

# *Software Product Line Research Topics*

**Kyo Kang**

**SPLC2006**

**August 24, 2006**



**Pohang University of Science and Technology  
(POSTECH)**



- Technical advances
- Technology
- Process
- Management



- Paradigm change
  - From single systems to product line/family
- Commonality and variability analysis
  - Feature analysis
- Components and architectures (from objects and collaborations)
  - Variants and variation points
  - “high option potentials”
- Domain specific languages and generators



- Domain analysis
  - Different domains may require different approaches
    - Service analysis may be good for business applications domains
    - Goal analysis may be good for some embedded controller applications domains
    - “Goal -> Service” as a unified method?
  - Modeling mechanisms
    - Feature model is popular but many extensions
      - Should this be standardized?
    - Formalization
  - Deciding the right level of abstraction; how to structure
  - Feature explosion problem
    - How to model, analyze, and manage
  - Feature interaction problem



- Goal-oriented assembly and adaptation of components
  - Knowledge-based adaptation
  - Quality attributes or user-goals (e.g., balanced use of equipments)
- Going from domain analysis to architecture and component design
  - Designing architectures and components based on the analysis results (commonality and variability information)
    - SOA vs. agent-based vs. other architecture models
  - Building variability into architectures and components
  - Selecting appropriate mechanisms for the problem



- Specification of models
  - Reuse contexts and assumptions
- Verification of quality attributes of integrated systems
  - Safety, reliability, etc.
  - Detecting feature interaction problems
- Configuration management
  - Version control of components and architectures with multi-product nature
  - Evolution of the product line itself



- PL for systems in the newly emerging computing environments
  - Service Oriented Architecture
  - Ubiquitous computing environment
    - Dynamic binding of features
  - From compile-time engineering to run-time engineering
    - Embedment of SE knowledge in running systems
- Tools!



- How to change to PL-based organization
  - How to evolve: staged process model for reuse adoption
  - Key process areas
    - Best practices
  - Metrics
    - Key indicators: cost of production, time to market, project completion time, etc.
    - Relationship between reuse, quality, and productivity
    - Relationship between reuse and ROI for sustainability of a reuse program
- Process models
  - Proactive vs. reactive vs. extractive models
    - Best practices
  - PL process vs. agile methods



- ROI analysis
  - Estimating ROI from a reuse program
  - Estimating benefits from strategic market position
- Asset management (How to make PL-based development happen in an organization)
  - Who should develop assets (with variation points)
  - Who should maintain assets (variation management)
  - Who will be responsible for quality assurance
  - Who should enforce the use of assets
  - Models (best practices)
    - Centralized vs. distributed

