Software Product Line Testing
Part V : SPL-Driven Test Processes

Myra Cohen Matthew Dwyer

Laboratory for Empirically-based Software Quality Research
Department of Computer Science
University of Nebraska - Lincoln

Work supported by NSF CCF through awards 0429149 and 0444167, by the U.S. Army Research Office through award DAAD190110564 and by an NSF EPSCoR FIRST award.

e² ESQuaReD

University of Nebraska Lincoln

Outline

Software Product Lines : What and Why?

Modeling Variability in Software Product Lines

Validating Product Lines

A Framework for Variability Coverage

- Toward Product Line Driven Test Processes

Outline

- Toward Product Line Driven Test Processes

1. Exploiting SPL Lifetime
2. Variable Interaction Coverage Criteria
3. Challenges and Open Issues

The Meaning of Validation

A program is validated if we have confidence that it will operate correctly

A software product line is validated if we have confidence that any instance of that produce line will operate correctly

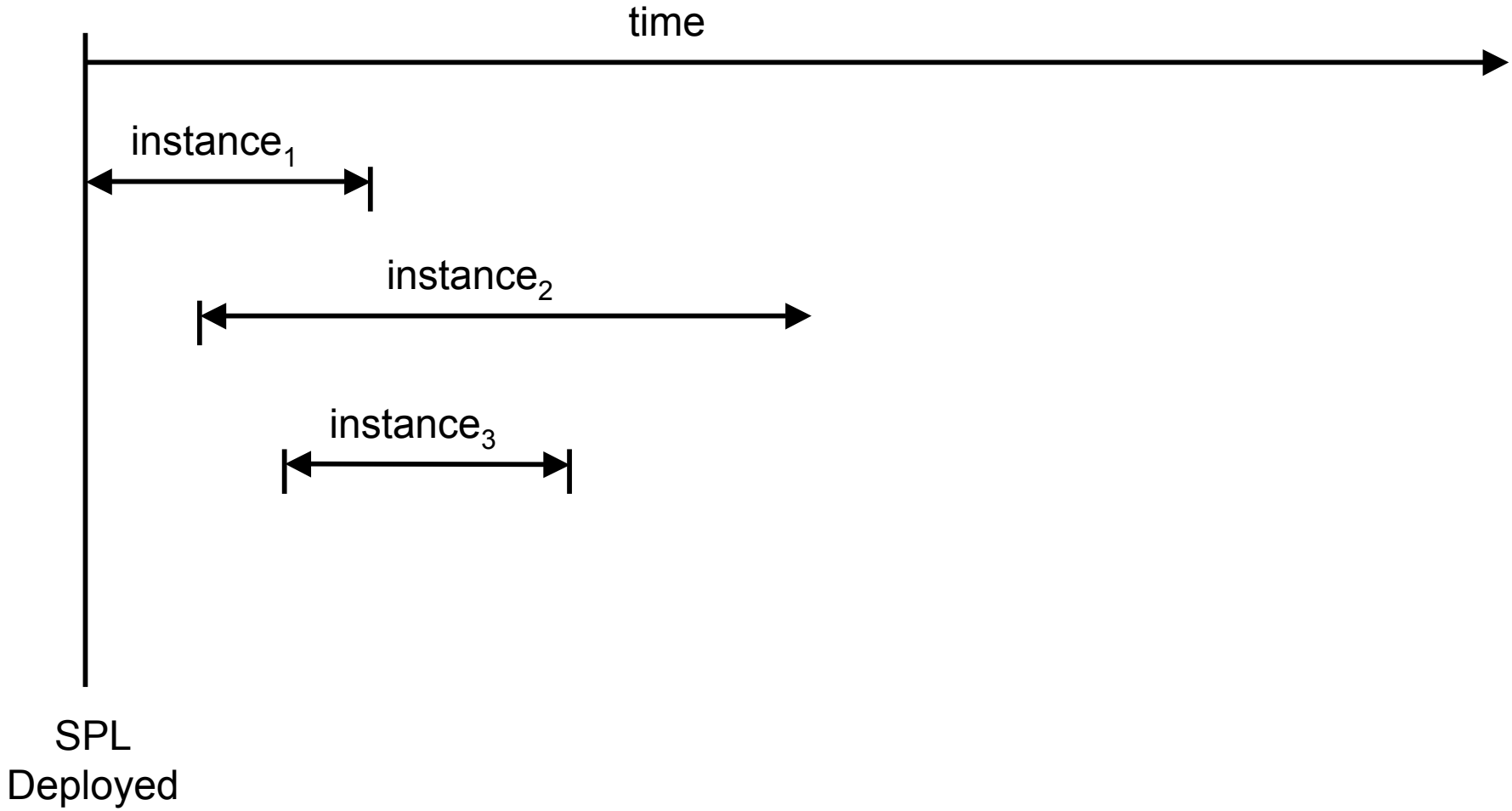
A Two Way Street

As we validate programs, that are instances of an SPL, we gain confidence in the validity of the SPL itself

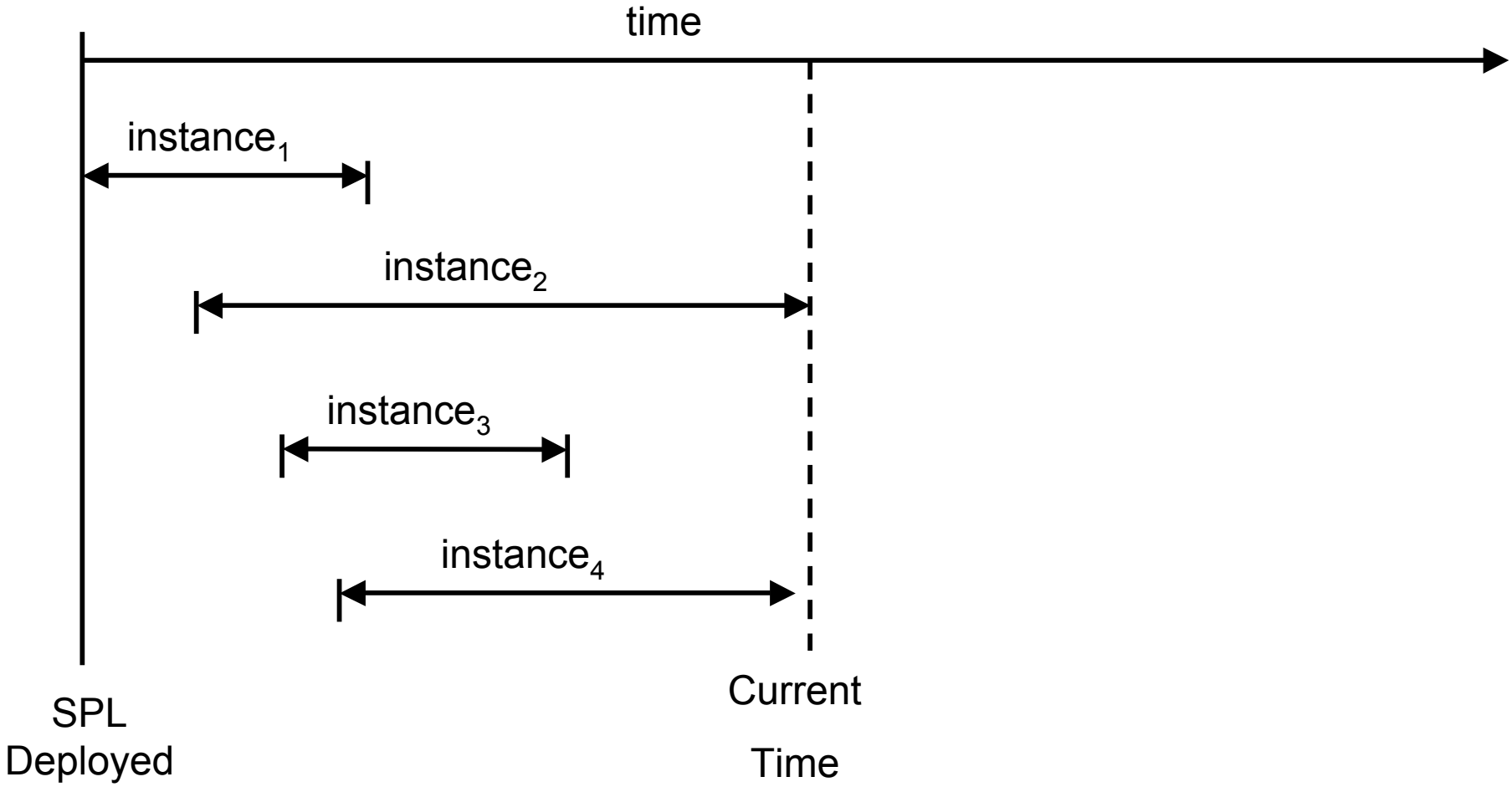
As we gain confidence in an SPL, our baseline belief in the validity of SPL instances is increased.

$cov(instance_1)$

$cov(instance_1)+cov(instance_2)$

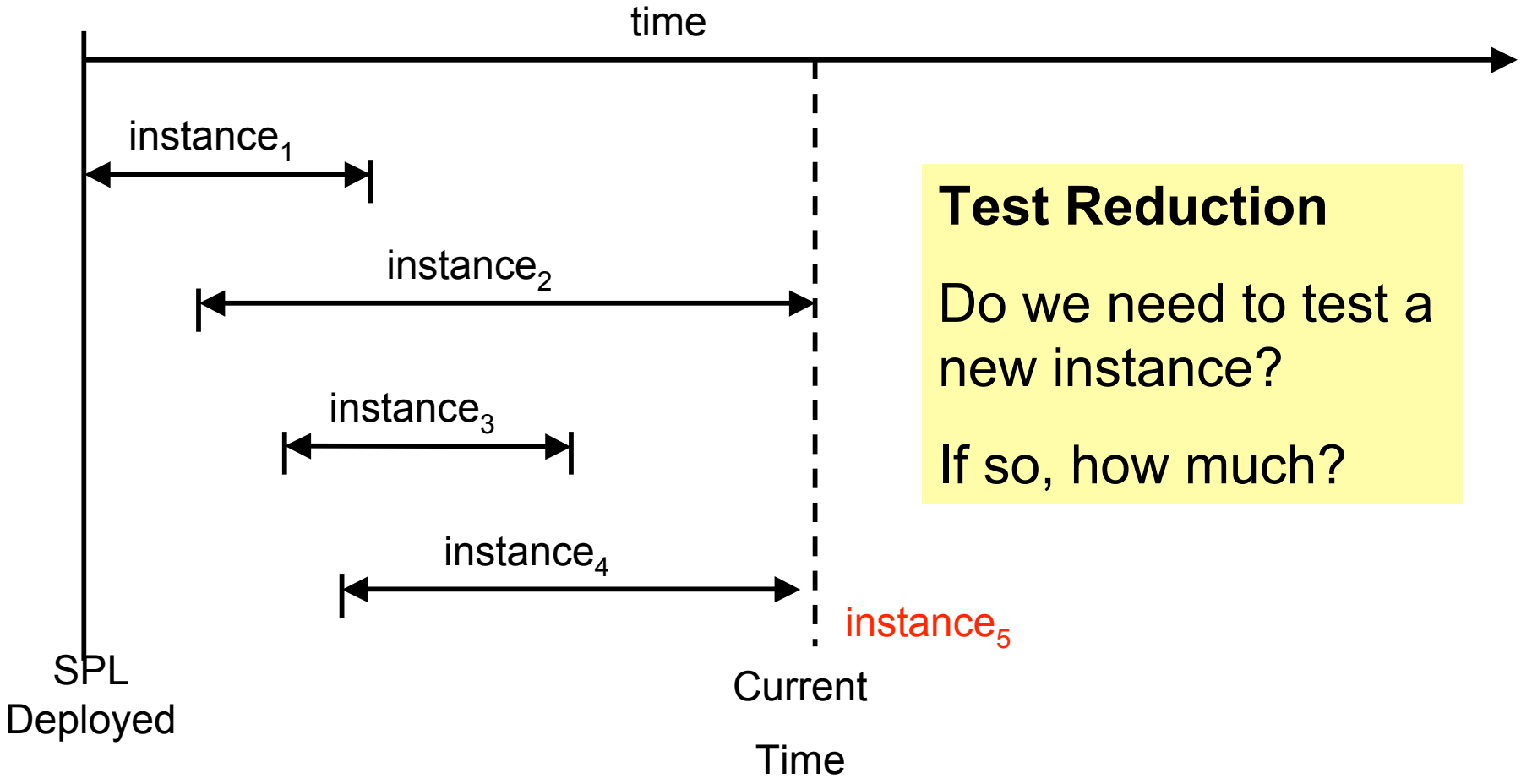


$$cov(instance_1)+cov(instance_2)+ cov(instance_3)$$



$$cov(t) = \Sigma_{j<t}\ cov(instance_j)$$

Cumulative Test Coverage

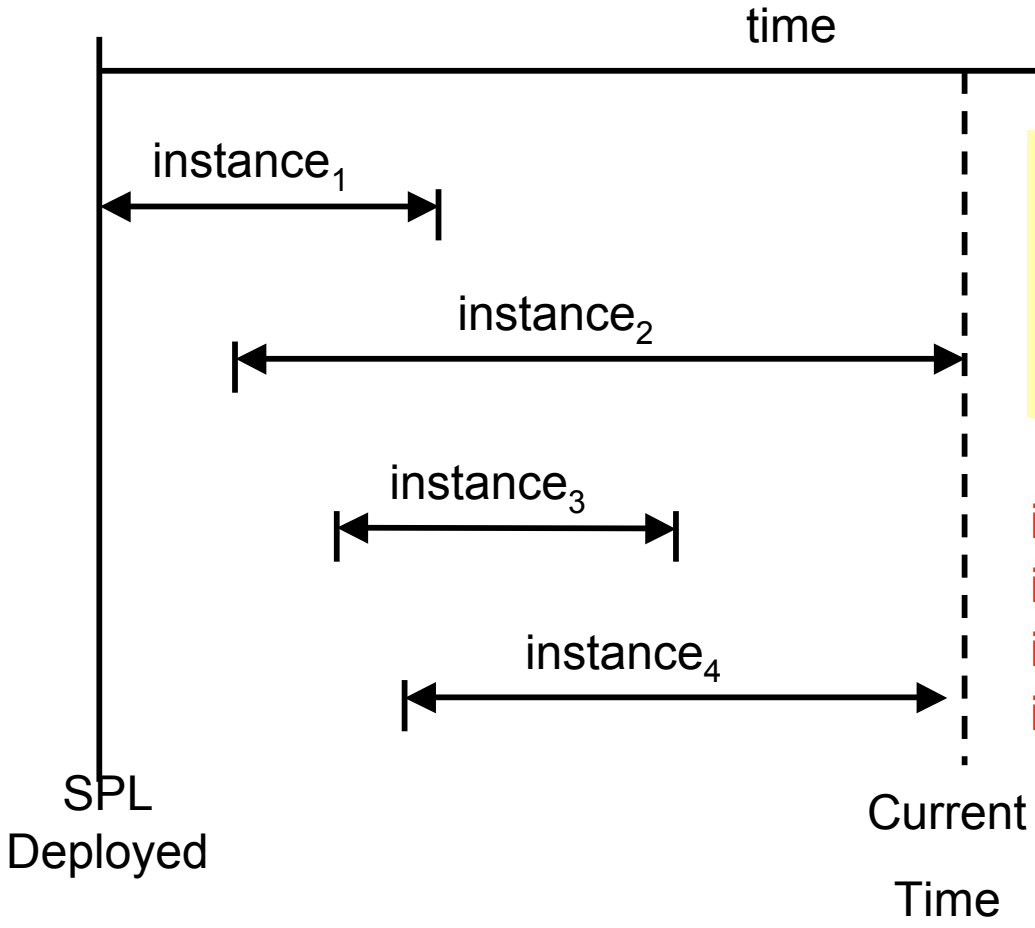


Test Reduction

Do we need to test a new instance?

If so, how much?

$$cov(t) = \Sigma_{j<t}\ cov(instance_j)$$



Test Generation

What instances will significantly increase cov(t)?

$instance_a$
$instance_b$
$instance_c$
$instance_d$

$$cov(t) = \Sigma_{j<t}\ cov(instance_j) + \Sigma_{j\ in\ \{a,b,c,d\}}\ cov(instance_j)$$

SPL Test Coverage

Variability is key

- Faulty interactions among sets of variants are a concern

Evaluate the extent to which sets of variants have been validated

Variability interaction coverage

- Apply interaction coverage to relational model of variability in an SPL

Variability Interaction Coverage

Consider a SPL with k variability-related factors

Recall that t-way coverage means

for all t-sized subsets of factors, $0 \leq t < k$,

all combinations of values those factors appears

2-way, or pair-wise, coverage means

all pairs of variant to VP bindings in an SPL are covered

Variability Interaction Coverage

We can define a family of coverage criteria for variability models based on coverage strength

Criteria are ordered by strength

t-way coverage subsumes (t-1)-way coverage

Variable-strength criteria : vs(min,max)

vs(min,max) subsumes min-way coverage

max-way coverage subsumes vs(min,max)

Important caveat …

We are not talking about how to test a SPL instance

- We assume that existing methods for program testing can be applied
- Clearly our inferences about SPL validation are dependent on the fault revealing power of the underlying program testing method

Challenges

Scaling Interaction Testing

- Only applied to simple models to date
- Sampling with CA's will reduce the test space and provide low-cost test adequacy criteria

But:

- Real SPLs may have hundred of VPs and several hundreds of variants
- Will the complexity introduced by real software product lines scale?

Challenges

Treating Rich Constraints

- As constraints grow in complexity and number, the difficulty of modeling and generating CA test suites increases

- Can emerging techniques for encoding and analyzing collections of constraints, e.g., SAT, BDDs, be integrated with CA techniques?

Challenges

SPL-driven Instance Testing

- An instance of a software product line is a program, but it's not an arbitrary program

- Can we exploit the SPL model to generate tests that are effective at revealing faults?
- Can such methods be made sensitive to cumulative coverage information?

Challenges

Empirical Evidence

– Empirical evidence for interaction testing is derived from non-SPL software testing

But:

– Variability in an SPL may differ significantly from configurable programs

– We need empirical evidence on product lines to:

- Understand sizes (number of variability points and variants)
- Quantify extent and complexity of constraints
- Effectiveness and feasibility of combinatorial testing methods

References

Richardson, D.J., and Wolf, A.L., Software Testing at the Architectural Level, Second Workshop on Software Architecture (1996)

McGregor, J.D., Testing a Software Product Line, Technical Report CMU/SEI-2001-TR-022 (2001)

Muccini, H., and van der Hoek, A., Towards Testing Product Line Architectures, ENTCS 82, No. 2 (2003)

Kauppinen, R., Testing framework-based Software Product Lines, M.S. Thesis, University of Helsinki (2003)

Bertolino, A., and Gnesi, S., PLUTO: A Test Methodology for Product Families, 5th International Workshop on Software Product-Family Engineering, LNCS 3014 (2004)

Kamsties, E., Pohl, K., Reis, S., and Reuys, A., Testing Variabilities in Use Case Models, 5th International Workshop on Software Product-Family Engineering, LNCS 3014 (2004)

Cohen, M.B., Dwyer, M.B., and Shi, J., Coverage and Adequacy in Software Product Line Testing, Technical Report, University of Nebraska - Lincoln (2006)