The Effect of Program and Model Structure on MC/DC Test Adequacy Coverage

ICSE '08: Proceedings of the 30th international conference on Software engineering

Ajitha Rajan
Mats P.E. Heimdahl
Dept. of Comp. Sci. and Eng.
University of Minnesota

Michael W. Whalen
Advanced Technology Center
Rockwell Collins Inc.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Arno Fiva
31.03.2009

MC/DC as a coverage metric for testing

- MC/DC widely used in critical systems such as in avionics or military
- **Paper states MC/DC criteria can be “cheated” and heavily depends on code structure**

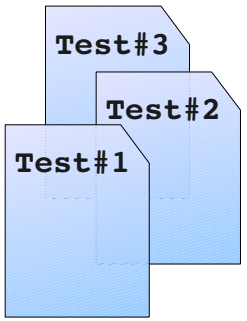
What is MC/DC?

Modified Condition/Decision Coverage

=> Source code metric for measuring the quality of a test suite

Condition Coverage + Decision Coverage + Each condition must independently affect the outcome of a decision

Test Suite



Implementation

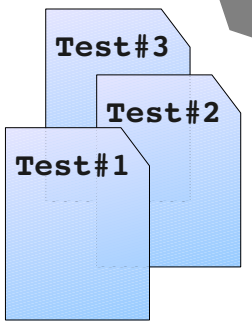
```
int myFunc (bool c1, bool c2, bool c3)
{
    bool d1 = c1 or c2;
    bool d2 = d1 and c3;
    if (d2)
        return 1;
    else
        return -1;
}
```

MC/DC example

```
int myFunc (bool c1, bool c2, bool c3)
{
    bool d1 = c1 or c2;
    bool d2 = d1 and c3;
    if (d2)
        return 1;
    else
        return -1;
}
```

c1	c2	d1 = c1 or c2
F	F	F
F	T	T
T	F	T
T	T	T

d1	c3	d2 = d1 and c3
F	F	F
F	T	F
T	F	F
T	T	T



Test Suite

Subset of all possible input tuples which satisfies MC/DC criteria

For Example: **{TFF, FTF, FFT, TTT}** // (c1 c2 c3)

Problems with MC/DC

Same program written in a different way (**d1** has been **inlined**)

```
int myFunc (bool c1, bool c2, bool c3)
{
    bool d2 = (c1 or c2) and c3;
    if (d)
        return 1;
    else
        return -1;
}
```

c1	c2	c3	d2 = (c1 or c2) and c3
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	T
T	F	F	F
T	F	T	T
T	T	F	F
T	T	T	T

Previous test suite does not satisfy MC/DC criteria anymore!

If correct expression should have been

```
bool d2 = (c1 and c2) and c3;
```

bug will not be revealed by current test suite!

Problem: A test suite satisfying MC/DC criteria would have detected fault, which shows that MC/DC coverage can be affected by program structure.

Case examples

Goal: show that test suite providing MC/DC over non-inlined version will achieve lower MC/DC over implementation that is inlined.

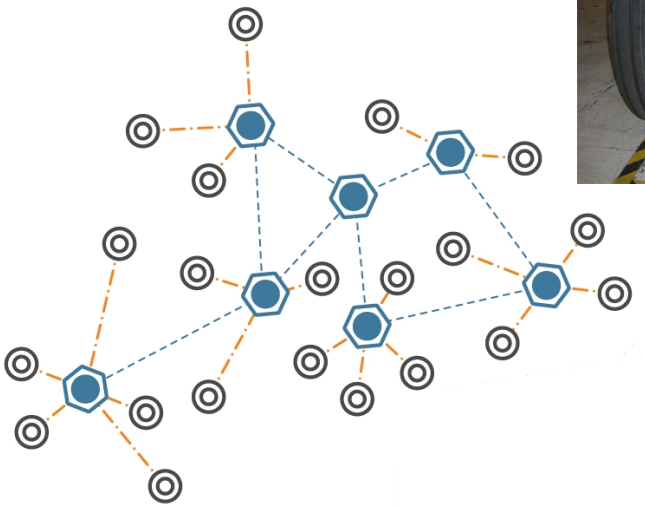
Case examples used in industry

- Aircraft Display Window Manager (3)
- Flight Guidance System (3)

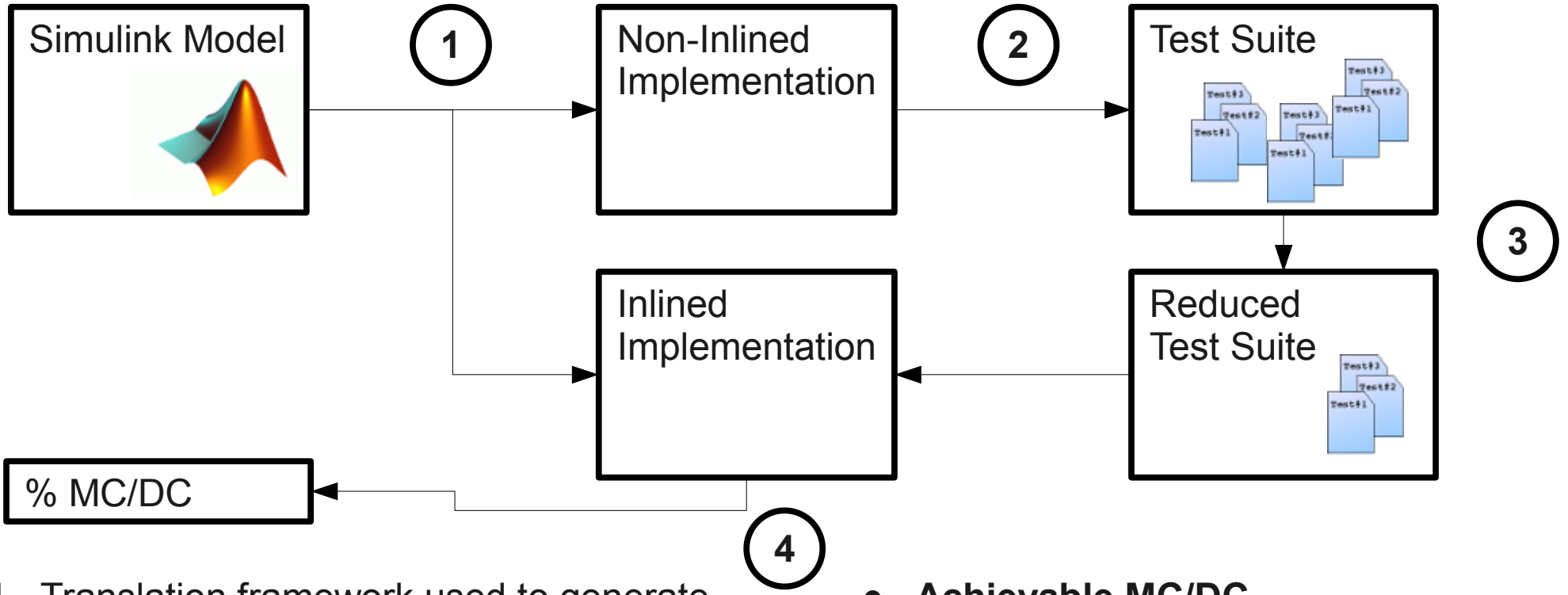
Toy examples

- Wheel Brake System
- Sensor Voting Example

Systems were available as Simulink Models



Experiment Setup



1 Translation framework used to generate different implementations

2 Test suite generated through NuSMV model checker

3 Obtain “minimal” test suite using a (naive) algorithm

4 Compare measured/achievable MC/DC

- **Achievable MC/DC**
  Complete coverage sometimes not possible (e.g. masking)
- **Measured MC/DC**
  Coverage provided by test suite

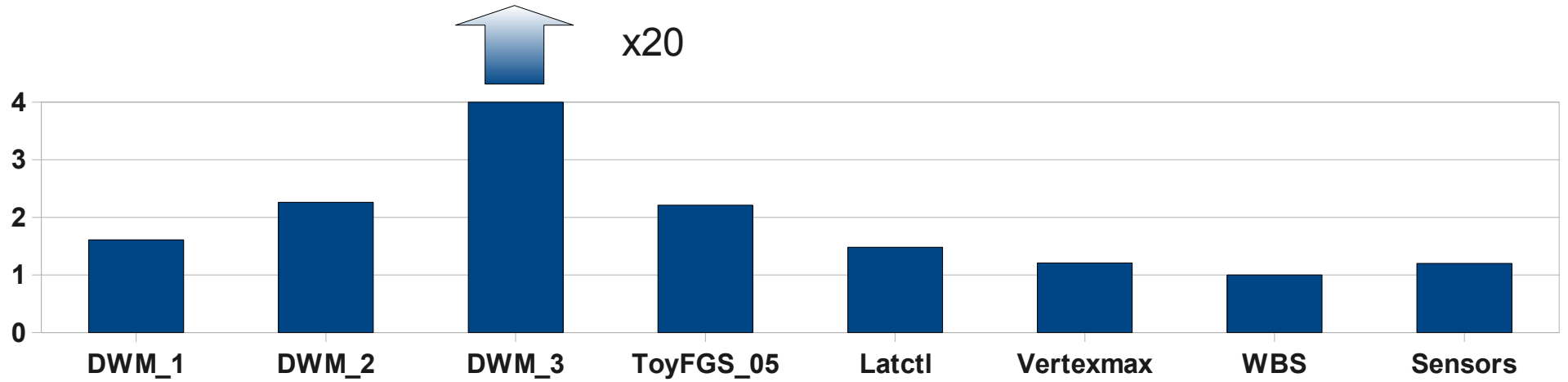
Size of Generated Test Suites

How many tests are needed for each implementation to achieve MC/DC?

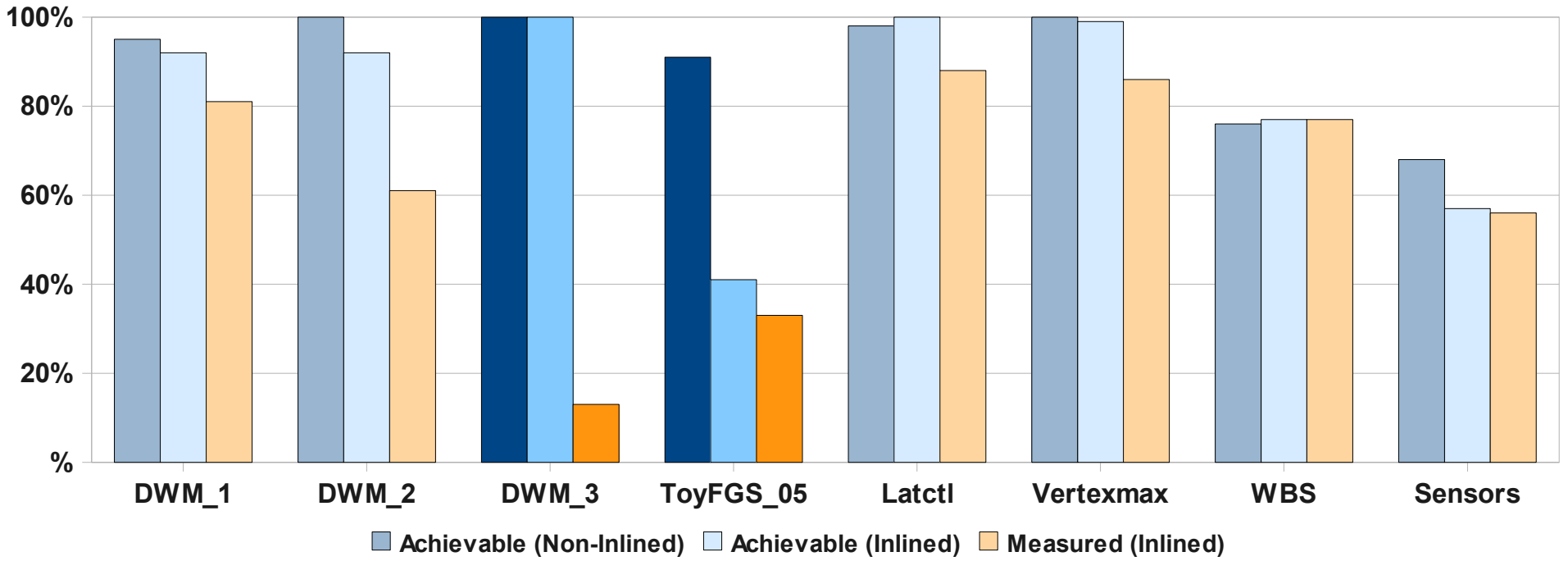
	Non-Inlined		Inlined	
	Full	**Reduced**	**Full**	**Reduced**
DWM_1	180	18	121	29
DWM_2	299	39	946	88
DWM_3	2522	**23**	2697	**463**
ToyFGS_05	4445	75	1909	166
Latctl	315	52	205	77
Vertexmax	1415	235	1464	285
WBS	271	10	125	10
Sensors	103	10	189	12

DWM_3 requires 20 times more tests to cover inlined version!

=> Mostly Boolean logic, leading to complex expressions in the inlined implementation



Achieved Coverage (MC/DC)



- Generated test suite for non-inlined version of DWM_3 achieves very low MC/DC on inlined implementation (13%)
- Interesting: ToyFGS_05 has a much lower achievable MC/DC in inlined version => many DNF expressions containing redundancy causing strong masking effect
- Inadequacy ranging from 13% to 86%, statistically supported on a 5% significance level (including industrial examples only)

Conclusions

MC/DC is indeed highly sensitive to structure of implementation!

- **Suggestions**
  - Different coverage metric that takes masking into account (independent of code structure)
  - Apply coverage on model domain instead of code domain
- **Problems with experiments**
  - Small number of examples
  - Test suite reduction too naive
- **Personally**
  - Removing toy examples from statistics is questionable
  - Effectiveness of MC/DC?