Variability Fundamentals

Eduardo Santana de Almeida

esa@dcc.ufba.br

www.dcc.ufba.br/~esa
Based on
Motivation

• Variability
  – The ability or the tendency to change

• Variability modelling
  – **Goal**: To support the development and the reuse of variable development assets
  – Iterative process

• For a SPL - **Essential** property of **Domain** assets

• At each level, variability from the previous level is refined and additional variability is introduced

• Abstraction levels
  – **Common** and **Variable** features of the domain (spl) are identified
  – Domain requirements
  – **Domain architecture**
  – Implementation
  – Test
Motivation

• Defining variability
  – The sum of all activities concerned with the identification and documentation of variability

• Where
  – Domain engineering {definition}
  – Application engineering {exploited}

• Managed Variability
  – Defining and exploiting variability throughout the different life cycle stages of a SPL
  – Issues
    • Supporting activities concerned with defining variability
    • Managing variable assets
    • Supporting activities concerned with resolving variability
    • Collecting, storing, and managing trace information necessary to fulfil these tasks
• Binding time
  – Each process in application engineering binds variability introduced by the corresponding sub-process in domain engineering

• The moment of variability resolution in realisation is often called the binding time of the variability

• You can decide the right time
Definitions

• Examples
  – A software component can support different implementations
  – A search can be active or passive
  – A GUI component for three different mobile phones

• How to identify variability?
  – What does vary?
    • Variability subject – is a variable item of the real world or a variable property of such an item
  – Why does it vary?
    • Stakeholder needs, technical reasons, market, laws
  – How does it vary?
    • Variability object – is a particular instance of a variability subject

• Examples
  – Variability subject – Search | Payment method
  – Variability object – Active, Passive, Content | credit card, cash, bill
Think about it

• Different security standards

• Different customer needs

• Keypad – database stores numerical keys

• Fingerprint – database stores graphical information

• Identification mechanism causes variability in the key database
Definitions (cont.)

• **Variation Point**
  – It is a representation of a variability subject within domain assets enriched by contextual information

• **Variant**
  – It is a representation of a variability object within domain assets

• **Variation Point and Variants**
  – Used to define the variability of a domain [spl]

• **How to identify them?**
  1. To identify the item of the real world that varies {variability subject}
  2. To define a variation point
  3. To define the variants

• **Examples**
  – **Customers, Analysts, Researchers....** – Search: “Passive”, “Active”
  – **Variation Point –**Search Types
  – **Variants –** Passive, Active, Content, Facets, Keywords....
Variability in SPL

- **Variability** is the variability that is modeled to enable the development of customised applications by reusing predefined, adjustable assets.

- **Commonality** denotes features that are part of each application in exactly the same form.

- **Variability in User Interface Language**
  - customers can decide the language **before** the system is installed
  - charged version **any time** to decide the language

- **Commonality in User Interface Language**
  - choice for the preferred language
Definitions (cont.)

• Variability in time
  – It is the existence of different versions of an asset that are valid at different times
  – Single-system engineering or domain {spl} engineering
  – Evolution {configuration management}

I design it
Definitions (cont.)

- **Variability in space**
  - It is the existence of an asset in different shapes at the same time
  - New trend in research
Definitions (cont.)

- External and Internal Variability
  - External – visible to customers
  - Internal – hidden from customers (context-aware)
External Variability

- Stakeholders needs
- Laws and standards

Internal variability

- Refinement of external variability
- Refinement of internal variability
- Technical reasons

Complexity, Business strategy, Market can guide and change it
Documentation of Variability

- **Required information**
  - **What varies?**
    - Documentation of the variation points
  - **Why does it vary?**
    - Textual annotations of variation points and variants
  - **How does it vary?**
    - Documenting the available variants and linking them to domain elements
  - **For whom is it documented?**

- **Benefits**
  - Decision making
  - Communication
  - Traceability

Different models for it
Technical Decision

• Integrate Development Assets
  – hard keeps consistence
  – hard traceability
  – general picture
  – ambiguity

• Different Variability model

It is not a consensus
Orthogonal Model
Orthogonal Variability Model

It is a model that defines the variability of a SPL. It relates the variability defined to other software development models such as feature models, use case models, design models, component models, and test model.

Elements

- Variation points
- Variants
Orthogonal Variability Model

Visible to developers

Visible to developers and customers
Orthogonal Variability Model - Choices

Visible to developers and customers
Variability constraints

- Variant constraint dependency
  - Variant *requires* variant
    - Notification x Interest
  - Variant *excludes* variant

- Variant to Variation Point constraint dependency
  - Variant *requires* variation point
    - Asset publish x Access control
  - Variant *excludes* variation point

- Variation Point constraint dependency
  - Variation Point *requires* Variation Point
    - Publish x Search
  - Variation Point *excludes* Variation Point
Orthogonal Variability Model - Constraints
On the Traceability
Feature Model
Features and Feature Model

• Feature
  – An *end-user-visible characteristic* of a system
  – A *distinguishable characteristic* of a concept that is relevant to some stakeholder of the concept
  – It should have a *concise and descriptive name*

• Elements
  – Feature diagram
  – Feature definitions
  – Composition rules
  – Rationale for features
Feature Modeling: The importance

It is the activity of modeling the common and the variable properties of concepts and their interdependencies and organizing them into a coherent model referred to as a feature model.

- Reusable software
  - Variability

- Key technique
  - To identify and capture variability

- To avoid
  - Relevant features and variations points are not included in the reusable software
  - Many features and variations points are included but never used {complexity, costs}
Feature Modeling – cont.

It is a creative activity

- New features and new knowledge are created during feature modeling using analysis of combinations of variable features
  
  can lead to the discovery of innovative features combinations
  
  and new features
Feature Models

- Provides an abstract (implementation independent), concise, and explicit representation of the variability present in the software

- Represents the common and the variable features of concept instances and ...

- Dependencies between the variable features

- Elements
  - Feature Diagram
  - Semantic descriptions of each features
  - Client programs
  - Exemplar systems
  - Constraints
  - Priorities
Feature Diagrams

• Mandatory Features
• Optional Features
• Alternative Features

There are several extensions on this classification
Mandatory Features

Feature set \{C, f_1, f_2, f_3, f_4\}

32
Optional Features

Feature set \{C\}, \{C, f_1\}, \{C, f_1, f_3\}, \{C, f_2\}, \{C, f_1, f_2\}, \{C, f_1, f_3, f_2\}
Alternative Features

Feature set \{C, f_1, f_3\}, \{C, f_1, f_4\}, \{C, f_1, f_5\}, \{C, f_2, f_3\}, \{C, f_2, f_4\}, \{C, f_2, f_5\}
Example Manual Car Mandatory Features Transmission Horsepower Ar-conditioning Optional Feature Automatic Alternative Features
Example – cont.

Repository

Search
  - Mandatory Features

Publish

Browsing
  - Optional Feature

Active
  - Alternative Features

Passive
Coffee Machine

Coin
  - Dollar
  - Euro

Beverage
  - Alternative Features
    - coffee
    - tea

Ringtone
  - Optional Feature
    - cappucino
Practical Session
Let’s have some fun


http://www.vizzed.com/vizzedboard/retro/game.php?id=4380
Documenting Variability in Requirements

• Domain requirements
  – They encompass requirements common to all applications of the domain as well as variable requirements enabling the creation of different applications
    • Input: Domain design

• Requirements artefacts
  – They are products of the requirements engineering process specified using natural language and/or requirements models
    • Textual requirements
    • Goals
    • Features
    • Use-cases
    • Scenarios....

• Variability in Textual requirements
  – Keywords or phrases {ambiguity}
  – Explicit representation

• Variability using XML/XSLT
  – Hard to perform it...
Variability in Requirements Models

- Feature Models
  - Optional features
  - Alternative features

- Use Case Models
  - Include
  - Extends

- Differences from Single-System Engineering
  - Requirements for each application {isolated}
  - Absence of the commonalities and variability among the applications

- Oriented-reuse
  - Common requirements are documented together with all variable requirements

41
On Requirements and Features

• A feature is a construct used to group related requirements

• Features are a way to abstract from requirements thus

• There is a $n$ to $n$ relation between features and requirements

On the notion of variability, (Svahnberg, van Gurp, Bosch, 2001)

They are simple issues. But, hard to follow and decide in SPL projects
Variability in Design assets

• **Requirements**
  – Customer’s needs

• **Design**
  – How the applications are built

• **Domain architecture**
  – Flexibility
  – It is a core architecture that captures the high-level design for the applications of the domain {spl}
  – Variation points, variants
Variability in the Development View

- Subsystems and Layers
- Components
- Interfaces
- Configuration

- Detailed design
  - Design of components and interfaces
  - Input: Domain architecture
  - Quality concern

- Interface
  - Careful design
    - What interfaces exist
    - Their role
    - Which components should provide or require the interface
Documenting Variability in Test

• Testing
  – Domain engineering
  – Application engineering

• How to develop reusable test artefacts?
  – Variability
  – Research direction

• Test artefacts
  – They are products of the test process containing plans, specifications, and test results
  – Artefacts
    • Test plan
      – Test cases, priorities, tools...
    • Test case
      – Conditions, inputs, outputs...
    • Test case scenario
      – actions
    • Test case scenario step
      – Instructions, results {optional}
    • Test summary report
DO MORE

www.rise.com.br