



# V&V for Adaptive Systems

Bojan Cukic

Lane Department of CSEE  
West Virginia University

**IFIP WG 10.4**  
January 2009



# Adaptive Systems V&V Framework

- Providing evidence that a set of stated (or emerged) properties are satisfied during system operation.

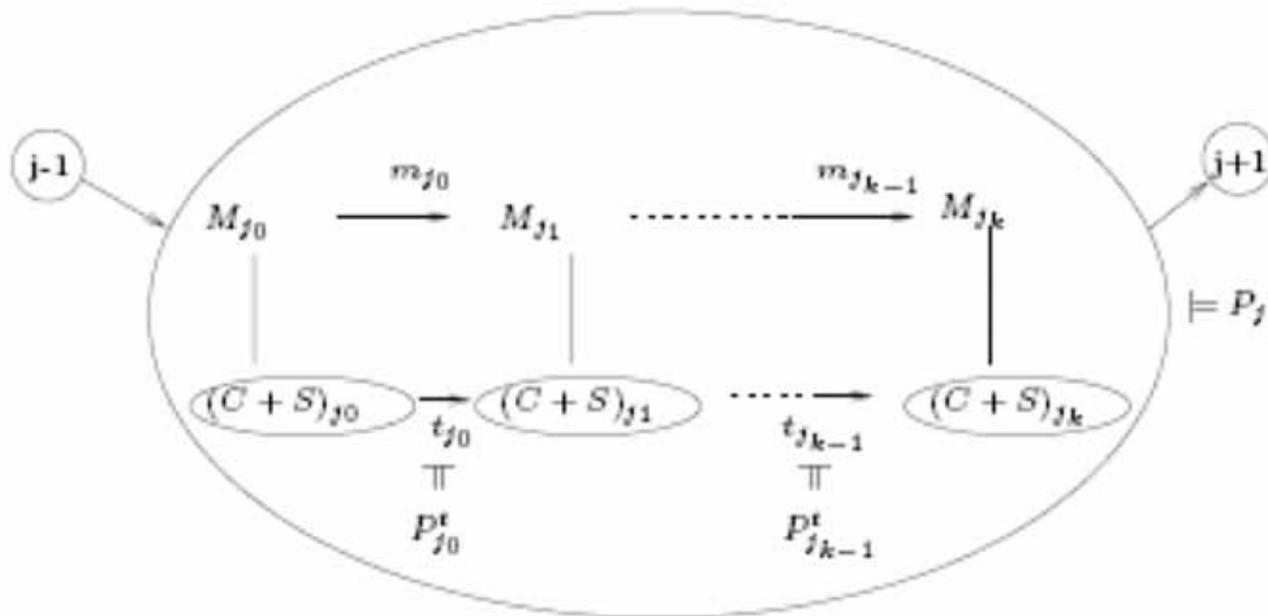


Figure 2: V & V model.

From “Software Engineering for Self Adaptive Systems, A Roadmap”, May 2008, Dagstuhl seminar 08031.



# (Non)traditional V&V

- **Systems are highly context dependent.**
  - Changing requirements (goals), context, functionality.
- **Development life cycle vs. computational (execution) life cycle.**
  - Detection and identification of changes
    - Emerging goals, properties
    - Context monitoring
    - Failure detection and identification
  - Sequences of computational adjustments
    - Model development
    - Monitoring
    - “Embedded” verification (in the computational lifecycle)



# Challenges (1)

- **Detection and identification of changing requirements and contexts**
  - Requirements/context change is either explicit or implicit.
  - Reliable detection/identification is a prerequisite for accommodation.
  - Promising ideas:
    - Failure detection/identification in distributed systems (failure classes, failure assumptions...)
    - Feedback loops, Reflection [Dawson et. al.].



## Challenges (2)

- **Adaptation specific model-driven environments**
  - The importance of models will only increase with adaptation!
  - Development time verification will need to be *supplemented* by run time model analysis.
    - Model based adaptation is a form of run-time verification.
  - But *beware* of model inaccuracy (uncertainty), synchronization problems, complexity of analysis.
  - Promising ideas:
    - Mechatronic UML & compositional pattern verification [Giese et. al.]
    - Checking of emerging models [Cheng et. al.]
    - Adaptive online program analysis [Dwyer & Elbaum]



## Challenges (3)

- **Agile run time assurance**
  - Necessary when accurate model updates may not be feasible.
  - The last line of defense for unanticipated changes.
  - Efficient, incomplete verification strategies with low space/time complexity.
  - What makes an adaptation “correct” or “desirable”?
    - *“Investigation of scientific principles needed to move software assurance beyond current conceptions and calculations of correctness” [NSF 07]*
  - Promising ideas:
    - Proof carrying codes (PCC) [Lee et. al.]
    - Self stabilizing systems
    - Convergence and stability monitoring, prediction [Cukic et. al.]



1536

# Monitors

Y. Liu et al. / The Journal of Systems and Software 79 (2006) 1527–1540

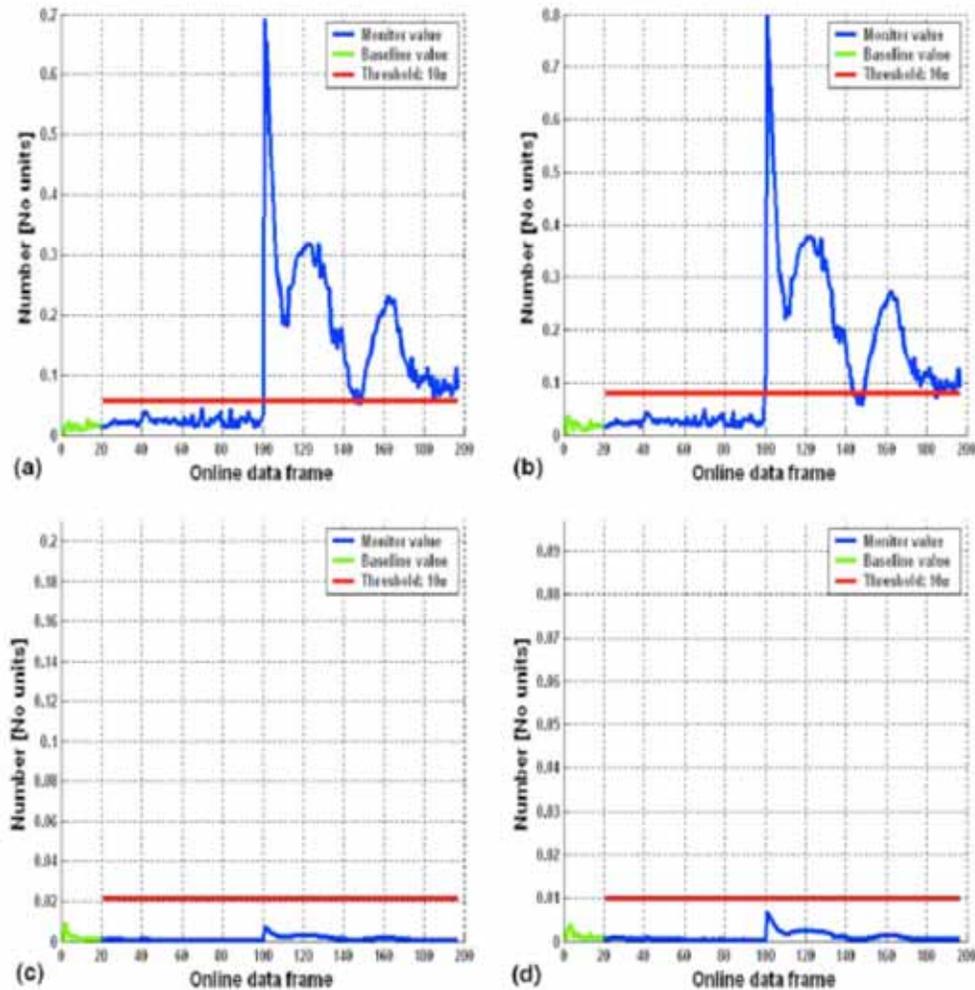


Fig. 8. Online Stability Monitoring System for a 50% loss in functionality of a control surface. (a) Monitor #1, BMU Error; (b) Monitor #2, SBU Error; (c) Monitor #3, NBR Error; (d) Monitor #4, Non-NBR Error.



## Challenges (4)

- **How would a developer organization argue they applied “expected care” in the assurance of adaptive applications?**
  - Context changes may be unexpected.
  - Unforeseeable states.
- **Risk aware adaptation?**
  - Can we argue that adaptation mitigates inherent application risks?
  - Could adaptation be used as a defense argument in liability claims?