



Domain Specific Languages for Acceptance Testing

Mats Ekhammar
Callista Enterprise AB



Agenda

- Acceptance tests
- Domain Specific language (DSL)
- Tooling
- Demo - Web GUI DSL
- Demo - Custom DSL
- Summary

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 2
Copyright 2007, Callista Enterprise AB



Acceptance tests

Common characteristics of acceptance tests today:

- Tests specified in text documents.
- Results specified in text documents.
- Performed on the applications user interface.
- Done by non-technical people from the business side.
- The whole application is tested, end-to-end.

Problems with the tests could be:

- Tests are manually and error prone.
- Takes a long time to complete.
- Requirement documentation, test documentation and test results are in different places.

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 3
Copyright 2007, Callista Enterprise AB



Acceptance tests wish list...

- Acceptance tests should be automated.
- Requirements (functional and non-functional), test documentation and tests results should be placed together.
- All tests should be easy to create and maintain.
- Easy to view result of acceptance tests.
- Tests should be expressed in a language in the context of the domain.



Domain Specific Language !

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 4
Copyright 2007, Callista Enterprise AB



Domain Specific Language (DSL)

Example 2:

Add 3 PCS of 2878172 To Order A1

Check Order Created with Reference REF12345

- The syntax describes how we can combine words, the structure of the language.
 - After Add we enter a number.
- The semantic describes the meaning of language expressions
 - What is going to happen when running a sentence?

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 5
Copyright 2007, Callista Enterprise AB



Domain Specific Language (DSL)

- The syntax should be easy to understand and use when writing the acceptance tests.
 - A generic table format as in FIT (Framework for Integrated Tests) is easy to use and FIT knows how to interpret this format.
- We also need a semantic that is intuitive in the domain where our acceptance tests are going to take place.
 - The freedom to define our semantic in domain specific terms is provided by FIT fixture construction.

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 6
Copyright 2007, Callista Enterprise AB



FIT – Framework for Integrated Tests

- FIT is a tool that helps writing automated acceptance tests.
- A FIT test use a table format, which is easy to understand. The result of the test when executed is also easy to verify.

CadecTest	
people	breaks()
0	0
10	1
24	1
25	2

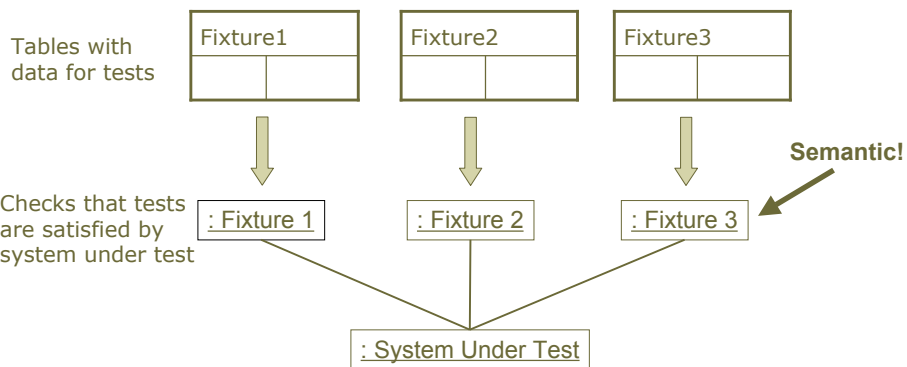
Syntax!

CadecTest	
people	breaks()
0	0
10	1
24	1 <i>expected</i>
	2 <i>actual</i>
25	2

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 7
Copyright 2007, Callista Enterprise AB



FIT



Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 8
Copyright 2007, Callista Enterprise AB



FitNesse

- FitNesse is an HTML "front-end" and wiki (web site) to FIT.
- Specify requirements, set up test tables and then run the tests on one wiki page
- Makes it easy to create, organize and share FIT tests.
- FitNesse will run all your tests and display the results in a nice way.



Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 9
Copyright 2007, Callista Enterprise AB



Web GUI DSL - demo

- Example of a technical approach to DSL.
- We want to record actions done on an applications GUI.
 - Select links, windows
 - Enter texts, select choices, check boxes, push buttons, ...
 - Verify texts, values, titles, alerts, selections,...
- Save them to Fitnesse for future use, automation.
- The recording of actions will be done by a tool called Selenium.

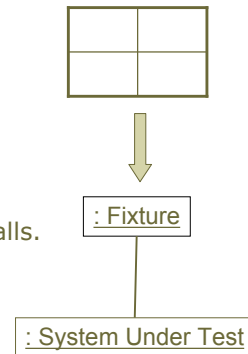
Where comes DSL in?

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 10
Copyright 2007, Callista Enterprise AB



Web GUI DSL - demo

- We record a test and paste it as a FIT table.
- To execute our FIT test we use a fixture that maps our recorded actions to Selenium remote calls.
This will be where we define our DSL!



Demo

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 11
Copyright 2007, Callista Enterprise AB



WEB GUI DSL - demo

- + Automated written acceptance test for a Web GUI!
- The tool (Selenium) defined our DSL's semantic, no enhanced semantic for a business user.
- Ended up with a technical and generic solution, not with a business specific domain language set.
- We need to create a more specific language that shield our business user from the inner steps of a test.

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 12
Copyright 2007, Callista Enterprise AB



Custom DSL - demo

The goal is to write a more customized DSL with a higher abstraction.

Example:

se.callista.cadec2007.CreateOrderFixture						
Create Order with Alias	A1					
Set Order Reference	REF12345	On Order	A1			
Set Order Class	DAYORDER	On Order	A1			
Add	3	PCS of	2878172	To Order	A1	
Add	5	PCS of	466634	To Order	A1	
Submit Order	A1					
Check Order Created with Reference	REF12345					

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 13
Copyright 2007, Callista Enterprise AB



Custom DSL – Fixture code - demo

The user writes the following in a FIT table:

Add	3	PCS of	466634	To Order	A1	
-----	---	--------	--------	----------	----	--



```
public class CreateOrderFixture extends DoFixture {
...
public boolean addPCSOftoOrder(int quantity, String
partNumber, String orderAlias) {
    selenium.open("/OrderEntry");
    selenium.type("orderBasketFld",
        (String)orders.get(orderAlias) );
    selenium.clickAndWait("openOrderBasket");
    selenium.type("inputPartNumberFld", partNumber);
    ...
}
```

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 14
Copyright 2007, Callista Enterprise AB



Custom DSL – Fixture code - demo

The user writes

Add	3	PCS of	466634	To Order	A1	
-----	---	--------	--------	----------	----	--

on the wiki page.



```
public class CheckOrderStatusFixture extends DoFixture {
...
public boolean addPCSOftoOrder(int quantity, String partNumber,
String orderAlias) {
return addPartsToDB(partNumber, quantity,
(String)orders.get(orderAlias));
}
```

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 15
Copyright 2007, Callista Enterprise AB



Custom DSL – Fixture code - demo

In the code we can do things like:

- Save reference to created objects in a static context.
- Call a script engine (Selenium) for execution in a browser
 - Enter texts, click links, ...
- Make calls to databases
 - Setup a test data environment, read db values, ...
- Use other resources
 - Mail, User repositories, Webservice calls, ...
- ...

[Demo](#)

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 16
Copyright 2007, Callista Enterprise AB



Custom DSL

- By using a Custom DSL with tooling we have reached all demands on future acceptance test stated in the beginning!
 - Acceptance tests should be automated.
 - Requirements (functional and non-functional) should be documented as close to the test documentation and tests as possible.
 - All tests should be easy to create and maintain.
 - Easy to view result of acceptance tests.
 - Tests should be expressed in a language in the context of the domain.

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 17
Copyright 2007, Callista Enterprise AB



Summary

By helping the customer to build a DSL for their own domain we can make it easier to create acceptance tests.

Combine the DSL with for example FIT, Fitnesse and Selenium and we get a fullfledged solution for our customers acceptance tests.

It will require some help from us developers...

Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 18
Copyright 2007, Callista Enterprise AB



Remember

Help your customers today to automate their acceptance tests with a language defined in their domain.



Cadec 2007, Domain specific languages (DSL) for acceptance testing, Slide 19
Copyright 2007, Callista Enterprise AB

