

# Contact Profile (1/2)

- **Cheng Thao & Ethan Munson**
  - Department of EE & CS
  - University of Wisconsin-Milwaukee
  
- **Focus & interests**
  - Version control for SPL

**Cheng Thao & Ethan Munson, UW-Milwaukee**

# Contact Profile (2)

- **Objectives / Looking for**
  - Looking for at the workshop?
    - Version control for SPL
    - SPL evolution in industry
    - Key SPL evolution research challenges
  - People that you would like to meet?
    - Industrial participants who know how SPL evolution really occurs
    - Researchers in SPL evolution
  - PLEASE 2011 “success criteria”?
    - Meeting people described above

**Cheng Thao & Ethan Munson, UW-Milwaukee**

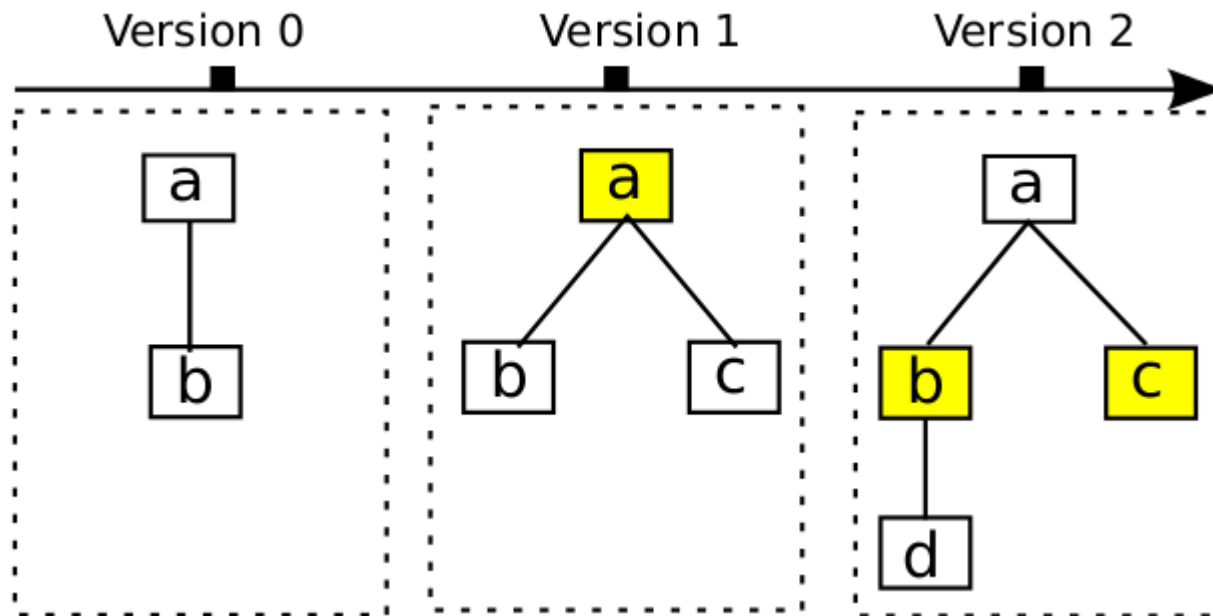
# The Problem

- Core assets shared among multiple products.
- Core assets and products evolve independently.
- Version support for independent development of products and core assets
  - Capture evolution of products and core assets
  - Propagate changes in both directions
  - Without exposing versioning complexity needlessly

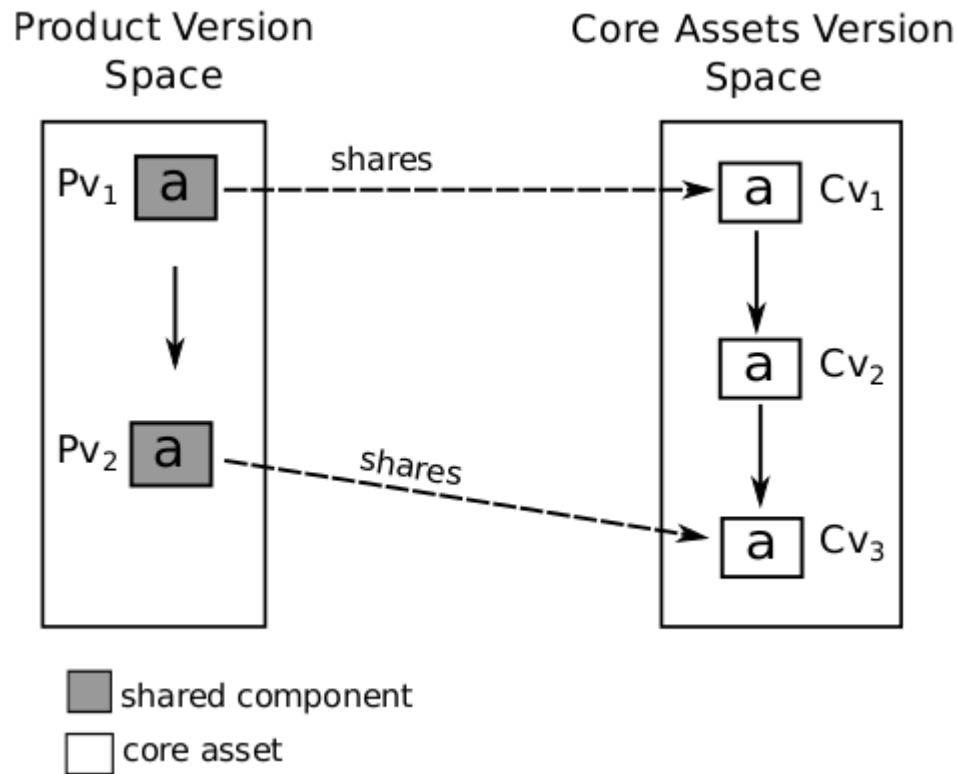
# Current Solution

- Version system with a SPL model:
  - a single core assets project,
  - multiple product projects,
  - sharing of artifacts between products and core,
  - product derivation support
- Supports change propagation in both directions.
- Supports independent development of products and core assets.

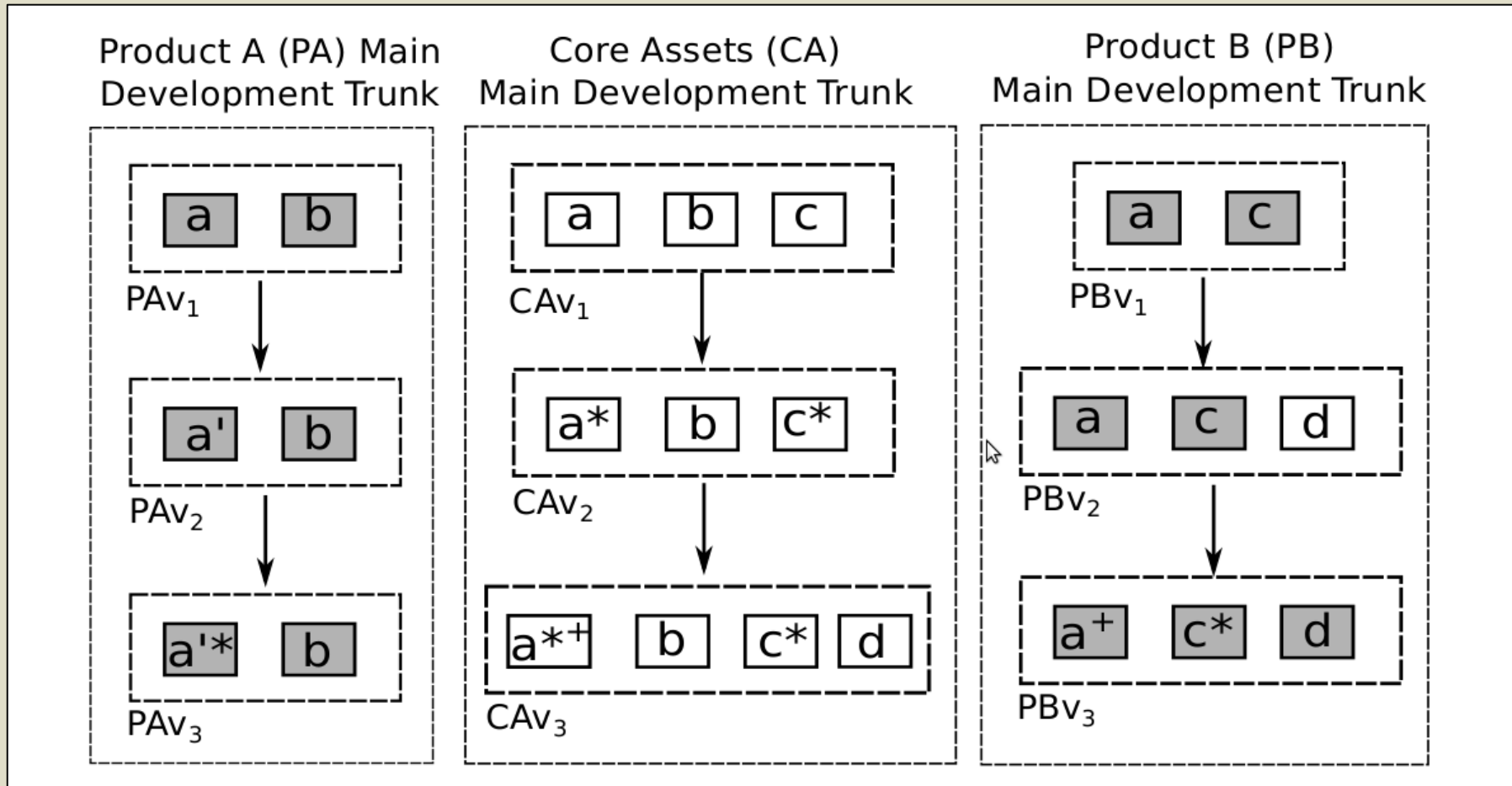
# Product Versioning Model



# Shared Component

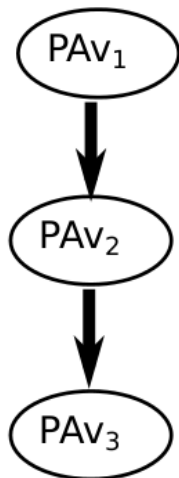


# SPL Evolution

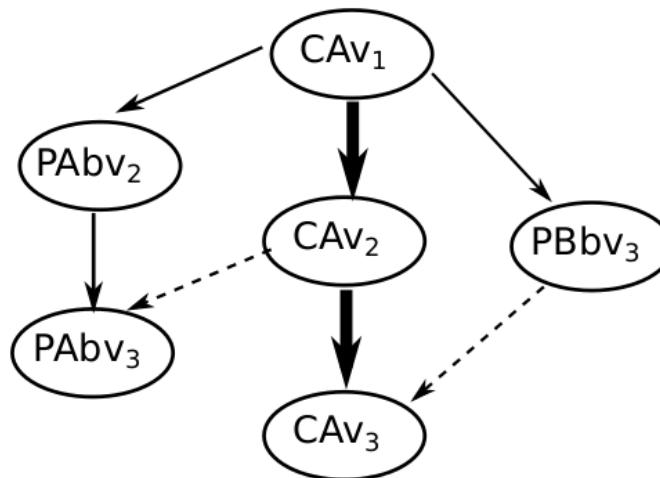


# Product Specific Support

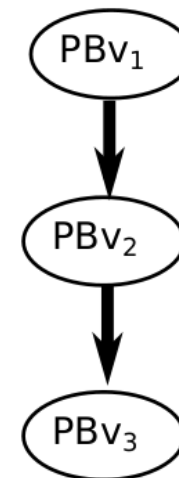
Product A Version Tree



Core Assets Version Tree

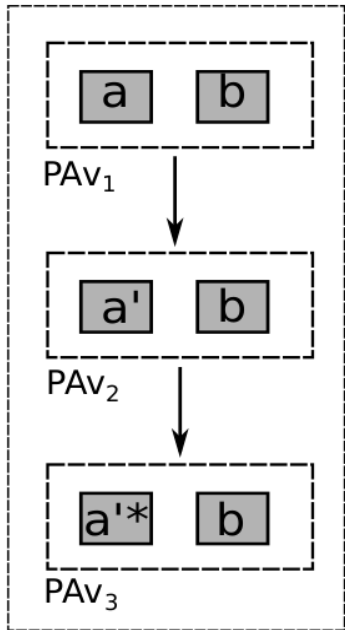


Product B Version Tree

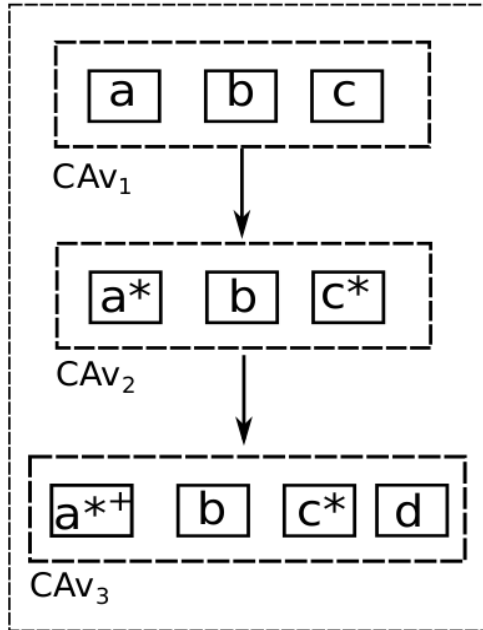




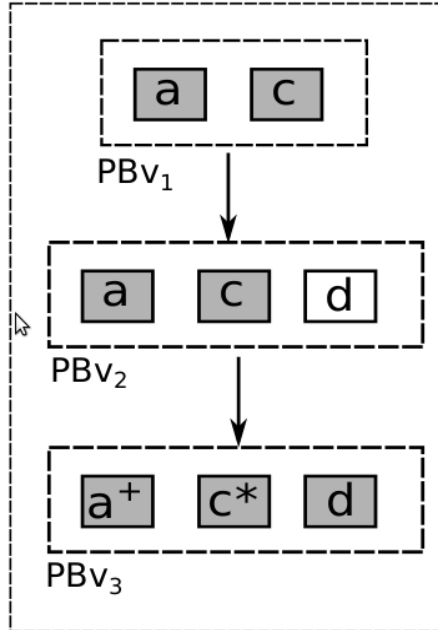
Product A (PA) Main Development Trunk



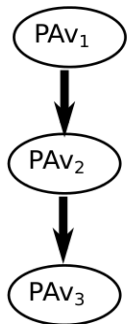
Core Assets (CA) Main Development Trunk



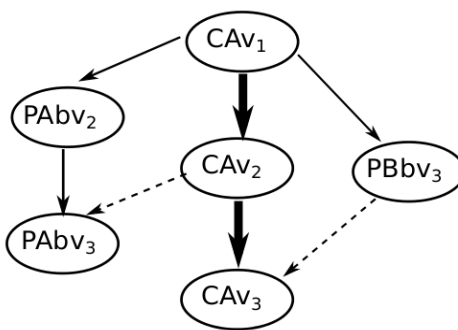
Product B (PB) Main Development Trunk



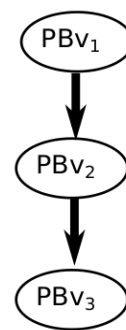
Product A Version Tree



Core Assets Version Tree



Product B Version Tree



# Change Propagation Cases

Case	Before		After	
	Core	Product	Core	Product
1	$a'$	$a$	$a'$	$a'$
2	$a'$	$a^*$	$a'$	$a^{*}$
3	$a'$	$a^*$	$a'$	$a'$
4	$a$		$a$	$a$
5	$a$	$a^*$	$a^*$	$a^*$
6	$a'$	$a^*$	$a'^*$	$a^*$
7	$a'$	$a^*$	$a^*$	$a^*$
8		$a$	$a$	$a$

# Why is it interesting?

- Traditional versioning system:
  - Support independent development of product and core assets but doesn't support variants well.
  - Can't distinguish core assets and products
  - No support for asset sharing and change propagation
  - Makes SPL evolution difficult to support
- Our system provides this

# Potential Collaborations

- A realistic example to test our prototype
- Wisdom about code sharing across product lines
- Version system could help automate product derivation based on a feature model
- Help us understand variability models to determine if we can support them.