



# Using Service Utilization Metrics to Assess and Improve Product Line Architectures

---

Ebru Dincel

Center of Software Engineering

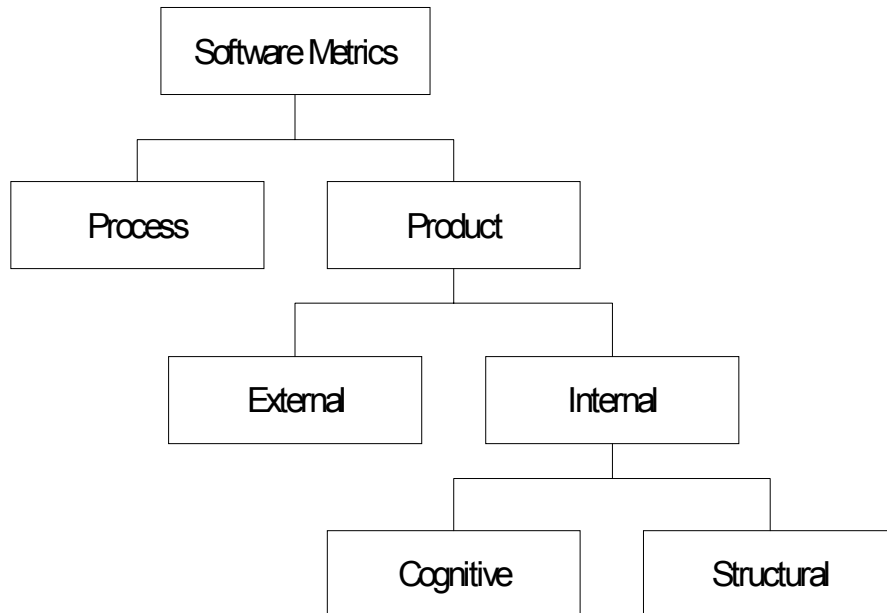
University of Southern California

(in collaboration with Andre van der Hoek and Nenad Medvidovic)



# Software Measurement

---



## Assessment Techniques

- Subjective

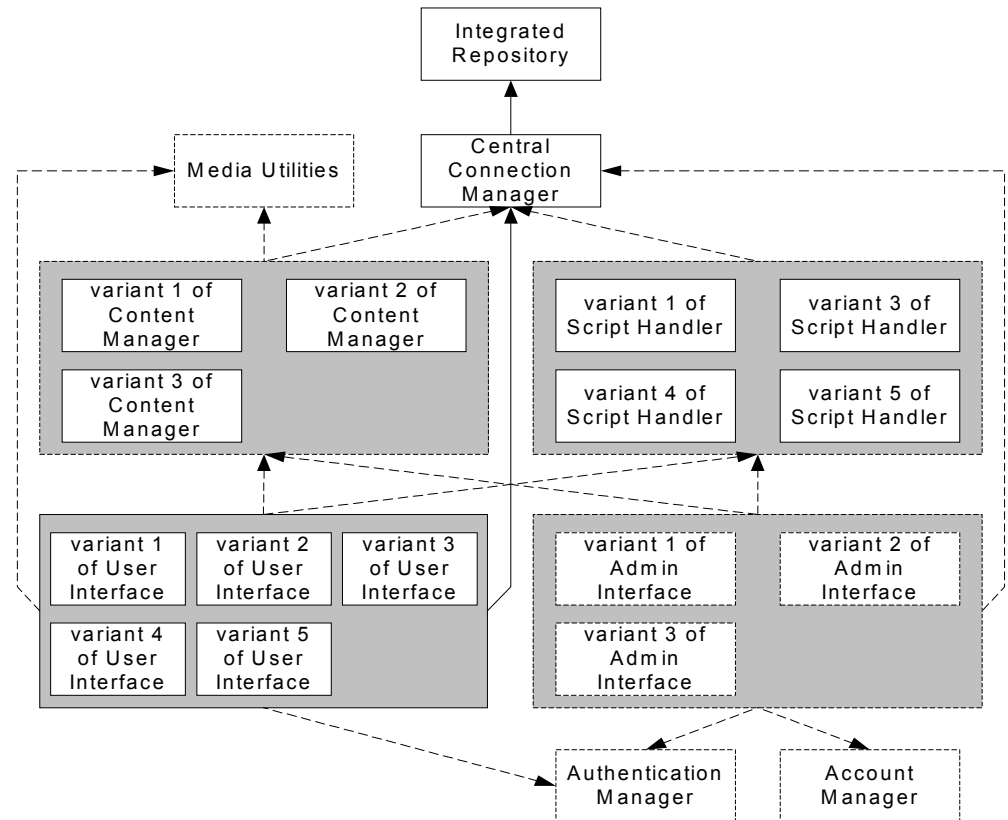
- Questionnaire based
- Scenario based
- Experience based

- Objective

- Modeling
- Simulation

# Product Line Architectures

- Shared core
- Optionality
- Variability
- Optional variability
- Hierarchy





# Service Utilization Metrics

---

- Structural quality assessment for a PLA
  - Absolute values for problem detection
  - Relative values for alternative assessment
- Based on service utilization of provided and required services
- Name/interface/behavior/protocol matching



# Basic Building Blocks: PSU & RSU

---

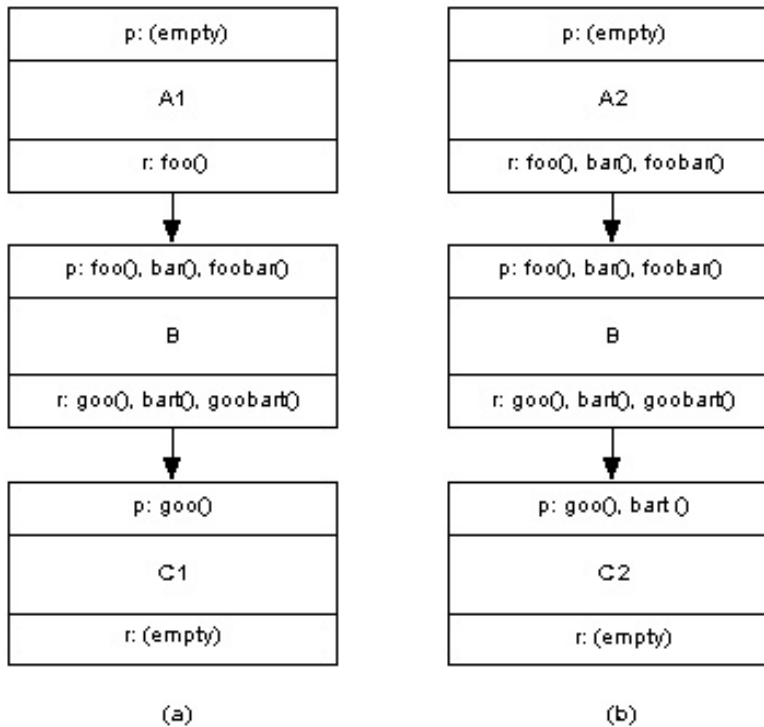
- PSU (Provided Service Utilization)

$$\text{PSU}_x = \frac{P_{actual}}{P_{total}}$$

- RSU (Required Service Utilization)

$$\text{RSU}_x = \frac{R_{actual}}{R_{total}}$$

# Example



- $PSU_B = 1/3, RSU_B = 1/3$
  - $PSU_B = 1, RSU_B = 2/3$
- => Component b is a better fit in architecture (b)



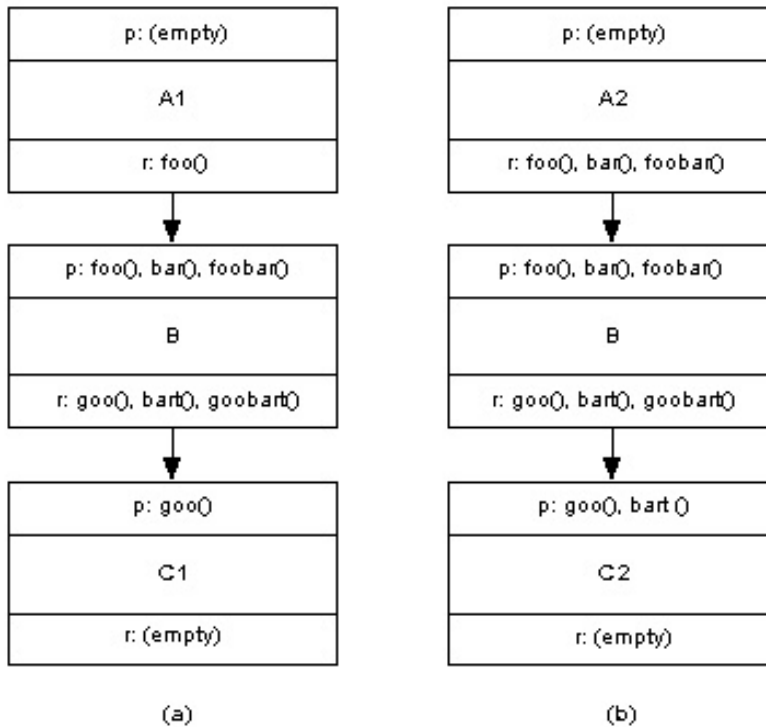
# Compound Building Blocks : CPSU & CRSU

---

$$\text{CPSU} = \frac{\sum_{i=1}^n P_{actual}^i}{\sum_{i=1}^n P_{total}^i}$$

$$\text{CRSU} = \frac{\sum_{i=1}^n R_{actual}^i}{\sum_{i=1}^n R_{total}^i}$$

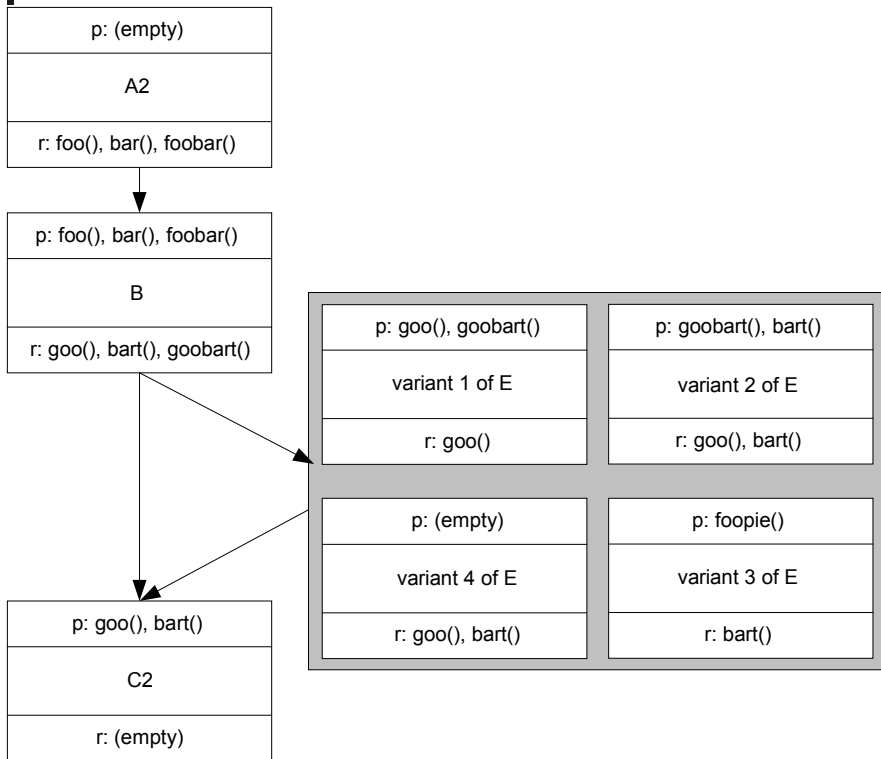
# Example



- $CPSU = 1/2$  ,  $CRSU = 1/2$
- $CPSU = 1$  ,  $CRSU = 5/6$

CPSU	CRSU	indicates
↑	↑	Cohesion
↑	↓	Imbalance, self insufficiency
↓	↑	Imbalance, redundancy
↓	↓	Disaster

# Variability



- Statistical Analysis: span (narrow) & average (high)
- Analysis of rest of the architecture is essential



# PLA evaluation

---

- Case study
  - Limited core assets
  - High variation, less reusability
  - COTS dependent configurations
- Structural Assessment
- Potential Improvement



# Improvements

---

- Redefining the component interfaces
- Splitting a component
  - MediaUtilities*: display/process image
- Splitting PLA into two subdomains
  - image storage, image manipulation
- Results: higher service utilization metrics values, and controlled variation



# Discussion

---

- + simple, relative, unbiased, incremental
- + Architecture Description Language(ADL) independent
- + more than assessment of structural quality
- + statistical means for PLA assessment extension
  
- iterative
- needs human judgment
- architectural pruning



# Future Work

---

- Quality attributes: architectural vulnerability, schedulability, etc..
- Architectural pruning
- Collective PLA evaluation
- Case study



# Q&A

---