is there a relationship between the aesthetic impression of a program and its quality?

Lodewijk Bergmans
[bergmans@acm.org]

Note

please consider that this is premature work, under investigation, only partially in my area of expertise..

(but it’s a fun topic, so contact me for more information)
Software quality

in particular:
comprehensibility of program specifications
• source code
• diagrams (e.g. class diagrams)

motivation:
• maintenance is 60-90% of total development effort
• during maintenance, software engineers spend between 30% and 60% of time on reading code
• on average 10 generations of maintenance programmers

This is a talk about: quality of software

UNIVERSITY OF TWENTE.
Function get_words_from_a_number_which_is_passed_as_a_perimeter_into_this_function _
(ByVal p_mode As String, ByVal p_numerical_value_equiv As String) As String

'Establish p_mode to define correct perimeter
If p_mode = "" Then
    p_mode = "No perimeter was passed through"
Else
    p_mode = p_mode
End If

If p_mode = "No perimeter was passed through" Then
    p_mode = ""
Else
    If p_mode = "" Then
        p_mode.Replace("", "No perimeter was passed through")
    Else
        p_mode = p_mode
    End If
End If

Response.Write(p_mode)

UNIVERSITY OF TWENTE.

examples

• in source code
• in (class) diagrams

source code example

[The Daily WTF]

UNIVERSITY OF TWENTE.
source code example—cont’d

Dim HK_373 As String = "pass_a_number"
If p_mode = HK_373 Then
    Dim NUMBER_CONVERTED_FROM_STRING As Integer = Convert.ToInt32(p_numerical_value_equiv)
    Dim WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER As String
    If NUMBER_CONVERTED_FROM_STRING = "1" Then
        WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER = "zero"
    ElseIf NUMBER_CONVERTED_FROM_STRING = "1" Then
        WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER = "one"
    ElseIf NUMBER_CONVERTED_FROM_STRING = "2" Then
        WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER = "two"
    ElseIf NUMBER_CONVERTED_FROM_STRING = "3" Then
        WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER = "three"
    ElseIf NUMBER_CONVERTED_FROM_STRING = "4" Then
        WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER = "four"
    ElseIf NUMBER_CONVERTED_FROM_STRING = "10" Then
        WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER = "ten"
    Else
        WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER = "Exceeds word range."
        End If
    Return WORD_CONVERTED_FROM_A_NUMBER_PASSED_THROUGH_A_PERIMETER
Else
    Return "This is a invalid perimeter passed through to this function. Please try again!"
End If

UML class diagram example 1
Another lesson I have learned is to distrust beauty. It seems that infatuation with a design inevitably leads to heartbreak, as overlooked ugly realities intrude. Love is blind, but computers aren’t. A long term relationship, maintaining a system for years, teaches one to appreciate more domestic virtues such as straightforwardness and conventionality.

Measuring program comprehensibility

examples of existing work:
• counting amount of comments [Lin 2006]
  • \( \rightarrow \) partially reverse with clean code intentions
• measuring spatial complexity
  • i.e. how far away (in LoC) are definitions and usage
  • \( \rightarrow \) partially reverse with separation of concerns
• measuring cognitive weight of language constructs
  • sequence(1) .. branch(2) .. loop(3) .. recursion(3) .. concurrency(4)
  • and then computing total/average for program.
• not very satisfactory....

This is not a talk about: art

art is mostly concerned about conveying messages that have emotional and intellectual impact, including:
• romance, anger, disgust, ..
• violence, sex, ...
• surprise, comedy, shock, ..
• beauty
• ...

UNIVERSITY OF TWENTE.
Nature of beauty

beauty $\rightarrow$ aesthetics

- aesthetic judgement involves aspects that are:
  - sensory
  - emotional
  - intellectual

we focus on intellectual beauty, or:
‘perfect form’
emotional

intellectual
perfect form (?)

is ‘perfect form’ objective?

even when disregarding personal taste & emotional interpretation: “beauty is in the eye of the beholder”

it is a combination of:
• the properties of an object (objective)
• and a mental model of the object by the observer (subjective)

now we discuss a theory about mental experience of perfect form
• from cognitive science
Mental models

we can imagine a mental representation of an object:
• as a vector: \( D = \{d_1, d_2, \ldots, d_n\} \)
• where each \( d_i \) can be:
  • activation level of a specific neuron (low-level)
  • an (more abstract) aspect of \( D \) (high-level)

instead of enumeration of \( \{d_1, d_2, \ldots, d_n\} \):
\( \Rightarrow \) can write a program \( D \) that generates \( \{d_1, d_2, \ldots, d_n\} \)
mental models –2

equals creating an internal mental model D

• cf. encoding/compressing the scene into a suitable vector or program
• involves a mental encoding procedure/method M
  • which may depend on personal experience & context
• this expresses that: observation is subject to personal experience!

Kolmogorov complexity

Kolmogorov complexity of an object \( x \); \( K(x) \):

• length of shortest code in a universal programming language
  • this is largely independent of precise language [Kolmogorov]

Conditional Kolmogorov complexity \( K(y \mid x) \):

• length of the shortest code that yield \( y \), given \( x \) as an input
Jürgen Schmidhuber

Schmidhuber’s theory of beauty

- among several sub-patterns $x_i$ classified as comparable
  - by a given observer
  - with a current particular method for encoding and memorizing
- the subjectively most beautiful is the one with
  - the simplest (shortest) description $D$
    - e.g. as expressed by its Kolmogorov complexity
- $\Rightarrow$ ‘the simplicity principle’:
  - the cognitive system prefers a simple, non-redundant code
example: faces
low-complexity face

= averageness hypothesis ??
composed average faces considered more attractive...
‘femme fractale’

Hypothesis
finally: my hypothesis

code that ‘looks beautiful’:
- according to schmidhuber:
  - can be easily represented by a simple mental model
- but that also applies equally to program comprehension!!!
  - that same mapping process will help to understand the essence
  - and see the program as a delta to some internal mental model
  - also ‘explains’ role of experience, common patterns
- hence: ‘beautiful code’ --> comprehensible code --> high-quality code

conclusion

• can argue that there is a logical relation between
  • beautiful code
  • comprehensible code
• perhaps the theory of beauty can be used to reason about program comprehensibility
  • but how does that take experience into account?
• cognitive science has proposed additional rules for beauty that may also be used
  • e.g. harmony/symmetry (⇔ low complexity?)
  • variety (interestingness: i.e. learning something new)
important references

- Jürgen Schmidhuber:
  - http://www.idsia.ch/~juergen/beauty.html

- Walter Kintsch: