Zero-Defect Widgets

A formal approach to testing Xt-based toolkits

Satyajit Nath

Sun Microsystems, Inc.

Widget behavior

Picture of the CDE Application Builder Main Window. Please contact `jit@eng.sun.com` *for the picture.*

Specifications are available

... provides a single line text editor ... used for single-line string entry ...
... allows the user to select regions of text ...

New Resources

XmNcolumns *Specifies the initial width of the text window ... The value must be greater than zero ...*

Can we start test development now?

- *Need concrete implementation for hand-crafted tests*
- *Partial implementations hard to work with*
- *Cannot react quickly to design changes*

Is there a way to overcome these problems?
Yes, it is ADL Technology.

ADL (Assertion Definition Language)

- *framework for specification and automated test generation*
- *writer decides the level of abstraction*
- *allows partial specifications*
- *behavior expressed as a set of post-condition assertions*
- *syntax is very close to 'C'.*

Example ADL specification

attribute *Pixel tf_background;*
⋮
Widget Set_XmNbackground(Pixel background)
semantics {
    tf_background == background,
    window_background(text_field) == background
};

Another example

```
attribute Pixel tf_columns;
⋮
Widget Set_XmNcolumns(long columns)
semantics {
      failure       := xt_warning_generated(),
      columns < 0 <:> (xt_warning() == NEG_VALUE),
      failure       --> unchanged(tf_columns),
      normally      (tf_columns == columns)
};
```


GUI ADL specification

```
Widget UA_TypeInChar(string typein_char_value)
semantics {
      tf_cursorPosition == @tf_cursorPosition + 1,
      equal_strings(tf_value,
         concat_strings(
            @(value_prefix(tf_cursorPosition)),
            typein_char_value,
            @(value_suffix(tf_cursorPosition))))
};
```

How does we get tests?

Assertion Checking Functions *are generated.*

Store call-state information

Call function being tested

Evaluate all assertions
Report results

Test Data

- *Specified using the TDD (Test Data Description) language*
- *Test data described symbolically*
- *Not a list of test cases; rather their* characteristics
- Interesting *and* relevant *properties of the* test variables
- *Test developers supply* provide functions *to instantiate these properties.*

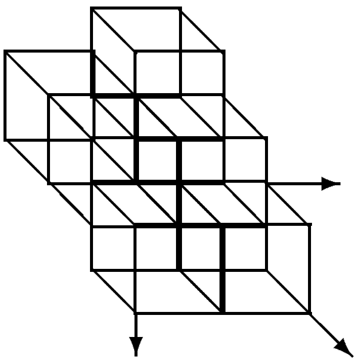
Example Test Data Specifications

prop *background_value_type* =
  { *light_bg*, *dark_bg*, *bg_same_as_fg* }
prop *char_pos_type* =
  { *typein_starting*, *typein_middle*, *typein_end* }

An example provide function:

```
bool provide_typein_chars_value(
        string              &typein_chars_value,
        prop_char_pos_type  prop_char_pos,
);
```


Test data + Specification ⟶ Generated Tests



Grid-points defined by test properties where ACFs are called.

Conclusion

$$Claim : \lim_{ADL\_use \to \infty} \mid Defects \mid = 0$$

- *Testing can start along with development using the same input*
- *Bug fixing will be integral to development — as partial implementations arrive, partial testing will happen*
- *Lower cost of bug fixing: will be on code "fresh" to the developers*
- *Design will gain from the process of formal specification*

Project background

- *Managed by X/Open*
- *Financed by MITI/IPA Japan, SMLI, X/Open*
- *Research and development done by SMLI*
- *Deliverables are open for industry review and freely available*
- *Currently, design documents are available*
- *Prototype implementation expected by March*

Sources for further information

- *Sun Microsystems Laboratories, Inc. Please contact Alberto Savoia (*`alberto.savoia@sun.com`*), Manager, Prima Vera group.*
- *X/Open Co. Ltd. Please contact Shane McCarron (*`s.mccarron@xopen.co.uk`*).*
- *Documents and reference implementations are publicly available by anonymous ftp from* `ftp.uu.net:/vendor/adl`*.*