

Bypassing the GUI

Brian Marick

Problemer med GUI

- GUI er designet for mennesker, ikke automatisering
- Automatiseret test af GUI kræver specialiseret værktøjer
- Har tildens til at bryde ned når der sker ændringer med GUI
- At lave test modstandsdygtige overfor ændringer kræver komplicerede og til tider besværlige test frameworks.

Program interfaces der muliggør scripting

- COM (Component Object Model)
- SOAP
- AppleScript
- SAP GUI Scripting API
- Mange andre

Fordele ved at teste GUI via scripts

- Automatisering
- Tilføje nye test er nemt
- Rette test nemt
- Ændringer i GUI berører ikke den underliggende API
 - Man kan have forskellige GUI til sammen program uden at skulle lave flere tests
- Scripting interface mere stabilt end GUI

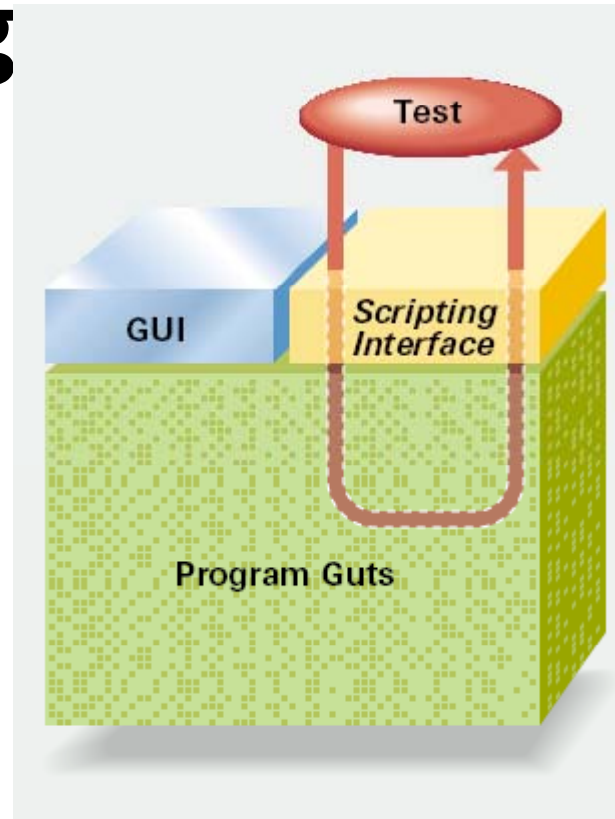
Fordele ved standart sprog

- Standart script sprog mere modne end de vendorscript der følger med de fleste GUI test værktøjer
- Dyrt at benytte GUI test værktøjer da de sjældent giver free runtime licenses
- Standart script sprog gratis
- Nemmere at få hjælp da der er mange brugere

Ideal situation for script test af GUI prog

GUI og script interface
virker begge som
tynde lag ovenpå
program kernen

Hvis GUI ikke er tynd
kan det give
problemer med
troværdigheden af
script test



Hvad kan gå galt?

- Hvis der ikke er et fuldstændigt API vil manuelle GUI testere ikke vide at der er dele der ikke bliver testet af scripts og dele kan derfor blive testet for lidt
- Hvis der er forskel i den kode der bliver kaldt af GUI og script API, vil den automatiserede test ikke finde fejl der forekommer ved brug af GUI
- Fejl i programmering af scripts
- Fejl i dokumentationen af API

Ruby eksempel

- Test af søge funktionen i MS Word
- Inkludering af biblioteker for at teste Word
 - Brug require
 - 'test/unit'
 - 'test/unit/ui/console/testrunner'
 - 'win32ole'

Ruby

- Ruby er et fortolket objektorienteret sprog
 - Kommer med to grænseflader
 - fxri – Instant Ruby Enlightenment
 - Kan benyttes til umiddelbar evaluering af udtryk
 - SciTE
 - Kombineret editor og resultat vindue.
 - Til udvikling og afprøvning af programmer

fxri

The screenshot shows the 'fxri - Instant Ruby Enlightenment' application. The left sidebar lists various array methods, with 'Array#<<' selected. The main window displays the documentation for 'Array#<< ob'.

Array#<< ob → array

Append--Pushes the given object on to the end of this array. This expression returns the array itself, so several appends may be chained together.

```
[ 1, 2 ] << "c" << "d" << [ 3, 4 ]  
=> [ 1, 2, "c", "d", [ 3, 4 ] ]
```

irb(main):001:0> require 'win32ole'
=> true
irb(main):002:0> Word = WIN32OLE.new('Word.Application')
=> #<WIN32OLE:0x568ac58>
irb(main):003:0> Word.visible = true
=> true
irb(main):004:0> Word.document
WIN32OLERuntimeError: unknown property or method 'document'
HRESULT error code:0x80020006
Ikendt navn.
from (rb):4:in 'method_missing'
from (rb):4
from ^C:0
irb(main):005:0> Word.documents
=> #<WIN32OLE:0x5a45d58>
irb(main):006:0> Word.documents.count
=> 0
irb(main):007:0> Word.documents.add
WIN32OLERuntimeError: add
OLE error code:800A17C0 in Microsoft Word
Den funktion du foreskriver indeholder makroer eller data, som Word understøtter af makroprog. Du (eller administratoren) har valgt ikke at installere understøttelse af makroer eller kontrolelementer, da programmet blev installeret.
HRESULT error code:0x80020009
Undtagelse opstod.
from (rb):7:in 'method_missing'
from (rb):7
from ^C:0
irb(main):008:0> Word.selection.typtext('hej alle')
=> nil
irb(main):009:0>

128 / 13438 entries

Sceti

```
TestWord.rb - SciTE
File Edit Search View Tools Options Language Buffers Help
1 TestWord.rb

8   # delete a first blank line.
9   gsub(/^[ \t]*/, "").gsub(/\n[ \t]*/, "\n")
10  end
11  def indices(string)
12    res = Array.new
13    index = index(string, 0)
14    while index
15      res = res.push(index)
16      index = index(string, index + 1)
17    end
18    puts res
19    return res
20  end
21  end
22
23  class TestWord < Test::Unit::TestCase
24    # Sæt variabelen Word lig en MS Word applikation
25    Word = WIN32OLE.new 'Word.Application'
26    # Nyt dokument i Word objektet
27    Word.documents.add
28    # Execute the given file using the associate app
29    def test_find_boundaries
30      original =
31        'MATCH at beginning of file
32         MATCH at beginning of interior line
33         at end of interior line, we find a MATCH
34         at end of file, we find a MATCH'
35      original = original.without_left_whitespace
36      # fjern blank tegn fra pæn opsætning
37      Word.selection.typetext(original)
38      # Placer cursoren i starten af dokumentet
39      Word.selection.start=0
40      Word.selection.end=0
41      # Fortæl Word at den skal finde teksten MATCH
42      Word.selection.find.text = 'MATCH'
43      # original.indices finder de steder hvor teksten findes så vi kan sammenligne med Word resultat
44      for start in original.indices('MATCH')
45        # Næste resultat fra Words søge funktion
46        result = Word.selection.find.execute
47        # Test at Word har fundet teksten
48        assert_equal(true, result)
49        # Test at det vi og Word har fundet starter sammenstet
50        assert_equal(start, Word.selection.start)
51        # Test at det Word har fundet har den rigtige længde
52        assert_equal(start+5, Word.selection.end)
53        # Test at Word har fundet den rigtige tekst
54        assert_equal('MATCH', Word.selection.text)
55      end
56    end
57    # Kør testen
58    Test::Unit::UI::Console::TestRunner.run(TestWord.suite)
59    # close(false) gør at vi ikke bliver bedt om at gemme dokumentet
60    Word.documents.close(false)
61    Word.quit
62  end

>ruby TestWord.rb
Loaded suite TestWord
Started
0
27
98
130
.
Finished in 0.531 seconds.

1 tests, 16 assertions, 0 failures, 0 errors
>Exit code: 0
>ruby TestWord.rb
Loaded suite TestWord
Started
0
27
98
130
.
Finished in 1.016 seconds.

1 tests, 16 assertions, 0 failures, 0 errors
>Exit code: 0
>ruby TestWord.rb
Loaded suite TestWord
Started
0
27
98
130
.
Finished in 0.437 seconds.

1 tests, 16 assertions, 0 failures, 0 errors
>Exit code: 0
>ruby TestWord.rb
Loaded suite TestWord
Started
0
27
98
130
.
Finished in 0.422 seconds.

1 tests, 16 assertions, 0 failures, 0 errors
>Exit code: 0
```

Test af Words søgefunktion

```
class TestWord < Test::Unit::TestCase
  #Sæt variablen Word lig en MS Word applikation
  Word = WIN32OLE.new 'Word.Application'
  #Nyt dokument i Word objektet
  Word.documents.add
  # Execute the given file using the associate appdef test_find_boundaries
  original =
    'MATCH at beginning of file
    MATCH at beginning of interior line
    at end of interior line, we find a MATCH
    at end of file, we find a MATCH'
  #fjern blank tegn fra pæn opsætning
  original = original.without_left_whitespace
```

```
Word.selection.typetext(original)
```

```
#Placer cursoren i starten af dokumentet
```

```
Word.selection.start=0
```

```
Word.selection.end=0
```

```
#Fortæl Word at den skal finde teksten MATCH
```

```
Word.selection.find.text='MATCH'
```

```
#original.indices finder de steder hvor teksten findes så vi kan sammenligne  
med Word resultat
```

```
for start in original.indices('MATCH')
```

```
#Næste resultat fra Words søge funktion
```

```
    result = Word.selection.find.execute
```

```
#Test at Word har fundet teksten
```

```
    assert_equal(true, result)
```

```
#Test at det vi og Word har fundet starter sammensted
    assert_equal(start, Word.selection.start)
#Test at det Word har fundet har den rigtige længde
    assert_equal(start+5, Word.selection.end)
#Test at Word har fundet den rigtige tekst
    assert_equal('MATCH', Word.selection.text)
end
end
#Kør testen
    Test::Unit::UI::Console::TestRunner.run(TestWord.suite)
#close(false) gør at vi ikke bliver bedt om at gemme dokumentet
    Word.documents.close(false)
    Word.quit
end
```