



# Introduction of MBT by G&D smartcard, feedback on ES 202 951

Dimitry Polivaev  
14.05.2012



Giesecke & Devrient  
Creating Confidence.

# Agenda

## Introduction of MBT by G&D smartcard

- G&D History
- MBT by G&D
- Test engineering concepts
- Increased reuse and reduced costs
- Rule-based Test Generation

## Feedback on ES 202 951

- How our MBT implementation matches the standard
- Requests for the future work



# G&D has been growing through continuous innovation

Server software and services



Token and embedded security



Cards for payment and telecommunications



Government solutions



Banknote processing



Banknote and security paper



Banknote and security printing



1852

2012

Global leader in secure solutions



# MBT by G&D smartcard

## Research since 2007

- Development of test generation methodology, language and components

## Productive projects since 2009

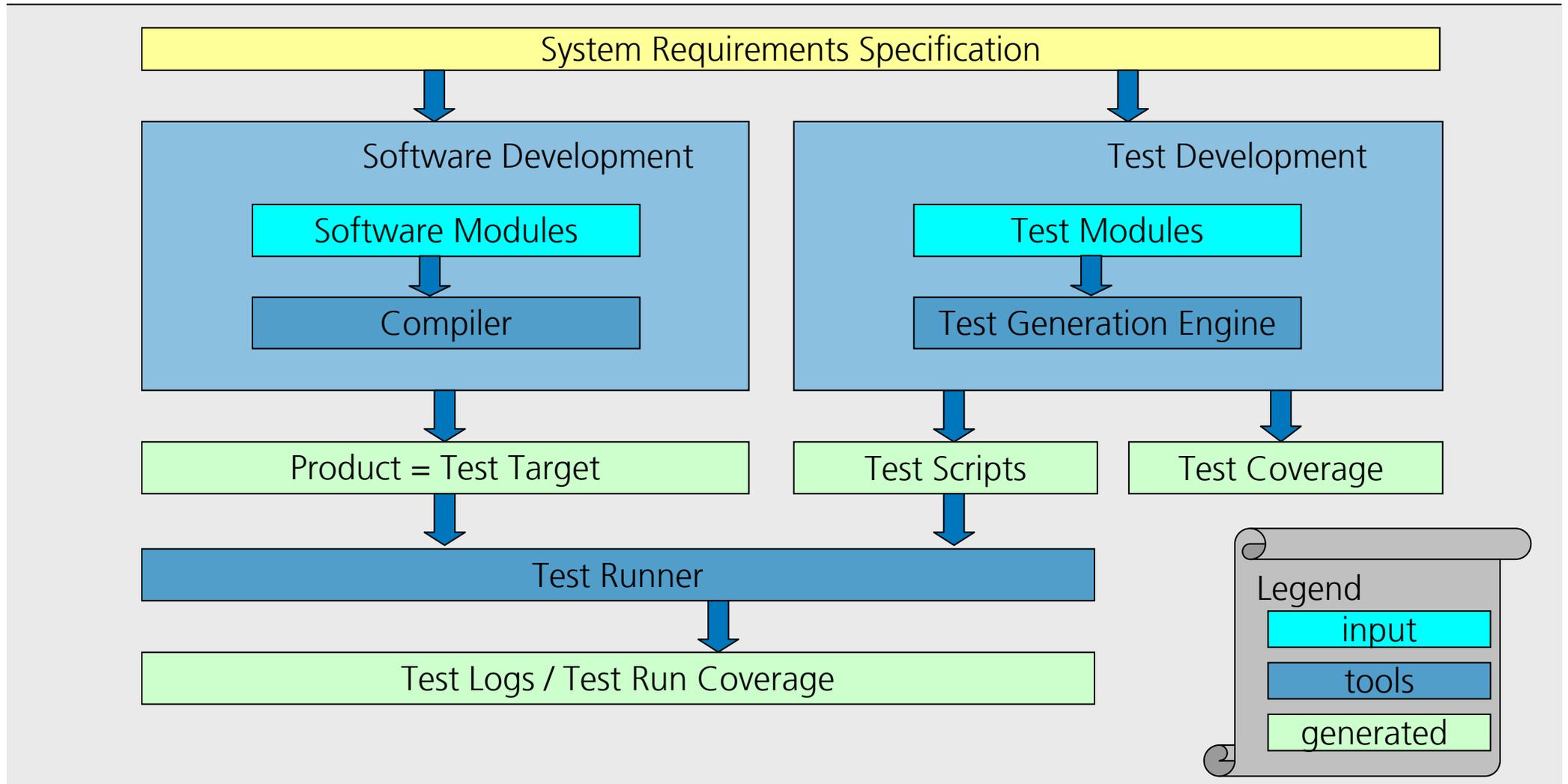
- Tests of smart cards for payment (VISA, MASTERCARD) mostly base on the MBT methodology
- Kern team of 5 testers

## Use in Developer Centers in India in Spain since 2011

- More testers, distributed project teams



# Test engineering is similar to product development



# The test modules

## Test generation components developed by our Test Engineers:

- Test space exploration rules (we call them a Test Strategy),
- Test case selection criteria (we call them Test Goals),
- SUT behavioral model (it is used for calculation of expected results and evaluation of test selection criteria based on code/boundary/MCDC and state coverage executing the tests against the model ),
- Precondition / postprocessing generator (we call it Solver),
- Test script writer producing the test scripts for concrete target language.

# Advantages of test component modularization

Testers can specialize on different component types which reduces their learning curve,

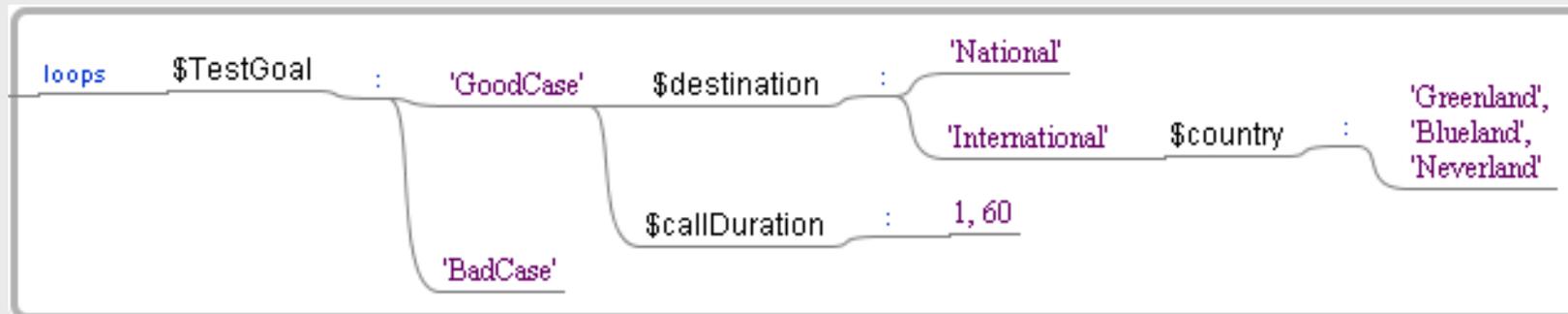
Separation of concerns helps to increase simplicity and the reuse of the components and to reduce costs.

- Strategies can be exchanged for testing of different aspects. For instance it is possible to have a small amount of test cases, with a high coverage of the requirements, available quickly. During the bug fixing of the found errors the strategies can be extended.
- SUT models can be exchanged for testing of different products / product variations.
- Different solvers can be applied for different ways or prepare initial state or reset test target after test.
- Different writers can be used for generating scripts for different script languages
- Different test selection criteria can be defined for test case selection for test depth variation, e.g. for smoke tests, regression tests etc.

# Rule-based Test Generation with mind maps

Test space exploration heuristics is defined as a system of rules.  
The rules are specified in a mind map.

Example: test strategy as a mind map:



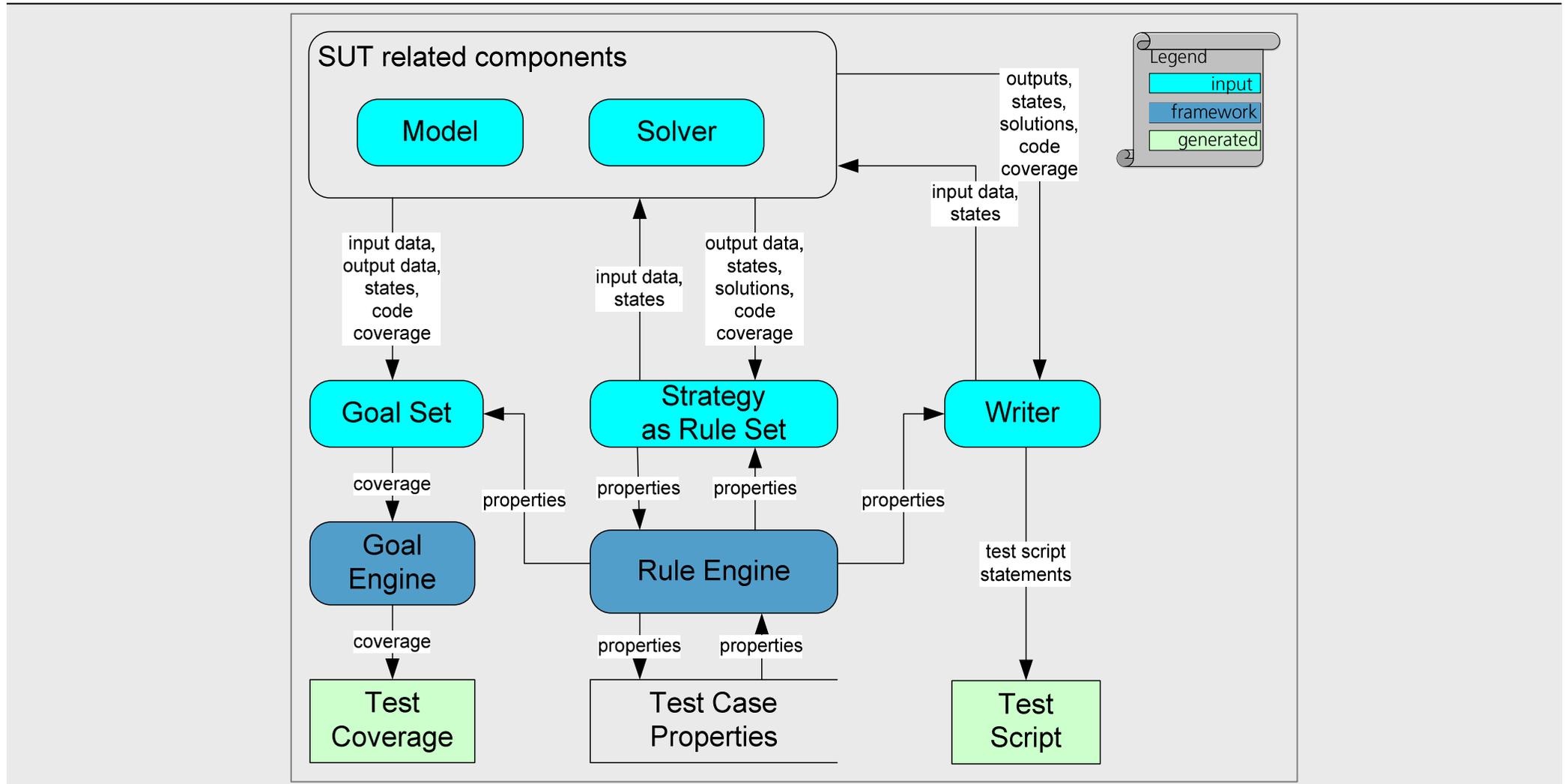
This technique was presented in MBT-Workshop 2012 in Tallinn,

- <http://mbt-workshop.org/>
- <http://arxiv.org/pdf/1202.6125.pdf>

# We use this technique since 2011.



# Test generator architecture by Giesecke & Devrient



## ES 202 951 v 1.1.1 concentrates on the modeling notation

The standard discusses different ways of modeling the SUT behavior.

- Our behavior models would satisfy the standard too.

The standard aims for defining the base concepts and abstractions,

- sometimes (particularly in chapter 7) it seems only to give some concrete examples of specific modeling notations.

It does not cover other aspects like test heuristics or selection, and therefore

- it does not allow to see how much modeling or testing is enough,
- it does not help to compare other aspects of the test generation.

Our engineers do not necessarily start with the behavior models.

They spend the most efforts on the other tasks.

# Further standardization of MBT tools and methods

Let us define standards and comparison criteria for different implementations of

- the test space exploration,
- the test case selection and coverage criteria,
- calculations of the preconditions and the postprocessing,
- a test script output procedure,
- as well as for the SUT behavior modeling.

The standardization should increase tool comparability and simplify tool selection due to common terminology and concepts,

Beyond the comparison it should allow to take the above modules from different vendors and to use them in a single tool chain.