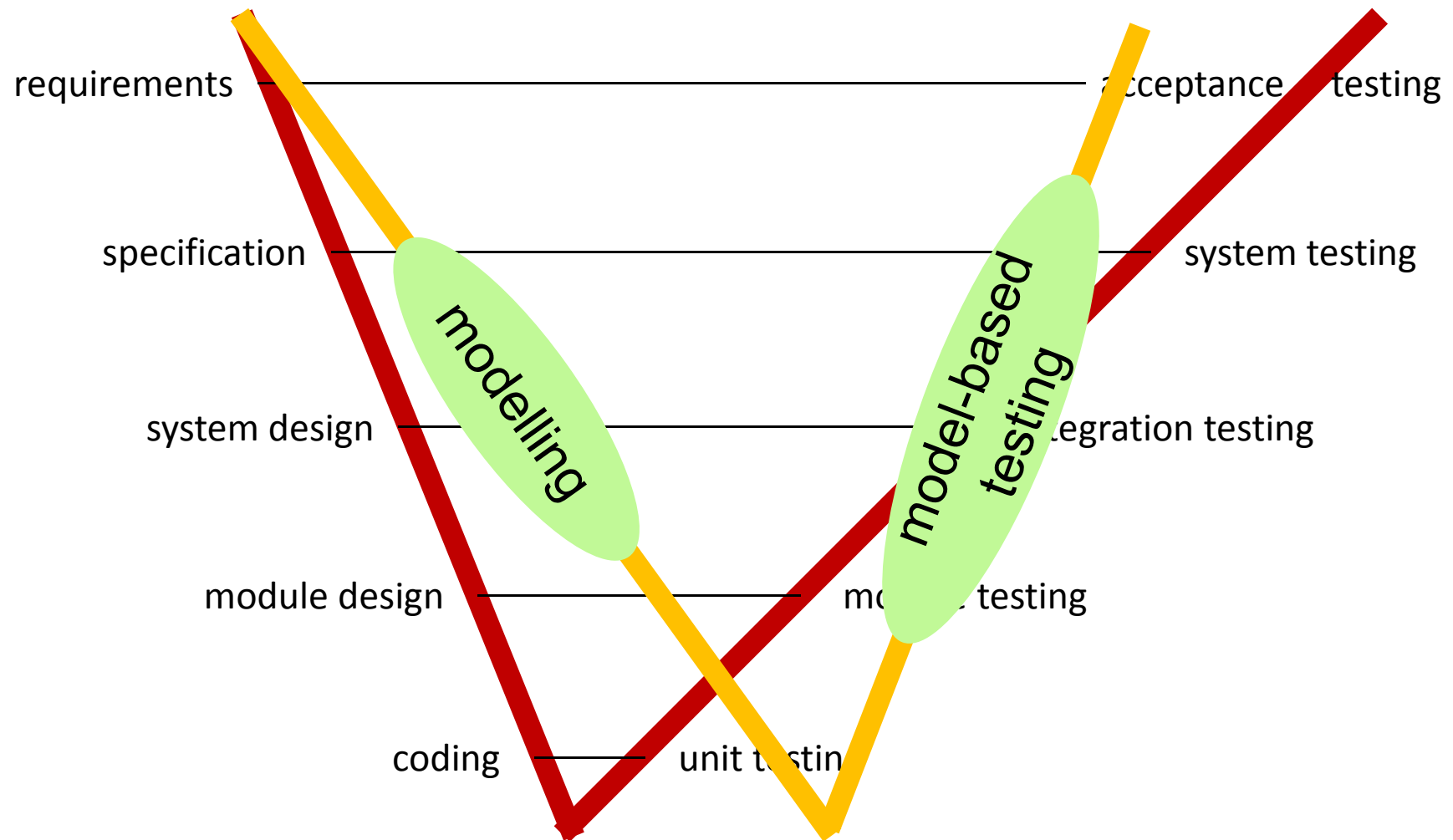


MBT : Status & Challenges

Promising, emerging approach, **but**

- Who does the modelling ?
extra effort of making models compensated by better tests
- Scalability
- Availability of complete tools
- Shifting in the development process

MBT : In the V-Model

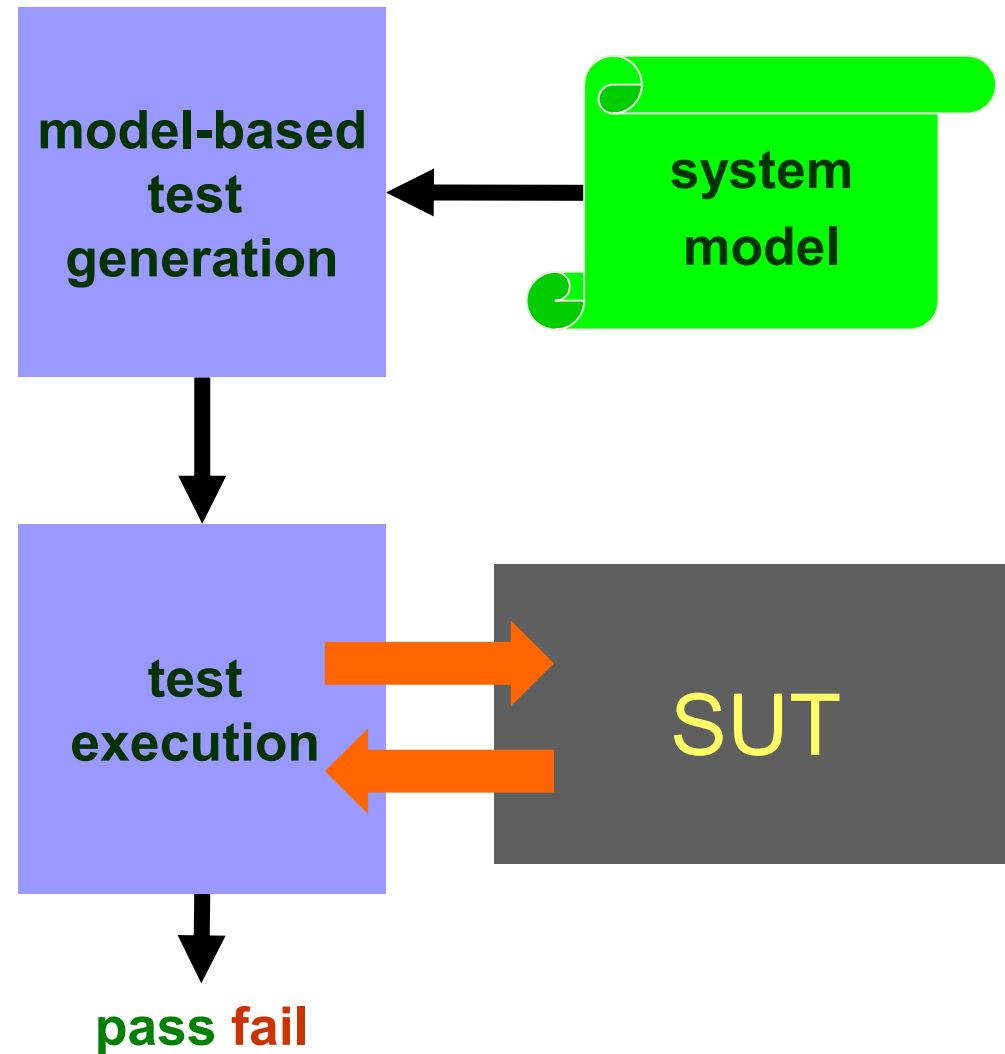




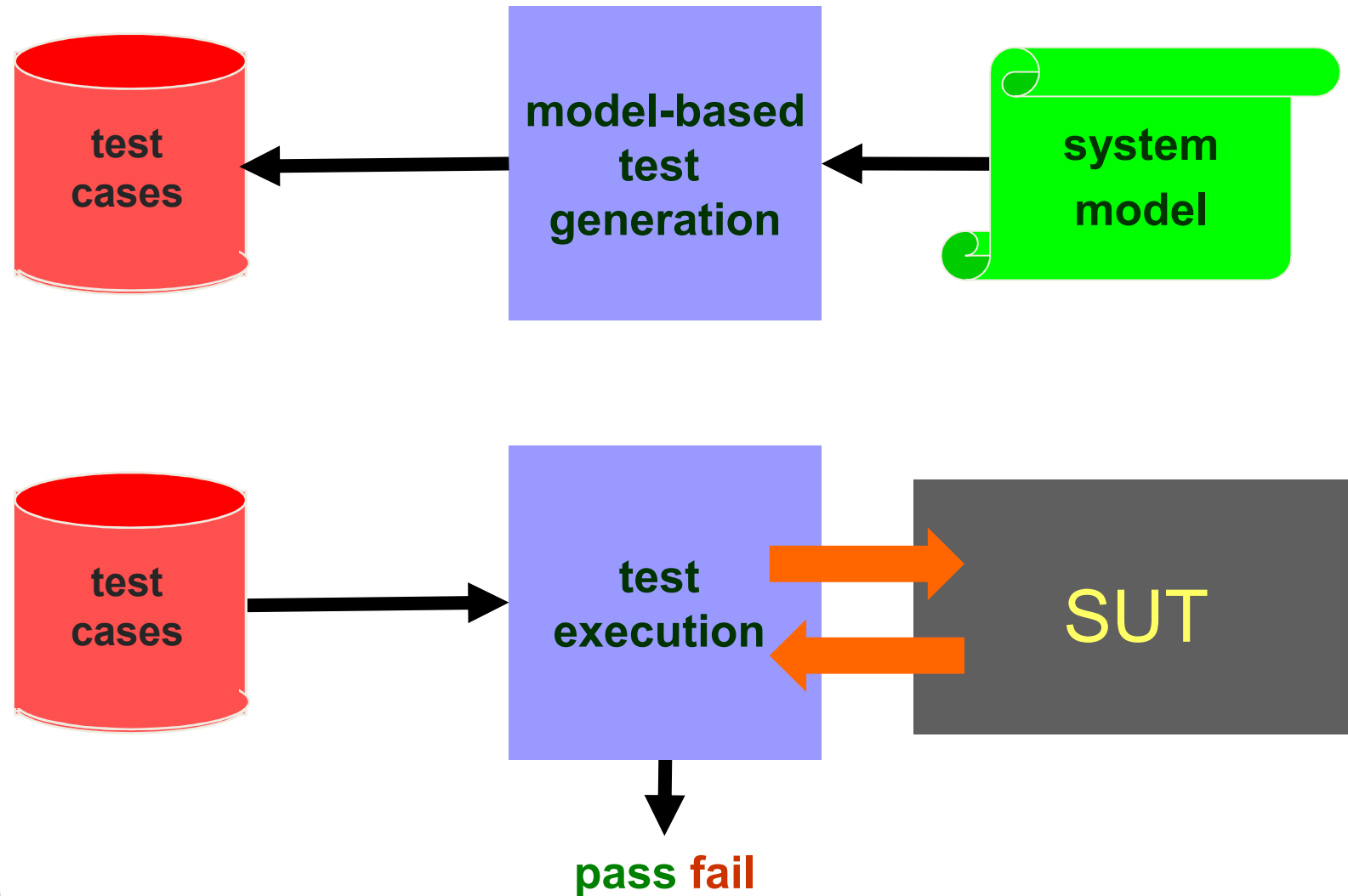
MBT : Some Tools

- AETG
- Agatha
- Agedis
- Autolink
- Axini Test Manager
- Conformiq
- Cooper
- Cover
- DTM
- G \forall st
- Gotcha
- JTorX
- MaTeLo
- MBTsuite
- MBTsuite
- NModel
- ParTeG
- Phact/The Kit
- QuickCheck
- Reactis
- Recover
- RT-Tester
- SaMsTaG
- Smartesting CertifyIt
- Spec Explorer
- Statemate
- STG
- Temppo
- TestGen (Stirling)
- TestGen (INT)
- TestComposer
- TestOptimal
- TGV
- TorX
- TorXakis
- T-Vec
- Uppaal-Cover
- Uppaal-Tron
- Tveda
-

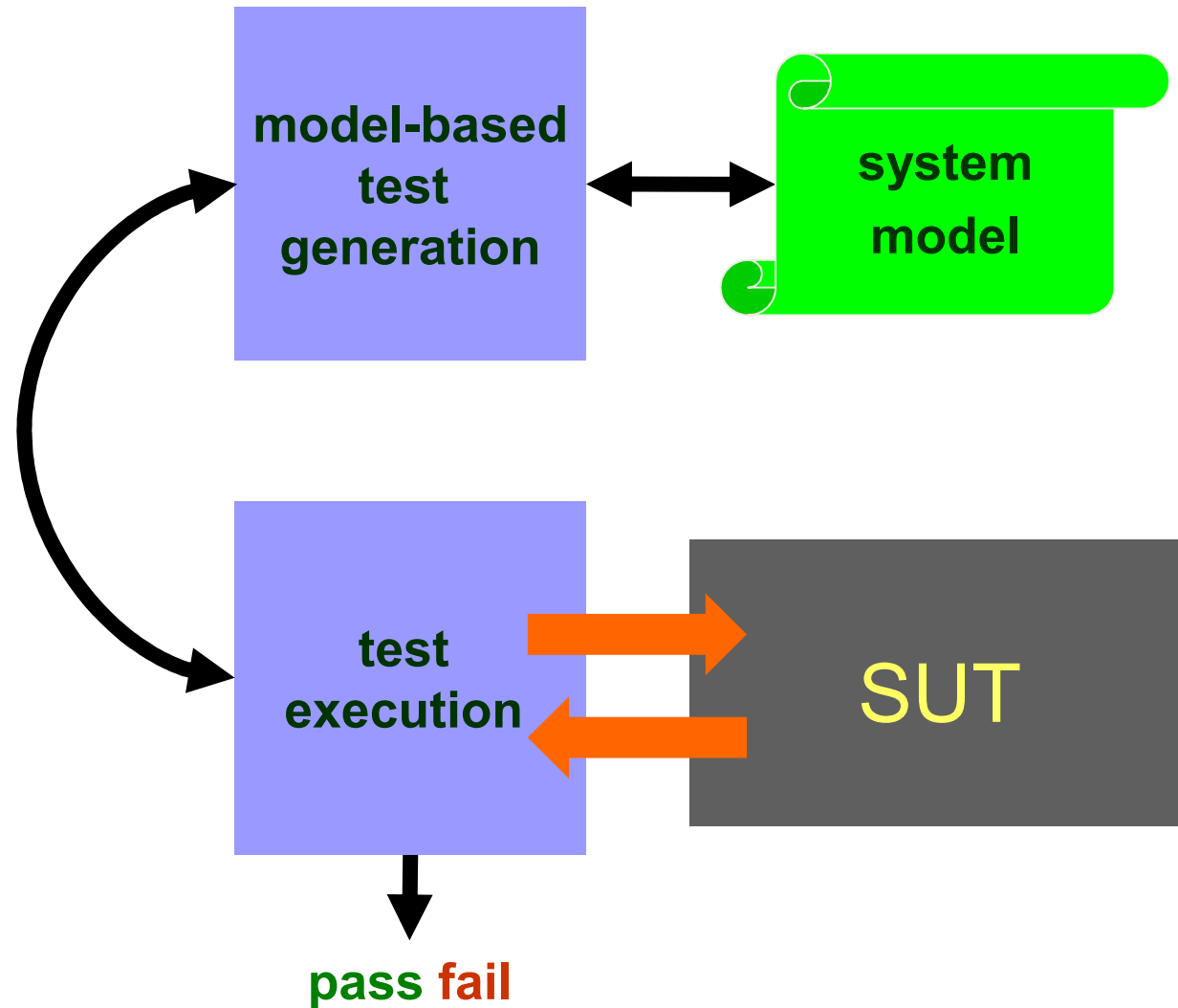
MBT Tools : Off-Line - On-Line



MBT Tools : Off-Line = Batch



MBT Tools : On-Line = On-the-Fly

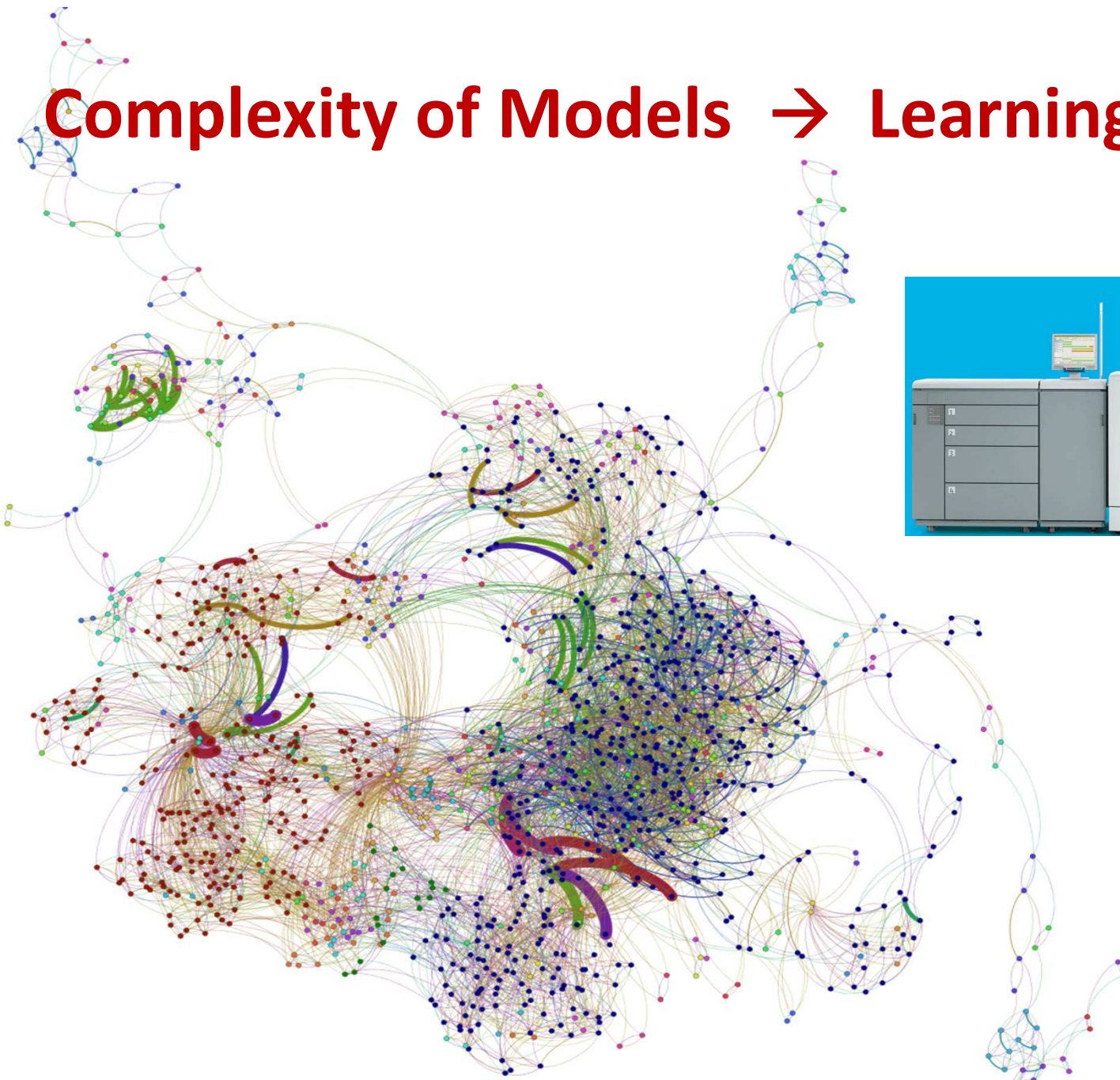




MBT Future : Current Research

- Partial and incomplete models
 - uncertainty in models and systems
- Test case selection
 - testing based on user profiles
 - risk-based MBT
- Integration of testing methods and tools
 - category partitioning, state-based, ioco, risk-based, combinatorial, constraint-based test-data generation, statistical test generation,
- Model learning
 - automatic generation of models from tests

Complexity of Models → Learning



MBT

Questions?

