WHY ANALYZE WHEN YOU CAN TEST?

Steven R. Hosner, PE

QinetiQ-NA, INC

steven.hosner@qinetiq-na.com

INTRODUCTION

- There is sometimes confusion about whether or not DO-178B* requires analysis to show achieved software coverage or if it allows, or even makes sense to , force testing to achieve the desired coverage.
- This presentation hopes to add fuel to the fire of discussion and provide rationale for coming down on the side of analysis.

*DO-178B is material copyrighted by RTCA, Inc. and is used with permission in this presentation. My thanks to RTCS, Inc. The complete DO-178B document and other RTCA documents may be purchased from RTCA, Inc. whose contact information is listed below:
RTCA, Inc.
1828 L St., NW
Suite 805
Washington, DC 20036
(202) 833-9339
www.rtca.org

TESTING, TESTING, TESTING

- DO-178B REQUIRES THREE TYPES OF TESTING:
  - HARDWARE/SOFTWARE INTEGRATION TESTING
    - To verify correct operation of the software in the target computer environment
  - SOFTWARE INTEGRATION (HIGH LEVEL) TESTING
    - To verify the interrelationships between the software requirements and components and to verify the implementation of the software requirements and software components within the software architecture
  - LOW-LEVEL LEVEL TESTING
    - To verify the implementation of software low-level requirements

TESTING, TESTING, TESTING (Continued)

- DO-178B SAYS TEST CASES SHOULD:
  - BE BASED ON SOFTWARE HIGH AND LOW LEVEL REQUIREMENTS **(WHICH EVERYONE ALWAYS DOES!!!!!)**
  - VERIFY CORRECT FUNCTIONALITY AND TO ESTABLISH CONDITIONS THAT REVEAL POTENTIAL ERRORS **(WHICH EVERYONE ALWAYS DOES!!!!!)**

TESTING, TESTING, TESTING (Continued)

- EVERYONE DOES REQUIREMENTS BASED TESTING.
- EVERYONE DOES A GREAT JOB ON THAT, DON'T THEY!
- NO???? WELL, OK. MAYBE NOT.
- WHAT ELSE DOES DO-178B REQUIRE?

COVERAGE ANALYSIS, OF COURSE!

- REQUIREMENTS BASED TEST COVERAGE ANALYSIS (PARAGRAPH 6.4.4.1)
  - PURPOSE IS TO DETERMINE HOW WELL THE REQUIREMENTS-BASED TESTING VERIFIED THE IMPLEMENTATION OF THE SOFTWARE REQUIREMENTS
    - Test cases exist for each software requirement
    - Test cases satisfy the criteria of normal and robustness testing
- DOESN'T SOUND TOO DIFFERENT FROM WHAT EVERYONE DOES NOW? WHY DO THIS ANYWAY?

COVERAGE ANALYSIS, OF COURSE! (CONTINUED)

- THE REQUIREMENTS BASED TEST COVERAGE ANALYSIS WOULD HELP CATCH MISSED REQUIREMENTS.
- I CAN SEE WHERE THIS ‘SECOND LOOK’ MIGHT HELP.
- WHAT ELSE?

COVERAGE ANALYSIS, OF COURSE! (CONTINUED)

- STRUCTURAL COVERAGE ANALYSIS (PARAGRAPH 6.4.4.2)
  - PURPOSE IS TO DETERMINE WHICH CODE STRUCTURES WERE NOT EXERCISED BY THE REQUIREMENTS BASED TESTING BY SHOWING TEST COVERAGE ACHIEVEMENT OF:
    - Software structure (data coupling and control coupling) coverage
    - Statement coverage
    - Decision coverage
    - *Modified condition/decision coverage (I am going to chicken out on this one!)
- THIS SOUNDS A BIT COMPLICATED!

*If you are particularly masochistic, see NASA/TM-2001-210876, *A Practical Tutorial on Modified Condition/Decision Coverage*

STRUCTURAL COVERAGE ANALYSIS

- *SOFTWARE STRUCTURE (DATA COUPLING AND CONTROL COUPLING) COVERAGE (YUP, OLD SCHOOL CONCEPTS!)
  - Coupling – a measure of the degree of dependence of one module on another
  - Data coupling – a type of coupling in which the output data of one module is provided as input to another module
  - Control coupling – a type of coupling in which the output of one module influences the execution of other modules

**The Practical Guide To Structured Systems Design*, Meilir Page-Jones
**Structured Design Fundamentals of a Discipline of Computer Program and Systems Design*, Edward Yourdon/Larry Constantine

STRUCTURAL COVERAGE ANALYSIS (CONTINUED)

- STATEMENT COVERAGE:
  - Every line of source code is executed at least once
- DECISION COVERAGE:
  - Every point of entry and exit in the program has been invoked at least once and every decision in the program has taken on all possible outcomes at least once
- MODIFIED CONDITION/DECISION COVERAGE:
  - OOPS! REMEMBER, I CHICKENED OUT OF THIS!

STRUCTURAL COVERAGE ANALYSIS (CONTINUED)

- WAIT A MINUTE! THIS ISN'T JUST A BIT COMPLICATED! IT IS WAAAY TOO COMPLICATED!
- IF YOU DO REQUIREMENTS BASED TESTING, DON'T YOU AUTOMATICALLY GET THE STRUCTURAL COVERAGE?
- EVEN EASIER, WHY NOT JUST FORCE YOUR TESTING TO GET THE STRUCTURAL COVERAGE YOU WANT?

BECAUSE…

- ***COVERAGE ANALYSIS IS A VERIFICATION OF VERIFICATION ACTIVITY (TABLE A-7), NOT A TEST ACTIVITY (TABLE A-6)!***
- ***THAT IS,***
  - ***REQUIREMENTS TEST COVERAGE ANALYSIS ANSWERS THE QUESTION, HAVE WE BUILT THE SOFTWARE RIGHT?***
  - ***STRUCTURAL COVERAGE ANALYSIS VERIFIES THAT THE SOFTWARE STRUCTURE REFLECTS THE REQUIREMENTS***

BECAUSE… (CONTINUED)

- ***THERE ARE INSTANCES WHERE REQUIREMENTS BASED TESTING WILL NOT CATCH REQUIREMENTS-IMPLEMENTATION ERRORS***
- ***FORCED TESTING IS INDEPENDENT OF REQUIREMENTS SO IT, BY DEFINITION, WILL NOT CATCH SUCH ERRORS***

I'M FROM MISSOURI!

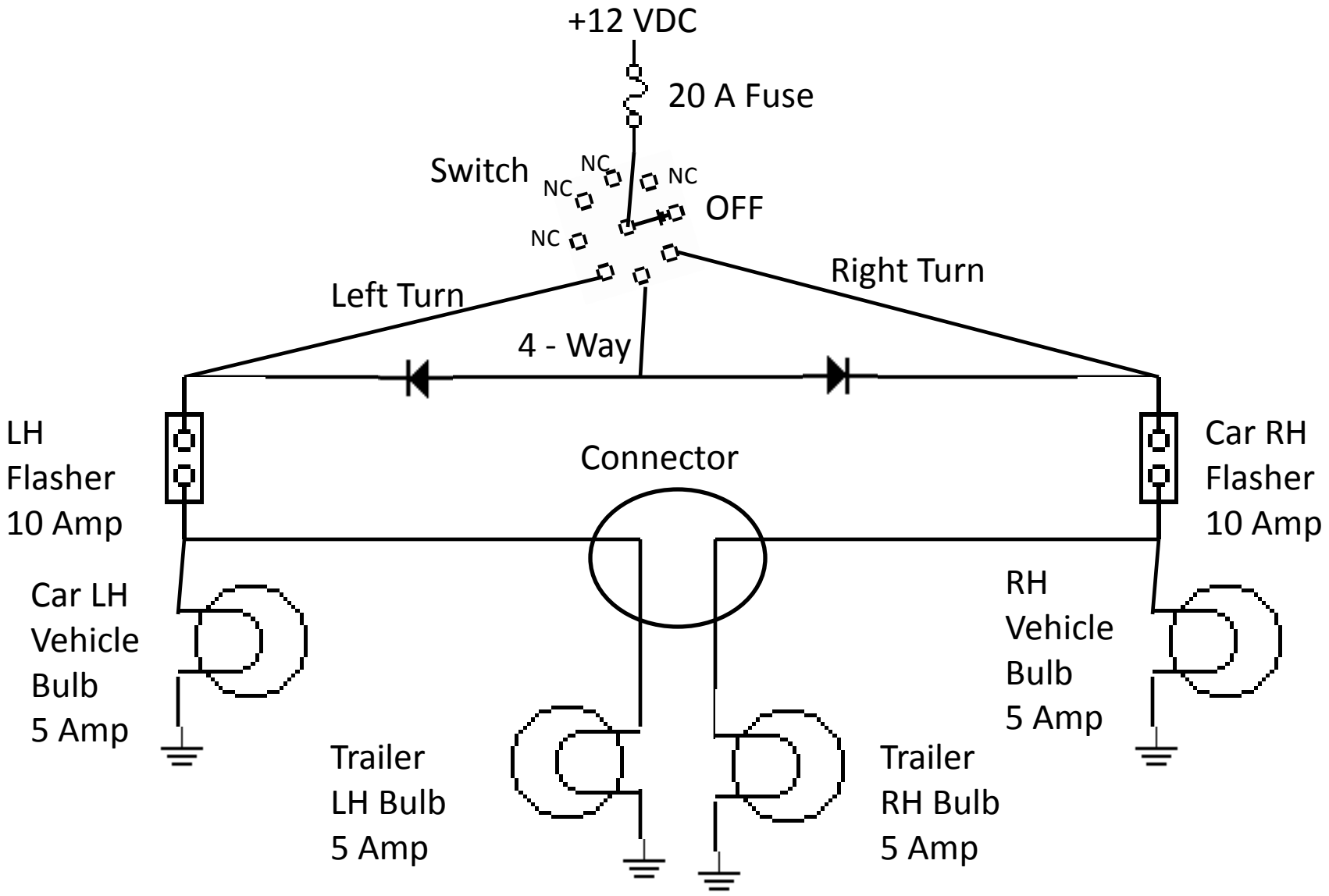
SOME SIMPLE ILLUSTRATIONS

- MODEL A SIMPLE ANALOG SYSTEM
- SPECIFY THE REQUIREMENTS AND SOFTWARE IMPLEMENTATION FOR SOME REPLACEMENT COMPONENTS OF THE SYSTEM

SOME SIMPLE ILLUSTRATIONS (CONTINUED)

- SEE THAT MISMATCHES BETWEEN REQUIREMENTS AND IMPLEMENTATION WILL NORMALLY BE CAUGHT BY TEST COVERAGE ANALYSIS BUT NOT BY FORCED TESTING
- SEE THAT TEST COVERAGE ANALYSIS WILL NOT NECESSARILY ENSURE DATA COUPLING, CONTROL COUPLING OR STATEMENT COVERAGE

A SIMPLE ANALOG SYSTEM



ILLUSTRATIONS OF REQUIREMENTS-IMPLEMENTATION MISMATCHES TEST COVERAGE ANALYSIS WILL CATCH BUT FORCED TESTING WOULD NOT

- RIGHT REQUIREMENT, MISSING IMPLEMENTATION
- NO REQUIREMENT, RIGHT IMPLEMENTATION OTHERWISE
- WRONG REQUIREMENT, WRONG (DIFFERENT) IMPLEMENTATION

SENSE_SWITCH_POSITION1

- RIGHT REQUIREMENT, MISSING IMPLEMENTATION
- REQUIREMENT: SENSE AND RETURN SWITCH PHYSICAL POSITION OF OFF, RIGHT_TURN, LEFT_TURN, 4-WAY AND NC
- IMPLEMENTATION:
  - SWITCH_POSITION IS A MEMORY LOCATION THAT HARDWARE WILL DUMP AN INDICATION OF THE PHYSICAL SWITCH POSITION INTO AT A 500 MS RATE

SENSE_SWITCH_POSITION1 (CONTINUED)

- IMPLEMENTATION (CONTINUED):

```
Function SENSE_SWITCH_POSITION1
Begin
Case SWITCH_POSITION of
  OFF: return OFF;
  LEFT_TURN: return LEFT_TURN;
  OTHER: null;
End SENSE_SWITCH_POSITION;
```

SENSE_SWITCH_POSITION1 (CONTINUED)

- UNIMPLEMENTED REQUIREMENTS – POSITIONS RIGHT_TURN, 4-WAY AND NC NOT RETURNED
  - TEST COVERAGE ANALYSIS WILL CATCH THIS!
  - FORCED TESTING WILL NOT CATCH THIS!

SENSE_SWITCH_POSITION2

- NO REQUIREMENT, RIGHT IMPLEMENTATION OTHERWISE
- REQUIREMENT: SENSE AND RETURN SWITCH PHYSICAL POSITION OF OFF, RIGHT_TURN, LEFT_TURN, 4-WAY AND NC

SENSE_SWITCH_POSITION2 (CONTINUED)

- IMPLEMENTATION:

```
Function SENSE_SWITCH_POSITION2
begin
Case SWTICH_POSITION of
  OFF: return OFF;
  RIGHT_TURN: return RIGHT_TURN;
  LEFT_TURN: return LEFT_TURN;
  4_WAY: return 4_WAY;
  NC: return NULL;
  OTHER: return BACKUP;
End SENSE_SWITCH_POSITION;
```

SENSE_SWITCH_POSITION2 (CONTINUED)

- BACKUP IS NOT A REQUIRED SWITCH POSITION
  - TEST COVERAGE ANALYSIS WILL CATCH THIS!
  - FORCED TESTING WILL NOT CATCH THIS!

FUSE

- WRONG REQUIREMENT, WRONG IMPLEMENTATION
- REQUIREMENT: TURN OFF POWER IF CURRENT FLOW EXCEEDS 15 AMPS
- IMPLEMENTATION:

```
Function FUSE
begin
IF CURRENT_FLOW > 10 AMPS THEN TURN_OFF_POWER
End FUSE;
```

FUSE (CONTINUED)

- ACTUAL REQUIREMENT IS 20 AMPS, SPECIFIED REQUIREMENT IS 15 AMPS AND IMPLEMENTATION IS 10 AMPS
  - TEST COVERAGE ANALYSIS WOULD CATCH THE MISMATCH
  - FORCED TESTING WOULD NOT CATCH THE MISMATCH
  - NEITHER WOULD CATCH THE WRONG REQUIREMENT
  - NOTHING, EXCEPT A GOOD SOFTWARE ENGINEER WOULD CATCH THE INCORRECT REQUIREMENT!

NOT BAD!

- LOOKS LIKE TEST COVERAGE ANALYSIS WILL DO PRETTY WELL FOR SINGLE REQUIREMENTS-IMPLEMENTATION MISMATCHES!
- ISN'T THAT GOOD ENOUGH?
- MAYBE NOT…

LET'S LOOK AT CONTROL COUPLING, STATEMENT COVERAGE AND DECISION COVERAGE PROVIDED BY REQUIREMENTS BASED TESTING!

SENSE_SWITCH_POSITION3

- WRONG REQUIREMENT, CONSISTENT WRONG IMPLEMENTATION
- REQUIREMENT: SENSE AND RETURN SWITCH PHYSICAL POSITION OF OFF AND LEFT_TURN
- IMPLEMENTATION:
  - SWITCH_POSITION IS A MEMORY LOCATION THAT HARDWARE WILL DUMP AN INDICATION OF THE PHYSICAL SWITCH POSITION INTO AT A 500 MS RATE

SENSE_SWITCH_POSITION3 (CONTINUED)

- IMPLEMENTATION (CONTINUED):

```
Function SENSE_SWITCH_POSITION3
Begin
Case SWITCH_POSITION of
  OFF: return OFF;
  LEFT_TURN: return LEFT_TURN;
  OTHER: null;
End SENSE_SWITCH_POSITION;
```

SENSE_SWITCH_POSITION1 (CONTINUED)

- UNIMPLEMENTED ACTUAL REQUIREMENTS – POSITIONS RIGHT_TURN, 4-WAY AND NC NOT RETURNED
  - TEST COVERAGE ANALYSIS WILL CATCH THIS!
  - FORCED TESTING WILL NOT CATCH THIS!

SWITCH

- REQUIREMENTS: IF SWITCH POSITION
  - OFF – TURN OFF POWER
  - RIGHT_TURN – FLASH RIGHT HAND BULBS
  - LEFT_TURN – FLASH LEFT HAND BULBS
  - 4-WAY – FLASH RIGHT AND LEFT HAND BULBS
  - NC – DO NOTHING

SWITCH (CONTINUED)

- IMPLEMENTATION

```
Procedure SWITCH
Begin
Case of SENSE_SWITCH_POSITION3
   OFF: TURN_OFF_POWER;
   RIGHT_TURN: FLASH_RIGHT_HAND_BULBS;
   LEFT_TURN: FLASH_LEFT_HAND_BULBS;
   4-WAY: FLASH_RIGHT_HAND_BULBS;
              FLASH_LEFT_HAND_BULBS;
   NC: NULL;
   OTHER: NULL;
   End case SENSE_SWITCH_POSITION;
End SWITCH;
```

SWITCH (CONTINUED)

- TEST COVERAGE ANALYSIS WOULD PASS BOTH PIECES OF SOFTWARE
- DECISION COVERAGE ANALYSIS NOT SATISFIED:
  - SENSE_SWITCH_POSITION3 WILL NOT RETURN RIGHT_TURN, 4-WAY AND NC
  - RIGHT_TURN, 4-WAY AND NC BRANCHES NOT TAKEN

SWITCH (CONTINUED)

- STATEMENT COVERAGE ANALYSIS NOT SATISFIED:
  - STATEMENTS ASSOCIATED WITH RIGHT_TURN, 4-WAY AND NC NOT EXECUTED
- CONTROL COUPLING ANALYSIS NOT SATISFIED:
  - SWITCH 'TURNING ON' OR CONTROLLING FLASH_RIGHT_HAND_BULBS WILL NOT OCCUR

THE GRAND FINALE!

- TEST COVERAGE ANALYSIS:
  - CAN BE EFFECTIVE IN SHOWING THAT REQUIREMENTS HAVE BEEN EXERCISED FOR AN INDIVIDUAL SOFTWARE MODULE
  - IT WILL NOT CATCH MODULE INTERACTION PROBLEMS

THE GRAND FINALE! (CONTINUED)

- COVERAGE ANALYSIS:
  - IS A VERIFICATION ACTIVITY, NOT A TEST ACTIVITY
  - CAN MAKE REQUIREMENTS BASED TESTING MORE EFFECTIVE BY CATCHING CERTAIN TYPES OF REQUIREMENTS-IMPLEMENTATION MISMATCHES
  - CAN CATCH SOFTWARE ARCHITECTURE (INTER-MODULE COMMUNICATION) PROBLEMS

THE GRAND FINALE! (CONTINUED)

- FORCED TESTING
  - IS ENTIRELY A TESTING ACTIVITY
  - WILL NOT CATCH REQUIREMENTS-IMPLEMENTATION MISMATCHES
  - WILL NOT CATCH SOFTWARE ARCHITECTURE (INTER-MODULE COMMUNICATION) PROBLEMS

THE GRAND FINALE! (CONTINUED)

QUESTIONS?