Specification of the Test Description Language*

Łukasz Olek, Bartosz Alchimowicz, Jerzy Nawrocki

This document describes the syntax of the Test Description Language (TDL). For the sake of readability, elements such as white spaces (used for indents) or new lines were omitted. The specification below uses three terminal symbols:

- `identifier` – a non-empty string build from the following characters: letters, numbers, underscores, spaces; this terminal symbol starts with a letter, if an identifier has spaces then it need to be put in quote characters, e.g. `UC11`, `System`, `Main_screen`, `"Main screen"`;
- `text` – a non-empty string containing any characters except new lines. A double quote character needs to be escaped, by another double quote character;
- `number` – any natural number (regular expression is `[0-9]+`).

```
TestCases -> TestCase*

TestCase -> 'TESTCASE' Tag Description
                Precondition
                TestStep+

TestStep -> 'TESTSTEP' Step_Tag ['(' Screen_Label ')' ] ':'
                Assertion*
                User_Action*
Precondition -> Precondition_Term Tag (',' Tag)*
              | Precondition_Term ':'
                    Precondition_Actions*
Precondition_Term -> 'PRECONDITION' | 'SETUP'

Assertion -> Basic_Assertion | Semi_Compound_Assertion | Compound_Assertion
Basic_Assertion -> 'CHECK' Component_Label '=' Value_Or_RegExp
Semi_Compound_Assertion -> 'CHECK' Component_Label ':'
                                Semi_Compound_Predicate+
Semi_Compound_Predicate -> Sorted_Predicate
                         | Selected_Predicate
                         | Expr_Predicate

Sorted_Predicate -> 'SORTED' ['DESC']
Selected_Predicate -> 'SELECTED' '(' Value_Or_RegExp ')'
Expr_Predicate -> Simple_Expr Relational_Operator Simple_Expr
Simple_Expr -> [Addition_Operator] Term [Addition_Operator Term]
Term -> Factor [Multiplication_Operator Factor]
Factor -> Number | Function | Value
Relational_Operator -> '=' | '<>' | '<' | '>' | '<=' | '>='
Addition_Operator -> '+' | '-'
Multiplication_Operator -> '*' | '/'
Function -> 'POS' '(' Regular_Expr ')'
          | 'COUNT' '(' Regular_Expr ')'
          | 'COUNT'
          | 'VAL' '(' Number ')'

Compound_Assertion -> CHECK Component_Label ':'
                          Quantifier*
                          Sorted_Compound_Predicate*
                          Count_Compound_Predicate?

Quantifier -> 'FOR' 'INDEX' '=' Number [Labeled_As] ':'
                    Assertion_List
Quantifier -> 'FOR' 'ONE' 'INDEX' [Labeled_As] ':'
                    Assertion_List
Quantifier -> 'FOR' 'ANY' 'INDEX' ':'
                    Assertion_List
Quantifier -> 'FOR' 'ALL' 'INDEX' ':'
                    Assertion_List

Assertion_List -> Assertion_List Assertion
Assertion_List -> empty
Sorted_Compound_Assertion -> 'SORTED' Component_Label
Count_Compound_Predicate -> 'COUNT' '=' Number
Labeled_As -> 'LABELED' 'AS' Pointer_Label

User_Action -> Click_Action | Set_Value_Action | Set_Multiple_Values_Action
```

*Appendix for *Acceptance Testing of Web Applications with Test Description Language*

```
            | Set_Checked_Action | Set_Multiple_Checked_Action
            | Select_Value_Action | Select_Multiple_Values_Action

Click_Action -> 'CLICK' Component_Ref
Set_Value_Action -> 'SET' Component_Ref '=' Value
Set_Multiple_Values_Action -> 'SET' Component_Ref Values
Set_Checked_Action -> 'SET' ('CHECKED' | 'UNCHECKED') Component_Ref
Set_Multiple_Checked_Action -> 'SET' ('CHECKED' | 'UNCHECKED') Component_Ref
                                  Values
Select_Value_Action -> 'SELECT' Component_Ref '=' Value
Select_Multiple_Values_Action -> 'SELECT' Component_Ref Values

Precondition_Actions -> Run_Test_Action | Run_Script_Action | Sleep_Action
Run_Test_Action -> 'RUNTEST' Tag
Run_Script_Action -> 'RUNSCRIPT' Label
Sleep_Action -> 'SLEEP' Number

Component_Ref -> [Pointer_Label '.'] Component_Label
Values -> '(' Value (',' Value)* ')'
Value_Or_RegExp -> Value | Regular_Expr
Value -> '"' Text '"'
Regular_Expr -> '/' Text '/'

Step_Tag -> identifier
Screen_Label -> identifier
Pointer_Label -> identifier
Component_Label -> identifier
Tag -> identifier
Label -> identifier
Description -> text
Number -> number
```

Acknowledgement

This work has been partially supported by the Polish National Science Centre based on the decisions DEC-2011/03/N/ST6/03016.