



Model Based Testing

Fabrizio Stortoni

Siemens Building Technologies

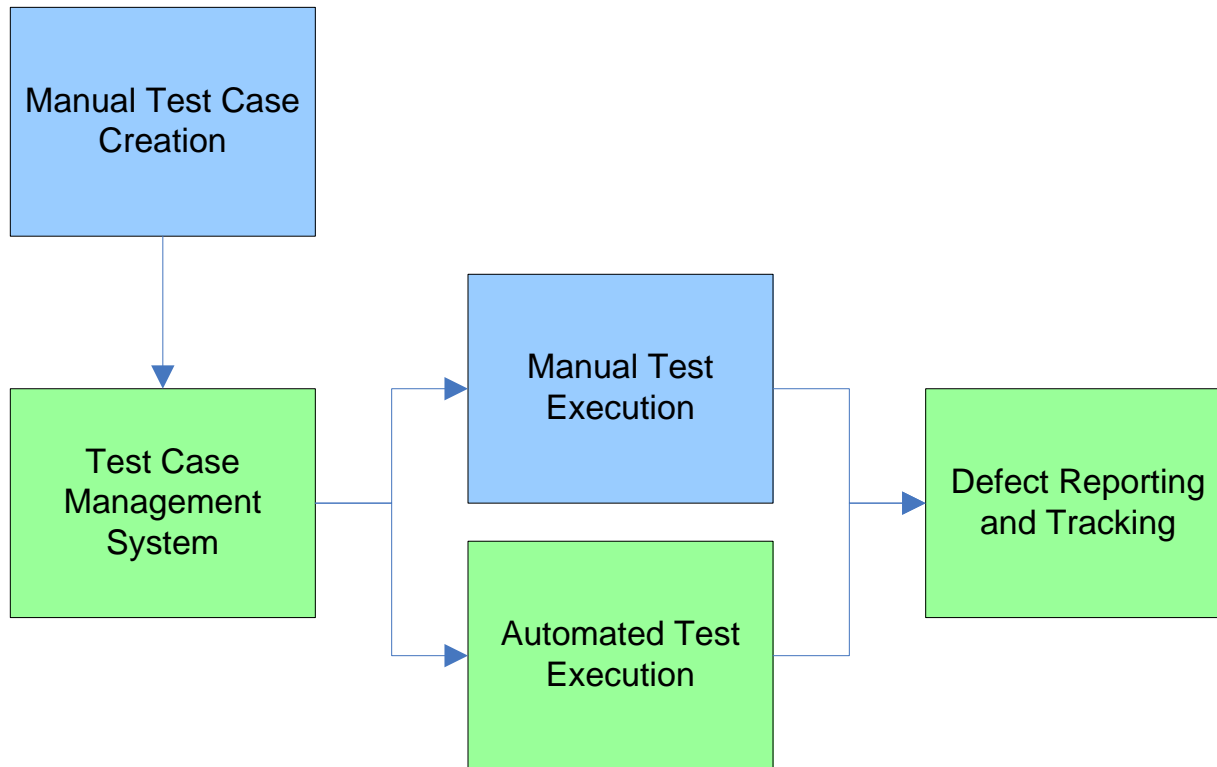
CQAA Lunch & Learn April 11, 2012



Overview

- Test Automation has focused on
 - Test case management (e.g. Quality Center),
 - Automated test execution (e.g. Quick Test Pro)
- In most organizations Test Case Creation remains a manual activity.
- Automated tests are manually created based on the manual test cases, and executed for regression, not to find defects in new functionality

Extent of Automation in a Typical Organization





Model Based Testing Goals

- To bring the benefits of automation to an additional portion of the test cycle.
- To provide testers with more effective tools to
 - Create test cases
 - Trace to requirements
 - Justify risk based decisions
 - Reduce cost and cycle time.



Model Based Testing Definitions

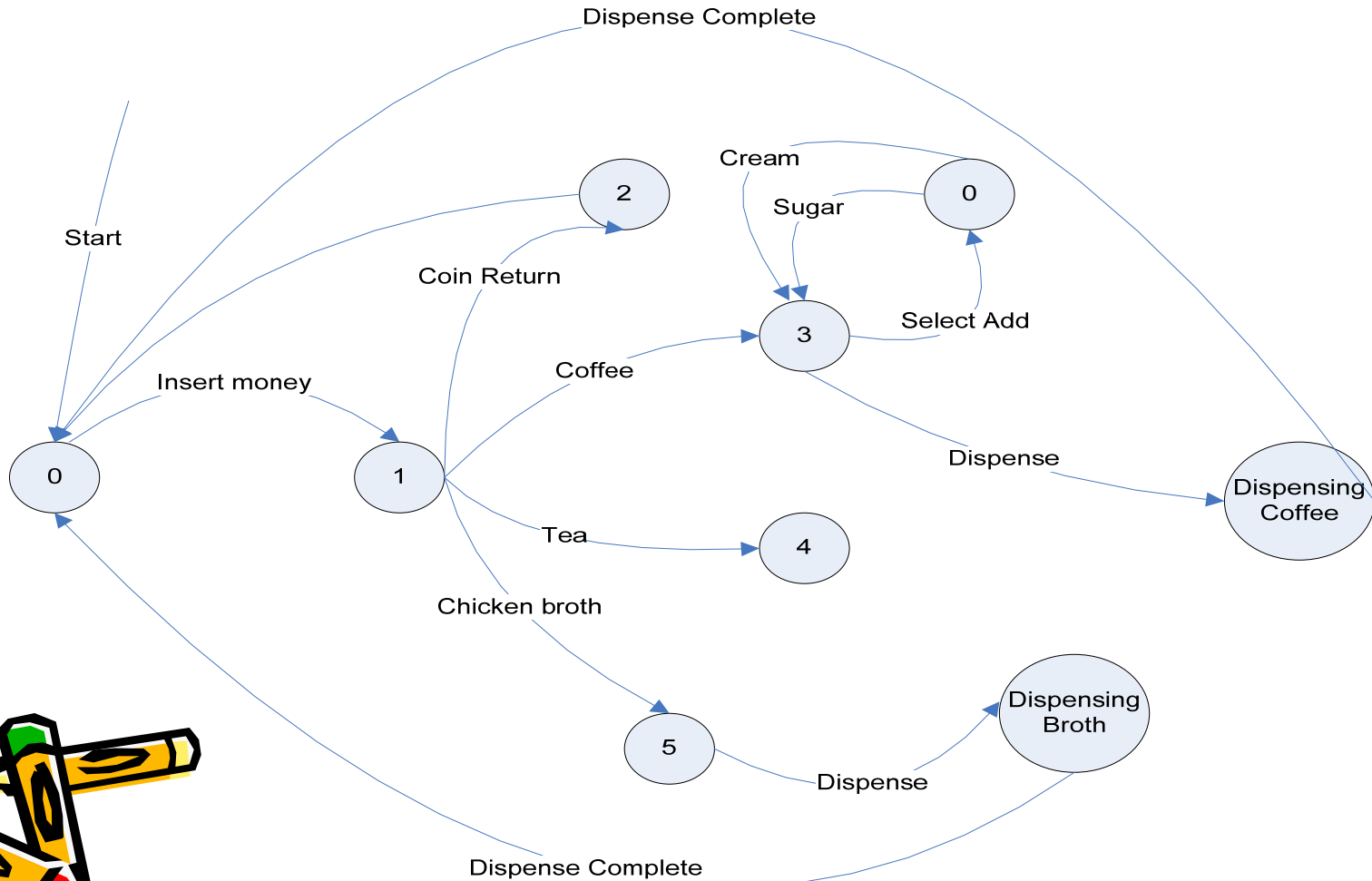
- MBT - A set of techniques and tools to automate the creation of test cases based on a model of a system.
- MODEL - typically a UML representation or a finite state machine that describes the behavior of a system (or part of a system).



What is a Model?

- A description of a system that helps us understand and predict its behavior
- It does not need to completely describe the system to be effective
- Two common representations for models are
 - Unified Modeling Language
 - Finite state machines

(FSM) Coffee Machine Model





Frequently Asked Questions

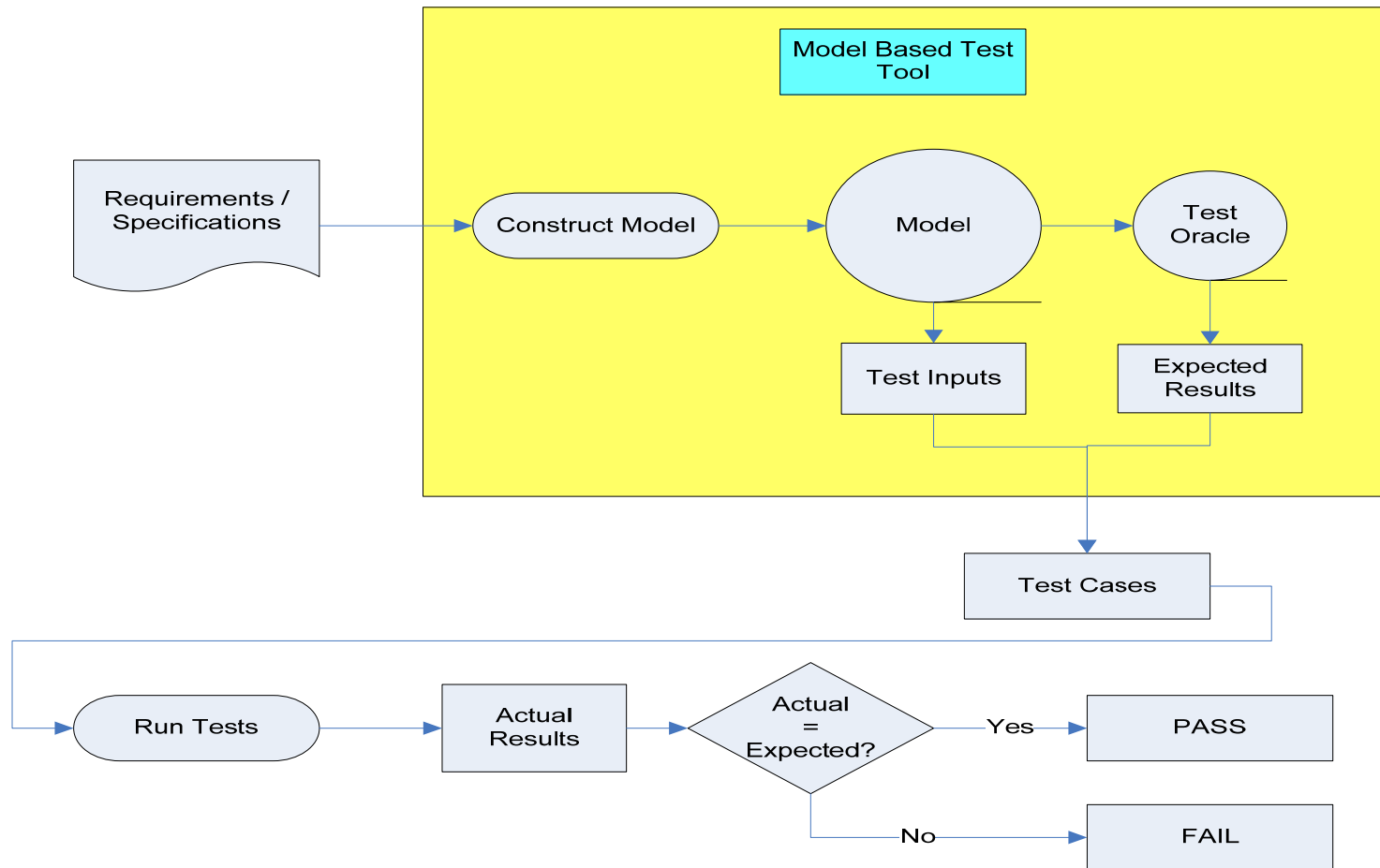
- Can you do MBT if Developers are not creating code from a model?
 - Yes. In that case testers create the model
- If there is a development model, can it be used as-is?
 - No. The test generation point of view is not the same. But a development model can be used as a starting point



Frequently Asked Questions

- How much work is it to create and use the model?
 - Modeling is a skill that needs to be learned. However in MBT it is not done in addition to test design. It replaces the effort of test design.
 - In conventional test design we work from a mental model of the application anyway. Here the model is externalized and preserved.

MBT Flow





Benefits of Model Based Testing

- Improved fault detection:
 - Studies compared Fault detection of manual vs. automatically generated tests (e.g. IBM, Microsoft, smart card industry)
 - Reported that the model based tests were equal or better at uncovering defects than the manual tests
- Reduction in cost and cycle time:
 - Model Based Testing can lower the cost of creating manual tests and/or automated scripts.
 - Once the model and the execution framework are in place, the model can be used to generate unlimited test cases, freeing human testers for more challenging test tasks such as exploratory testing.

Benefits of Model Based Testing

- Improved test quality and traceability:
 - With model based testing the quality of the tests is uniform, the design process is systematic and repeatable, and linkage of tests and requirements is clear.
 - Testers can choose the desired level of coverage when starting the test generation, based on risk and other factors.
 - The level of coverage is clearly documented
- Requirements Defect Detection:
 - Creation of the model requires a level of understanding of the requirements and specifications equal to or greater than is required for manual test design.
 - That can uncover inaccurate or incomplete requirements and foster cooperation between developers and testers. A Microsoft study showed that the modeling process uncovered twice as many design issues as implementation bugs.



Benefits of Model Based Testing

- Requirements evolution:
 - Model Based Testing lowers the cost of updating test suites for changing requirements.
 - Since the test creation is automated, a requirements change can be accommodated by making the corresponding change to the model, and re-generating the tests.
- Allows automation to be used for more than regression:
 - An MBT tool creates generic test cases that can be executed manually or tied to an automation framework for automated execution.
 - The need for manual test cases to precede automated test cases is removed or minimized



Benefits of Model Based Testing

- Improved fault detection:
 - Studies compared Fault detection of manual vs. automatically generated tests (e.g. IBM, Microsoft, smart card industry)
 - Reported that the model based tests were equal or better at uncovering defects than the manual tests
- Reduction in cost and cycle time:
 - Model Based Testing can lower the cost of creating manual tests and/or automated scripts.
 - Once the model and the execution framework are in place, the model can be used to generate unlimited test cases, freeing human testers for more challenging test tasks such as exploratory testing.



Tuning the test creation

- Parameters to the test generation
 - Path coverage
 - Happy path
 - All paths
 - Requirement coverage
 - Force all states marked as requirements to be visited
 - Type of test
 - Boundaries
 - All Pairs



Tuning the test creation

- Many more, list depends on the tool
- The point is
 - You can use the same model to produce
 - A smoke test
 - A thorough product test
 - An acceptance test
 - You can report rigorously the level of coverage delivered



Two main variants

- Offline MBT

- Test cases are generated and output in a specified format
- Execution is a separate step

- Online MBT

- Execution is done while the test steps are being generated



So, why is everyone not doing this?

- A huge leap from comfort zones
 - Testers
 - Test Leads
 - Project Managers
 - Managers

So, why is everyone not doing this?

- A significant knowledge gap
 - Testers require some level of programming and modeling skill
- Trust factor
 - Do I trust it to equal my design abilities?
 - Will it take away my job?



So, why is everyone not doing this?

- Schedule pressure
 - I can estimate my project – I can't estimate this twist
 - I have a drop dead date



Adoption Strategies

- POC
- Gather references from other tool users
- Prepare an economic justification – be realistic about benefits and costs of adoption
- Find a manager who will help to champion this innovation
- Generally tool suppliers will offer a knowledge transfer period with trained consultants



Adoption Strategies you Don't Want

- In response to a crisis
- In response to competitive disadvantage



Linking MBT to an Automated Execution Framework

- Directly output code in the scripting language of the automation tool
 - Various output options available, e.g. text for manual execution, QTP, Python, etc.
- In a keyword driven system, produce data files to be consumed by a script processing engine

Examples of MBT

- TDE UML (From Siemens Corporate Research)
 - Testers construct model based on requirements
 - Testers choose the breadth and depth of the test cases to be generated, and input various parameters to guide the tool. (Happy path to all paths)
 - Tool employs test design techniques such as boundary value analysis and all pairs testing
 - Coverage is predictable
 - Offline



Examples of MBT

- Conformiq Designer
 - Similar features and benefits
 - UML state charts and Java coding to define the model
 - Offline

Examples of MBT

- As used by Harry Robinson (Microsoft, later Google)
 - Thousands of tests are generated and implemented in automated execution scripts, which run overnight on a large array of inexpensive computers
 - The paths followed are guided by various heuristics such as
 - Shortest paths first
 - Most likely paths
 - Random walk
 - All paths of length N
 - All Transitions
 - Paths that find bugs are retraced in segments to isolate the minimum path to recreate the bug

References and Suggestions for Further Reading

- <http://www.amazon.com/Practical-Model-Based-Testing-Tools-Approach/dp/0123725011>
- <http://www.harryrobinson.net/>
- <http://www.robertvbinder.com/docs/arts/MBT-User-Survey.pdf>
- <https://www.goldpractices.com/practices/mbt/>
- <http://www.scribd.com/doc/9131626/Model-Based-Testing>