Software Product Line Engineering

Lab Class 2

Outline

- Assignment 1 / Tasks 1 + 3
- Recap:
    - Variability / Feature Modeling
    - FeatureIDE
- Assignment 2:

Task 1: Software Product Lines

a) ~ "*A software product line (SPL) is a* ***set of software-intensive systems*** *that share a* ***common, managed set of features*** *satisfying the* ***specific needs*** *of a particular market segment or mission and that are developed from a* ***common set of core assets in a prescribed way****.*" (SEI @CMU)

b) ~ Keywords: Variants, Customizability, Extensibility

c) Not a product line: *Good old* Notepad on Windows

d) ~ Embedded Systems (Automotive, Avionics), Drivers, Operating Systems

Task 1e)

- Android ✓
  - operating system derived from the Linux kernel
  - highly customizable: stock android vs flavored variants, e.g., HTC Sense
- BusyBox✓
  - bundle of *nix tools for embedded systems
  - feature-rich, can be compiled with a variety of commands included/exclude
- GIMP ✓
  - image manipulation software
  - extensible with plug-ins, provides interfaces for automated scripts

Task 3: SPLE development

a) Domain Engineering: Feature Implementations
Application Engineering: Final product/variant

b) SPLE development
- *proactive*: designing a SPL from scratch
- *extractive*: dividing existing software to derive features

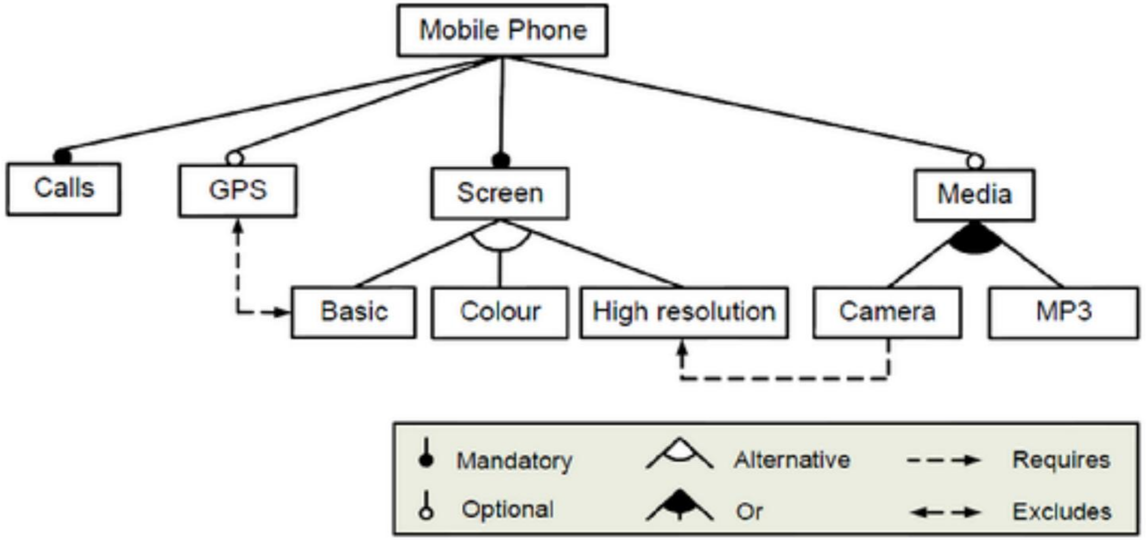
Recap: Feature Models

- *Feature*: Selectable functionality for a SPL
- *Configuration*: (De-)Selection of features

- *Feature Model (FM)*:
  - tells us what configurations are valid and which are not
  - configurations can be checked against a FM (validation)

- FM representations include
  - grammars (validation with parsers)
  - logical formulas (validation with satisfiability solvers)
  - most notably: tree-like **feature diagrams**

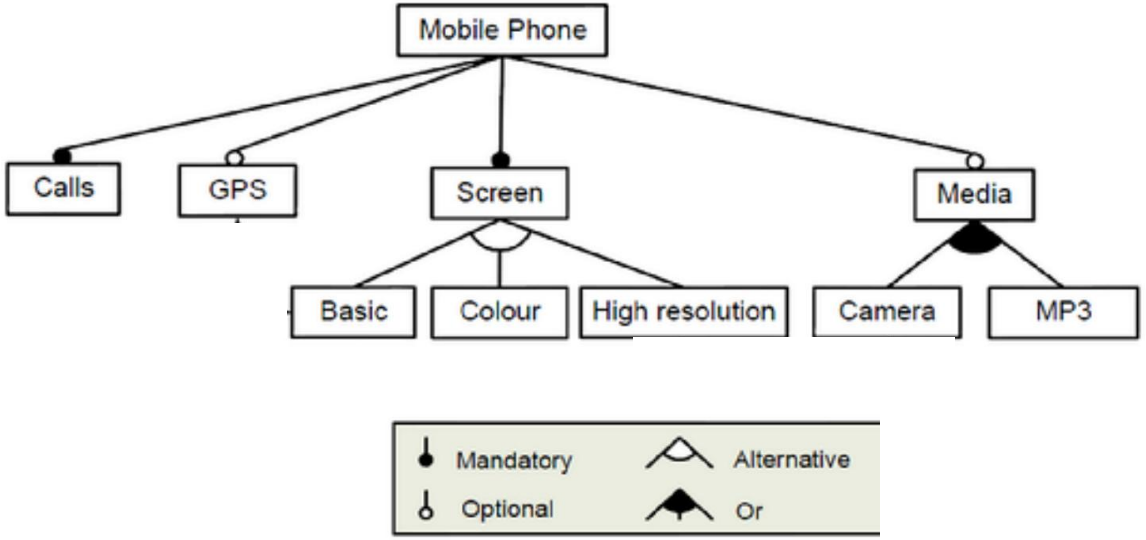
Recap: Feature Model Semantics

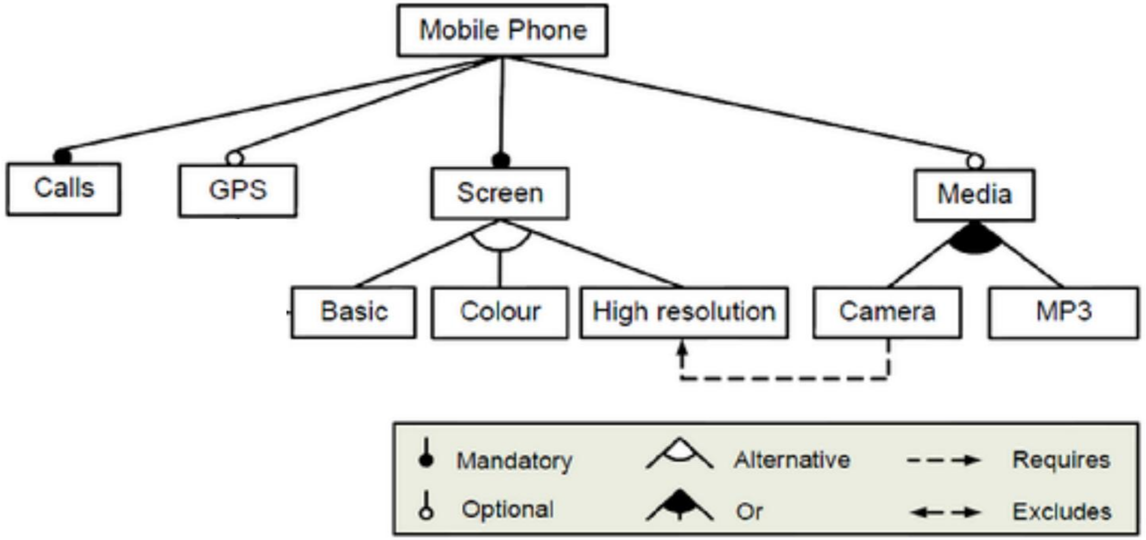


- Optional or mandatory features
- “Alternative Group"
  - exactly one sub-feature selectable
- “Or Group"
  - at least one sub-feature must be selected
- Cross-Tree constraints (CTC)
  - not consistently used…
  - implication or mutual exclusion across sub-trees
  - arrows or plain formulas

Recap: Feature Model Semantics

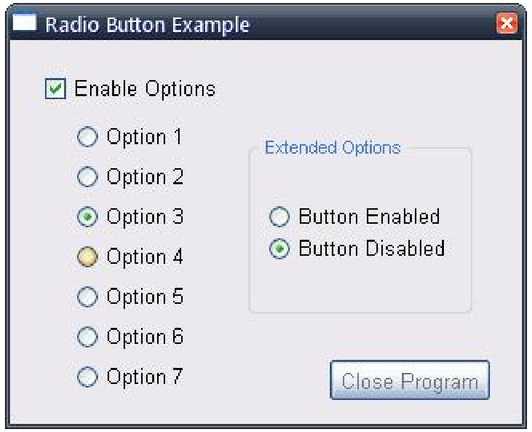
- How many configurations are valid in both cases?





Detour: Numerical Features/Options

- Binary options: enable/disable functionality

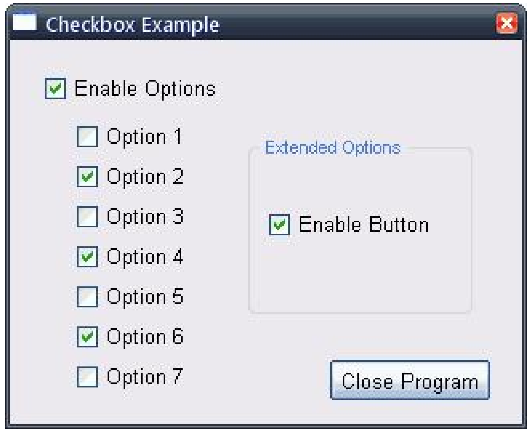


☑ Enable Options

○ Option 1
○ Option 2
● Option 3
○ Option 4
○ Option 5
○ Option 6
○ Option 7

Extended Options

○ Button Enabled
● Button Disabled



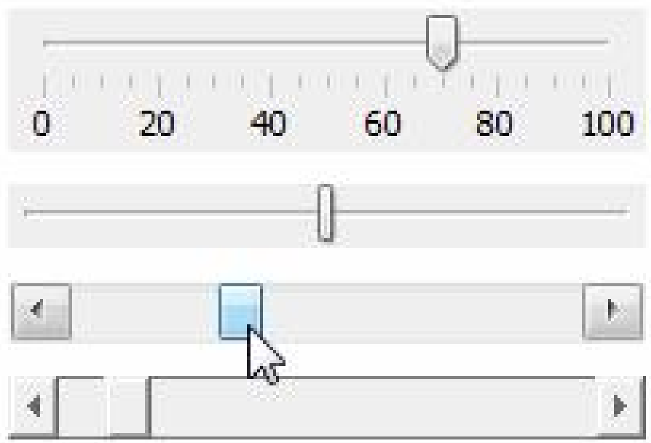
☑ Enable Options

☐ Option 1
☑ Option 2
☐ Option 3
☑ Option 4
☐ Option 5
☑ Option 6
☐ Option 7

Extended Options

☑ Enable Button

- Numerical options: tune functionality



Assignment 2

- Task 1: Domain/Application Engineering
- Task 2: Design Patterns
- Task 3: Feature Model Extraction
- Task 4: Feature Modeling + Warmup with FeatureIDE
  - no Java in this one ;)