

# **“V&V of COTS Dormant Code: Challenges and Issues”**

**GSAW99**

**Los Angeles, CA**

**Ronald J. Kohl**

**AverStar, Inc.**

**kohl@averstar.com**



# Background

- **Ideally, there is always a 1-1 mapping between requirements and operational software**
- **With increased use of COTS, this is not always the case**
  - some requirements are not met w/COTS (custom/glue code needed)
  - some COTS features are not required (Dormant Code exists)



# How did we get here?

- **Government directive to use COTS**
  - reduced cost, reduce single provider dependency, more open systems
- **Supplier desire to be more competitive**
  - reduce costs, win bids
- **COTS are ‘market driven’**
  - popular features, huge follow-on business, billions of users
- **System Requirements are ‘domain specific’**
  - specialized needs, limited follow-on business, tens or hundreds of users



# Mathematical Description

- **Given a set of requirements, R**
- **Given a set of COTS capabilities, C**
- **Given a Traceability mapping, Tr, from R to C, define the Dormant Code (DC) of C to be that subset of C that has no inverse image, under Tr, in R.**
- **That is:  $DC = C - Tr(R)$**



# What are the DC issues?

- **DC identification and analysis**
- **Impacts of activation & contingency plans**
- **Disabling/prevention**
- **Contractual impacts**
- **Cost impacts**



# What are some approaches?

- **Finding DC (vendor, peers, user's groups)**
- **Determine activation stimuli (events, msgs, etc)**
- **Disabling/prevention (filters, firewall, procedures)**
- **Contractual impacts (rqmts changes, contract changes, stakeholder involvement)**
- **Cost impacts**



# Challenges of V&V of DC

- **Whereas most COTS can be V&V'd via 'black box' techniques, for DC you may need to develop/use 'puzzle box' techniques**
  - I wonder what it really does?
  - I wonder how to make it do something weird?
  - I wonder if it could really do ...?
  - I can't believe it would really do ...!
- **Performance and capacity issues**
  - How fast should it really work?
  - How big should it really be?



## How to attack these challenges?

- **Use good black box techniques**
- **Don't abandon V&V techniques used for custom built items**
  - establish good vendor relationships
  - ask the vendor for access to: requirements, source code, processes
  - negotiate with vendor in order to satisfy your needs
- **Use system-wide perspectives**
  - operational scenarios
  - system performance/capacity testing and monitoring
- **Learn all you can about the COTS product!**



## Further Areas of Research

- **Just how prevalent is this problem?**
- **Expand current approaches**
  - improve detection/analysis, (i)v&v methods
- **Identify new concerns and approaches**
  - apply to other NDI
  - cost models
  - new detection schema



## Summary

- **COTS Dormant Code poses potential problems (as does other NDI code)**
- **Education of customer, user, developer, vendor is needed**
- **Identification and mitigation techniques should be considered**
- **Balance cost of problem with cost of fix**

