

# **“Safer Language Subsets”**

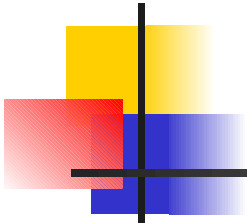
by

Les Hatton

The Computing Laboratory, University of Kent  
L.Hatton@ukc.ac.uk, lesh@oakcomp.co.uk

Version 1.1: 29/Apr/2002

# Overview

- 
- 
- Some history
  - Taxonomy
  - Enforceability
  - Special issues with C
  - Future direction

# Some history



---

**Safer subsets (also mixed up with programming standards), have been around for a long time**

- Fortran subsets have been around for 20 years
- Ada subsets, (Spark ...) have also been around for a long time.
- C subsets do not appear to have been so well-entrenched,

# Taxonomy

## Define the following taxonomy

- *Category A*: Purely stylistic stuff like indentation standards, variable naming conventions and so on.
- *Category B*:
  - B.1: Rejection of features based on *folklore*, for example the *goto* statement
  - B.2: Rejection of features based on known failures through measurement, for example type conversions which lose bits.

# Definition of a safer subset



---

**A safer subset is constructed from:-**

- Category B.2 rules as mandatory
- Category B.1 rules as recommendations only

# Programming standards make poor safer subsets

**N = (B/A) for a number of commercial standards**

Safety-related	Number of pages	B/A
Yes	15	5%
No	75	0%
No	66	8%
Yes	37	2%
No	33	35%
Yes	17	6%
No	18	8%

# Enforceability



---

**Rules in a safer subset ideally must be worded to have the following properties:-**

- Orthogonality (lack of interaction with other rules)
- Decidability (knowing what the rule means precisely)
- Atomicity (A rule can either be enforced or not.)

# Enforceability is hopeless in practice

**$N = (B/A)$  for a number of commercial standards**

Language	Safety-related	Transgression rate
C	Yes	1 every 14 lines
C	No	1 every 12 lines
C	No	1 every 121 lines
Fortran	No	1 every 66 lines
Fortran	No	1 every 771 lines

# Special issues with C



---

- Measurement support and what can be achieved
- Standardisation problems
- Influential nature of language
- Existence of MISRA C

# Measurement support



---

## **The good news**

- There is an abundance of measurement-supported information to assist in the construction of  $C$  safer subsets.

## **The bad news**

- People don't in general use it
- The language needs subsetting more than most.

# Two examples of what can be achieved in C



---

## **CDIS – component of LHR air-traffic control system**

- 400 KLOC source format lines
- Asymptotic defect density 0.71 per KLOC, (after 4 years in field).

## **Safer C toolset**

- 125 KLOC source format lines
- Asymptotic defect density 0.26 per KLOC, (after 3 years in field).

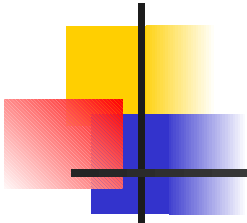
# Standardisation problems



---

- Like most standardised languages, each successive ISO revision generally creates more problems for safer subsetting than its predecessor. C99 is *no* exception.
- Language committees are too interested in features and not interested enough in reliable features.

# Influential nature of language



---

- C has spawned many languages such as C++, Java, Javascript, PHP, Perl, Tcl, Objective C.

The good and bad news ...

- This means that most of these languages have inherited a large subset of C's known defects, but at least the same principles can be used in rendering them safer.

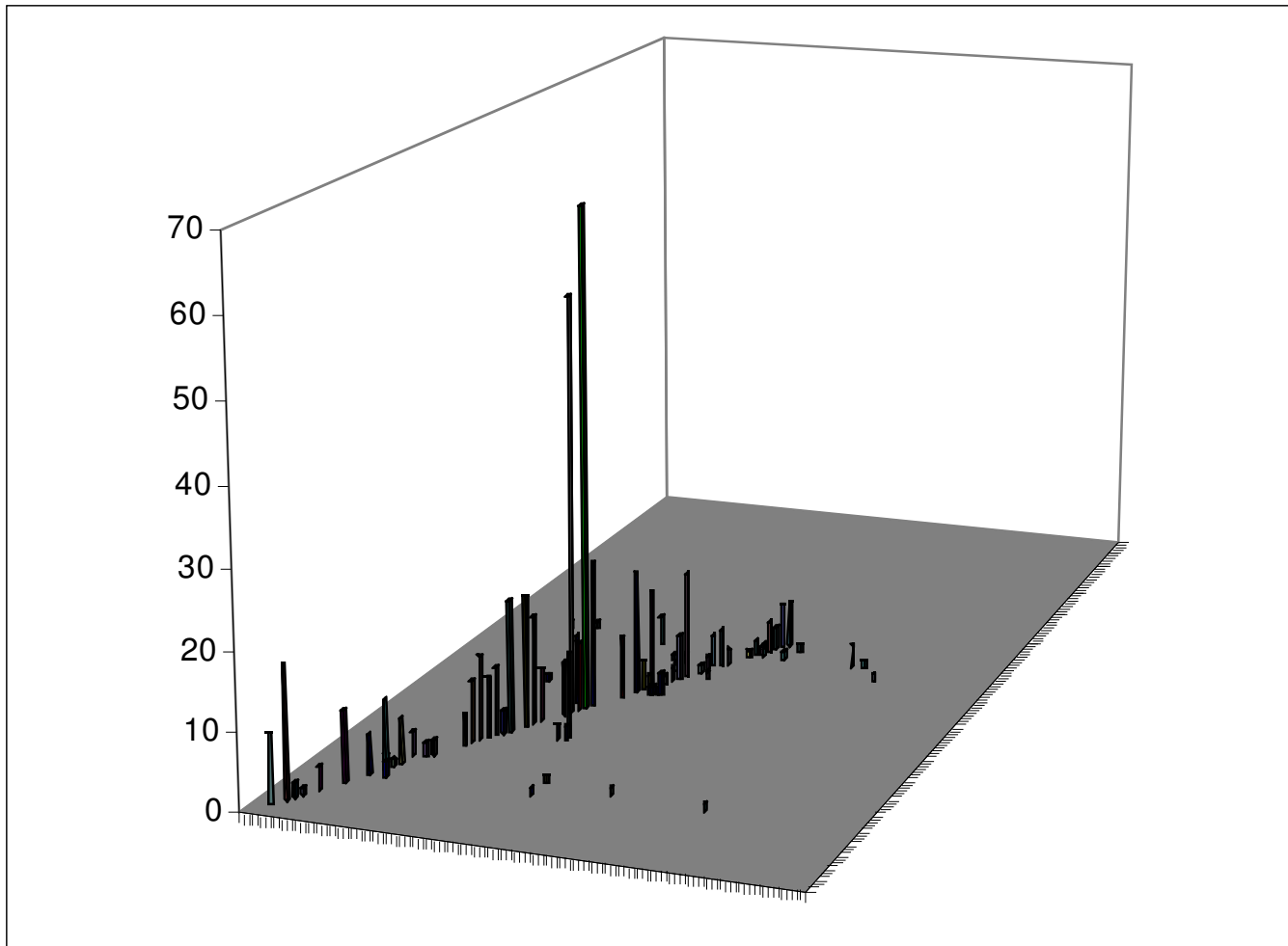
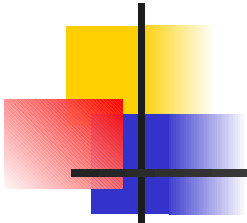
# Existence of MISRA C



---

- In April 1998, the auto industry decided that a safer subset of C was necessary. MISRA C was the result.
  - 127 rules, B/A close to 100%
  - Some problems with decidability, atomicity and cross-talk but better than previous attempts. However these currently mean that enforceability can not be satisfactorily defined. Test cases remain test cases and cannot define validation.

# Rule crosstalk so far ...



# Initiatives for improvement



---

## **MISRA**

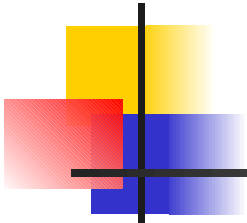
- Steering committee now attempting to address decidability, atomicity, orthogonality and test case implementation questions.

## **UKC**

- Large project starting to attack safer subsetting and enforceability in the C-like languages from a number of directions, both theoretical and empirical. It will work closely with the MISRA steering committee but has a broader scope.

In both cases, test case deliverables will be available under the GPL.

# What do you do about C++ ?



---

## **There are two possible approaches:-**

- Through folkloric approaches such as EC++, i.e. accepted but non measurement supported practice.
- Through measurement based on careful extensions from C. The problem here is that OO in general and C++ in particular is a measurement-free zone.

# UKC safer subset site (under extensive development)



---

## Download sites:-

- The as yet incomplete exemplary test case suite for MISRA C is available at the UKC national Safer Language Subsets project site:-  
<http://www.cs.ukc.ac.uk/national/SLS>