A comparison of free tools for Domain Specific Test Languages

Martin Gijsen
martin@DeAnalist.nl
Test automation architect

Overview

- A brief introduction to tools
- Domain Specific Test Languages
- How do tools and their approaches compare?
  - Wiki
  - Plain text
  - Keywords

© 2010 Martin Gijsen

A tool is a tool is a tool?

So ...

- Some tools are ineffective
- Some tools are effective if used well
- An inexpensive tool can still be effective

© 2010 Martin Gijsen

Automated functional testing

- The tool is not the main factor for success
- Focus on how to use it first
  - Consider processes and available skills
  - 3 R's: Easy to write, review & revise
  - Reduce testware maintenance
- Select a tool that supports your approach
  - Check that the tool has the features you need

© 2010 Martin Gijsen

Record & Playback

Good reasons to use the 'record' button to create an automated test are rare

A Domain Specific Test Language

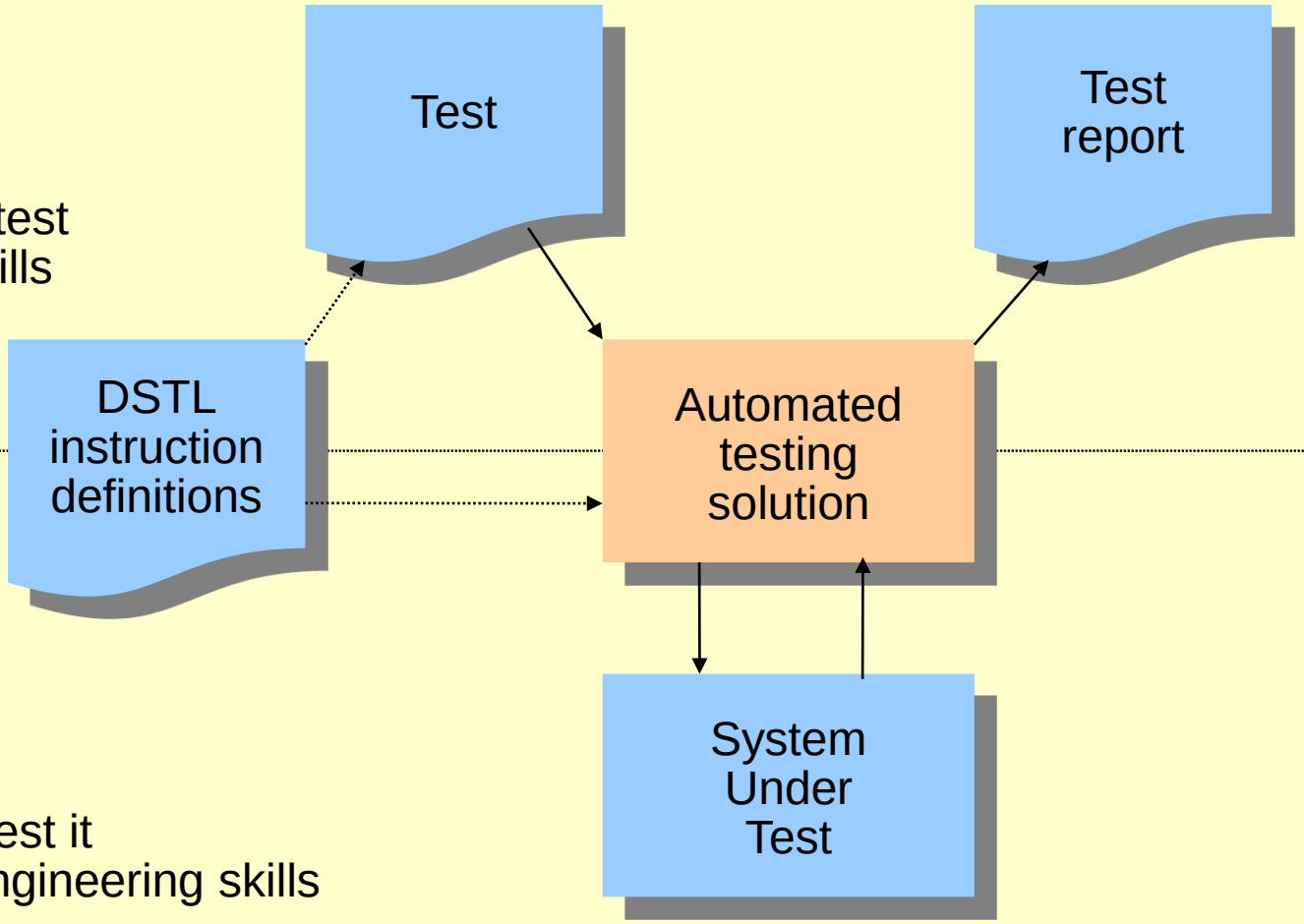
- Instructions for expressing test cases
- Specific to the system under test
- Defined by (or with) the testers
- Only the essence of test cases remains:
  - Natural, high abstraction level (business level)
  - No irrelevant test execution details
  - No irrelevant interfacing details
  - No tooling details
- Clear test report

© 2010 Martin Gijsen

The tester and automator roles

Tester domain:
- DSTL instructions

Focus is on ***what*** to test
Requires analysis skills



Automator domain:
- Scripting
- Source code

Focus is on ***how*** to test it
Requires software engineering skills

© 2010 Martin Gijsen

Keyword-driven

- Keywords with arguments
- Table format
- Easy to use
- Low level keywords lead to maintenance

open	/	
type	q	nederlandse testdag 2010
clickAndWait	btnG	
clickAndWait	link=leiden	

A non-trivial test case

Removing items from the web shop cart:

- Select a book and add it to the cart: book 1
- Select another book and add it to the cart: book 2
- Open the cart and check that the books are there
- Set the quantity for book 1 to zero
- Check that book 1 is gone and book 2 is still there
- Delete book 2 using the 'delete' button
- Check that the cart is empty

	item	*quantity*	*basket quantity*
add book to basket	book1	1	1
add book to basket	book2	3	4
open the basket			

	item	*quantity*	*item*	*quantity*
check quantities	book1	1	book2	3

	item	*quantity*	*basket quantity*
update quantity	book1	0	3

Natural language: Cucumber

Given I added book "book1" to the basket with quantity "1" and total quantity "1"

And I added book "book2" to the basket with quantity "3" and total quantity "4"

When I open the basket

Then the quantity for "book1" is "1"

And the quantity for "book2" is "3"

When I update the quantity for "book2" to "0"

Then the quantity for "book1" is "1"

And item "book2" is not listed

When I delete "book1"

Then item "book1" is not listed

And item "book2" is not listed

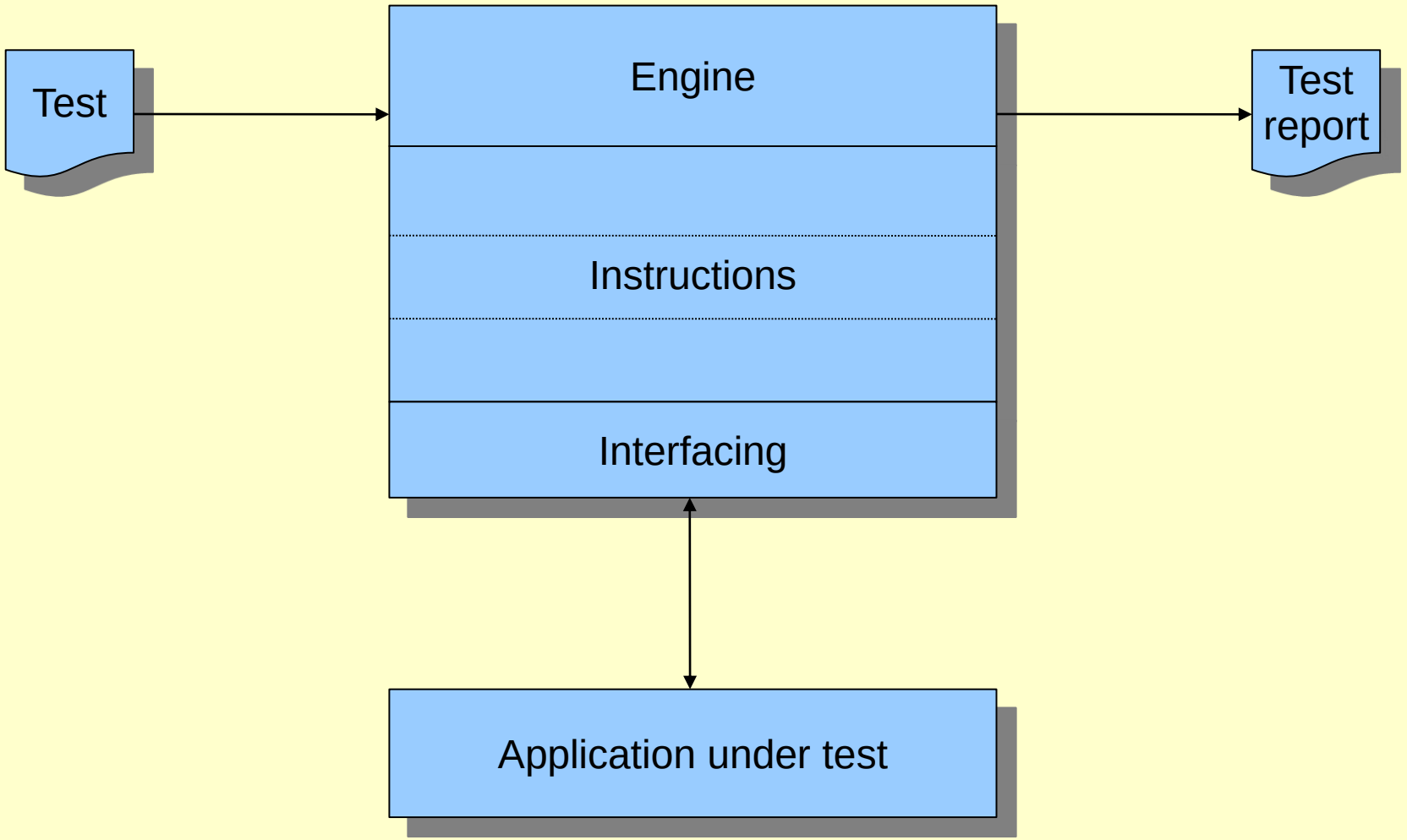
© 2010 Martin Gijsen

Wiki Based: FitNesse

Demo

© 2010 Martin Gijsen

Design of an ATS



Comparing the not-so-technical side

- The concept is the same: DSTLs
- Test-first / TDD are possible
- Comments are supported
- Reports are generated
- The form and editing differ
- Given/When/Then is mandatory for Cucumber
- Optional arguments are “” with Cucumber
- No expressions in Cucumber and FitNesse

Comparing the more technical side

- Continuous integration is supported
- Useful with interfacing tools like Selenium
- Version mgmt is easiest with text
- Very few tools support output parameters

Some freeware tools

- FitNesse: Slim test system, Wiki, Java/.NET/...
- Cucumber: 'natural language', Ruby/Java/...
- Rspec: 'natural language'
- RobotFramework: keywords (BDD), python
- Power Tools: keywords + 'natural language', Java

Conclusion

- DSTLs support an excellent approach to automated testing
- Let testers test and automators automate
- Free modern tools hit the nail on the head just fine
- Select those that suit your organisation

Q & A?

Thank you

© 2010 Martin Gijsen