Engineering Requirements in Product Lines

Mike Mannion, Glasgow Caledonian University
Hermann Kaindl, Vienna University of Technology
Structure

- Introduction
- Product Line Models of Reusable Requirements
- Using a Product Line Model for Building a Single Product
- Verifying a Single Product Model
- Selection Decision Interdependencies
- Tool Support
- Lessons Learned
- Summary
Part I

Introduction to Software Reuse and Product Line Engineering
Product Line Factors

Product Management

- Product Performance
- Pricing Product Marketing Collateral, etc.
- Customer Feedback
- Technology Advances
- Roadmap & Priorities
- Industry Forecasts
- Competitive Intelligence

Product Portfolio

Sales & Marketing Channels
Reuse Framework

- Reuse Management
- Reuse Planning
- Reuse Learning
- Product Line Engineering
Reuse Drivers

Business

- Mass Customisation
- Time to market
- Response to increase in IT system volume, size, complexity
- Increase quality e.g. reliability, interoperability
- Lower costs
- Greater control over sub-contracted work if insist on use of reusable component
- Staff shortages

Technical

- Object-oriented technologies are a catalyst
- “glue” technologies
- Advances in collaboration technologies: internet, intranet, extranet
- Component development toolkits
Reuse Barriers

- Changing markets, changing products
- Priority on short-term profits
- Clarity of strategic objectives
- Unclear return on investment models
- Leadership and management capability
- Changes to organisational structures and culture
- Expertise in reuse processes, methods and tools
Product Line Engineering

• Product Line
  – set of software products sharing a set of common features satisfying the needs of a particular market but containing significant and predictable variability

• Product Line Engineering is a process that delivers software artefacts that can be reused to support the development of new products in the domain

• Product Variation Points, Product Variants

• New product construction grounded in selecting variants at variation points
Software Product Line Engineering Cycles

Requirements

Platform
- Domain Requirements Definition Phase
- System Requirements Definition Phase

Product
- Domain Design Phase
- System Design Phase

Design
- Domain Implementation Phase
- System Implementation Phase

Implementation
- Domain V & V Phase
- System V & V Phase

V & V

⇒ = Development

⇒ = Filtering and Feedback
Product Line Engineering Methods

- 3 Tiered SPL [BigLever, ongoing in 2011]
- Feature-Oriented Reuse Method (FORM) [Kang 2002]
- Feature-Oriented Domain Analysis [Kang 1990]
- PuLSE [Fraunhofer IESE, 2000]
- Organisation Domain Modelling [Simos 1996]
- Domain-Specific Software Architecture [Tracz 1993]
- Synthesis [Software Prod Consortium 1993]
Use Product Line Model to Build Single System Model

Build Product Line Model

Domain Sources e.g. previous specifications, experience, predicted needs

Selected Single System Model

Verified Single System Requirements

new or amended requirements

new requirements

selection

selected single system requirements

error report
Why Requirements Reuse?

- Well-understood requirements are basis for reusable architecture and components
- There are similarities in existing systems and often similar requirements
- Reuse requirements rationale, terminology, expression, validation information
- Acceptance tests and acceptance test plans and procedures can be reused
- Saves considerable effort
Exercise

Why is Requirements Reuse difficult?
Requirements Reuse

• Elicitation difficult because requirements document structures differ eg terminology, writing styles
• Products evolve, need process to manage variability
• Need requirements database to avoid omission, conflict, inconsistency and manage complexity
• need domain expert(s), expensive
Part II
Product Line Models of Reusable Requirements
Product Line Model

- Natural language, atomic
- Tree structure
- Classification of reusable requirements
  - common
  - variable (Variation Point)
  - not reusable
- Mobile phone example
  - Common: There shall be the capability to make a telephone call.
  - Variable: The mobile phone shall have a TV facility.
Product Line Model As Tree Structure
Common Requirements

• **REQ 1**
  — There shall be a telephone number address book facility.

• **REQ 1.1**
  — There shall be the facility to add a telephone number.

• **REQ 1.2**
  — There shall be the facility to search for a telephone number.

• **REQ 1.3**
  — There shall be the facility to delete a telephone number.
Parent-Child Relationship

• Often undefined semantics.
• In our experience elaboration on lower level.
• Mutual dependency of parent and child.
• Both “in” or “out”.
NB: A requirement’s variability profile can evolve over time.
Variation Points

• Definition: any requirement which makes a system different from another in the product line.
• Can come from (many) functional or non-functional requirements.
• We model qualitative variation using Variation Points.
• We model quantitative variation using parameters.
• We model qualitative and quantitative variation using parameterised Variation Points.
Variation Point Types

1. *Mutual exclusion*: a set of mutually exclusive features from which only one can be used in any system in the domain

2. *List of Alternatives*: a set of features which are optional but not mutually exclusive and at least one will be chose.

3. *Option*: A single optional feature

4. Combination of above.
Mutual Exclusion Example

- **REQ 2**
  - The mobile phone shall have a display.
- **REQ 2.1**
  - The mobile phone shall have a black and white display.
- **REQ 2.2**
  - The mobile phone shall have a colour display.
Graphical Representation of a Mutual Exclusion Example

Mutual Exclusion REQ 2 The mobile phone shall have a display.

REQ 2.1 Black and White

REQ 2.4 Colour
List of Alternatives
Example

• **REQ 3**
  — There shall be the facility to make a telephone call.

• **REQ 3.1**
  — A telephone call shall be made by **dialling a number on the numeric keypad**.

• **REQ 3.2**
  — A telephone call shall be made by **pressing a memory recall button to recall a stored number**.

• **REQ 3.3**
  — A telephone call shall be made by **pressing a ring-back facility to dial the number of the last incoming call**.

• **REQ 3.4**
  — A telephone call shall be made by **using speech recognition technology to interpret voice commands**.
Graphical Representation of a List of Alternatives

List of Alternatives REQ 3
There shall be the facility to make a telephone phone call by:

- **REQ 3.1** Dialling number on numeric keypad
- **REQ 3.2** Pressing memory recall button
- **REQ 3.3** Pressing ringback button
- **REQ 3.4** Interpreting voice commands
Option Example

• **REQ 4**
  
The mobile phone shall have an email facility.
Option REQ 4 The mobile phone shall have an email facility.
Variation Point Combination Example

• **REQ 4 (Option)**
  The mobile phone shall have an email facility.

• **REQ 5 (Parent of List of Alternatives)**
  The email facility shall use one of the following protocols.

• **REQ 5.1**
  There shall be the facility to use the Post Office Protocol.

• **REQ 5.2**
  There shall be the facility to use the Internet Message Access Protocol.

• **REQ 5.3**
  There shall be the facility to use the Simple Mail Transfer Protocol.
Option REQ 4 The mobile phone shall have an email facility.

List of Alternatives REQ 5 The email facility shall use one of the following protocols

- REQ 5.1 POP
- REQ 5.2 IMAP
- REQ 5.3 SMTP
**Mobile Phone Example**

**Mutual Exclusion:** The mobile phone shall have a display

- Black and White
- Colour

**List of Alternatives:** There shall be the facility to make a phone call by:

- Voice
- Pressing ringback button
- Pressing memory recall button
- Dialling number on numeric keypad

**Option:** email facility

**List of Alternatives:** email protocol

- POP
- IMAP
- SMTP

There shall be an address book facility.

- Add to address book
- Search address book
- Delete from address book
What type of Variation Point is the availability of Office applications on mobile phones e.g. Word, Excel, Powerpoint?
Parameterised Requirements

• Example: The mobile phone shall respond to \(@X\) commands simultaneously within \(\$Y\) seconds.

• Global parameters i.e. across many requirements (denoted by \(@\)).

• Local parameters i.e. local to this requirement only (denoted by \($\)).

• A parameterised variation point is a variation point that also happens to contain parameters.

• If parameters removed, requirement remains variation point.
## Introducing New Requirements Into A Product Line Model

<table>
<thead>
<tr>
<th>Step</th>
<th>Requirement Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Common requirement</td>
<td>If yes, go to Step 4.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Variable requirement - qualitative</td>
<td>If yes, determine whether the requirement is a Mutual Exclusion, List of Alternatives or option and go to Step 4.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Variable requirement - quantitative</td>
<td>If yes, determine how many parameters there should be and whether they are local or global.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Determine if the requirement can be a leaf child node of an existing Variation Point requirement in the hierarchy or whether a new parent requirement must be formed.</td>
<td>End.</td>
</tr>
</tbody>
</table>
European Space Missions

- Cluster II: to investigate solar wind interaction with the Earth's magnetic field in three dimensions.
- European Remote Sensing (ERS) ERS-1 ERS-2: earth observation satellites measuring data and taking images regardless of cloud and lighting conditions.
- Infrared Space Observatory (ISO): infrared waves to investigate molecular structure of space.
Spacecraft Control Systems

- A Spacecraft Mission Control System (MCS) monitors and controls a satellite.
- Communicates to satellite using spacecraft command sequences.
- MCS has a Mission Planning System (MPS).
- MPS generates commands but can be overridden.
Advanced Mission Planning Infrastructure Viewpoints

- Mission Planning System
  - End User
  - MPS Designer
  - Mission Support
  - Ground Station(s)
  - Spacecraft Operations Engineer
  - Mission Information Base
  - Flight Dynamics
  - MPS User(s)
  - Mission Control System
  - Simulator
  - AMPI Designer

Case Study
Mission Planning Systems

- 4 MPS requirement spec: 100-200 requirements each.
- Product Line Requirement Spec: 539 requirements; 13 viewpoints.
- 66% requirements emanated from existing requirement specifications
  — of these, 49% were common to more than 1 existing requirements specification.
- 12% requirements brand new; 23% from other sources.
- 263 person-hours reviewing previous documentation and talking to staff.
- 450 person-hours creating, reviewing, writing product line requirement specification.
Part I and II
Summary

- Plan, commit and organise for Product Line Software Engineering.
- Product line and systems engineering lifecycles.
- Managing variability.
- Structure and examples of product line models of requirements.
Part III

Using a Product Line Model for Building a Single Product
Free Selection

- Free selection means allowing a single system requirements engineer (user) to browse a product line model and simply copy and paste a single requirement from anywhere in the model to the single system model.

Product Line Model \[\rightarrow\] copy and paste \[\rightarrow\] Single System Model
Problems of Free Selection

- Selecting a single requirement is often not sufficient.
- Random choice can mean illegal choice
  - e.g. 2 mutually exclusive requirements
  - e.g. not choosing generic requirement.
- Untenable number of choices.
- BUT engineers like freedom of choice!!!
Variation Point-Based Selection

- Use tree structure and Variation Points to direct requirements selection.
- Start at one of the roots.
- Traverse depth first.
- Ask user to make a choice at each Variation Point.
- Common requirements are automatically selected if their parents are already selected or if they are a root node.
Example

Mutual Exclusion

List of Alternatives

Option R2.1.3.1

R2.1.3.1.1
Example

Mutual Exclusion

List of Alternatives

Option R2.1.3.1

R1
R1.1
R1.2
R1.2.1
R1.2.2
R2.1
R2.1.1
R2.1.1.1
R2.1.2
R2.1.2.1
R2.1.2.2
R2.1.2.3
R2.1.3
R2.1.3.1
R2.1.3.1.1
R2.2
R2.2.1
R2.2.1.1
R2.2.1.2
R2.2.1.3
Example

List of Alternatives

Mutual Exclusion

Option R2.1.3.1
Example

List of Alternatives

Option

Mutual Exclusion

© Glasgow Caledonian University, Vienna University of Technology
Example

List of Alternatives

Mutual Exclusion

Option R2.1.3.1

R1

R1.1

R1.2

R1.2.1

R1.2.2

R2.1.1.1

R2.1.2.1

R2.1.2.2

R2.1.2.3

R2.1.3

R2.1.3.1

R2.1.3.1.1

R2.2

R2.2.1

R2.2.1.1

R2.2.1.1

R2.2.1.1

R2.2.1.1
Mobile Phone Example

**Mutual Exclusion:** The mobile phone shall have a display
- Black and White
- Colour

**List of Alternatives:** There shall be the facility to make a phone call by:
- Voice
- Pressing ringback button
- Pressing memory recall button
- Dialling number on numeric keypad

There shall be an address book facility.
- Add to address book
- Search address book
- Delete from address book

**Option:** email facility
- List of Alternatives: email protocol
  - POP
  - IMAP
  - SMTP
**Mobile Phone Example**

**Mutual Exclusion:** The mobile phone shall have a display

- Black and White
- Colour

**List of Alternatives:** There shall be the facility to make a phone call by:

- Voice
- Pressing ringback button
- Pressing memory recall button
- Dialling number on numeric keypad

**Option:** email facility

**List of Alternatives:** email protocol

- POP
- IMAP
- SMTP

There shall be an address book facility.

- Add to address book
- Delete from address book
- Search address book
Mobile Phone Example

Mutual Exclusion: The mobile phone shall have a display

- Black and White
- Colour

List of Alternatives: There shall be the facility to make a phone call by:

- Voice
- Pressing ringback button
- Pressing memory recall button
- Dialling number on numeric keypad

Option: email facility

List of Alternatives: email protocol

- POP
- IMAP
- SMTP

There shall be an address book facility.

- Add to address book
- Search address book
- Delete from address book
Mobile Phone Example

**Mutual Exclusion:** The mobile phone shall have a display

- Black and White
- Colour

**List of Alternatives:** There shall be the facility to make a phone call by:

- Voice
- Pressing ringback button
- Pressing memory recall button
- Dialling number on numeric keypad

**Option:** email facility

**List of Alternatives:** email protocol

- POP
- IMAP
- SMTP

There shall be an address book facility.

- Add to address book
- Delete from address book
- Search address book
Mobile Phone Example

**Mutual Exclusion:** The mobile phone shall have a display

**List of Alternatives:** There shall be the facility to make a phone call by:

- Colour
- Voice
- Pressing memory recall button
- Pressing ringback button
- Dialling number on numeric keypad

There shall be an address book facility.

- Add to address book
- Delete from address book
- Search address book

**Option:** email facility

**List of Alternatives:** email protocol

- POP
- IMAP
- SMTP

Delete from address book Add to address book Search address book
Case Study
SCOS 2000 Commanding Requirements

- Spacecraft commands are organised into *tasks*
  - Tasks can be grouped together into *command sub-systems*
  - Tasks have parameters, which have attributes of type and length so that each task can be executed with varying amounts of information.

- Commanding Specification organised as hierarchy of sections of requirements.

- Requirements well written: 10% re-written to make the implicit variability more explicit.

- 2000 requirements; 778 *commanding* requirements
<table>
<thead>
<tr>
<th>Reusable Requirement Type</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td>737</td>
</tr>
<tr>
<td>Parameterised Requirement</td>
<td>14</td>
</tr>
<tr>
<td>Mutual Exclusion Variation Point</td>
<td>3</td>
</tr>
<tr>
<td>List of Alternatives Variation Point</td>
<td>8</td>
</tr>
<tr>
<td>Option Variation Point</td>
<td>15</td>
</tr>
<tr>
<td>Parameterised Variation Point</td>
<td>1</td>
</tr>
</tbody>
</table>
Advantages of Variation Point-Based Selection

- Time & effort savings
  - max 35 choices, not 778
  - system requirements took days not weeks.
- Proportional to number of variation points.
- Time spent on selection implications, not definition, specification, linkage, traceability.
- Selections consistent with the product-line model are made.
- Limited user input because selections pruned.
Part IV
Verifying a Single System Model
# Product Line Model Relations and Formal Definitions

<table>
<thead>
<tr>
<th>Product Line Relation</th>
<th>Formal Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product line model</td>
<td>$T_1 \land T_2 \land \ldots \land T_n$</td>
</tr>
<tr>
<td>Tree (T)</td>
<td>$a_1 \land a_2 \land \ldots \land a_n$</td>
</tr>
<tr>
<td>Parent-Child</td>
<td>$a_i \land a_j$</td>
</tr>
<tr>
<td>Mutual Exclusion Variation Point</td>
<td>$a_i \oplus a_j$</td>
</tr>
<tr>
<td>List of Alternatives Variation Point</td>
<td>$a_i \lor a_j$</td>
</tr>
<tr>
<td>Option Variation Point</td>
<td>$(a_i \lor \neg a_j)$ if i=j, $(a_i \leftrightarrow a_j)$ if i≠j</td>
</tr>
</tbody>
</table>
Product Line Model Definition

- For a product line model $P$ of product line requirements a logical expression can be defined as

$$E(P) = \{ T_1 \land T_2 \land \ldots \land T_n \mid \{ T_i = a_{i1} \ M_{i1} a_{i2} \ M_{i2} a_{i3} \ M_{i3} \ldots M_{i(n-1)} a_{in}; a_{ij} = s(r_{ij}) \}
$$

- where $r_{ij}$ must be a directly reusable requirement or Variation Point;

- and $M_{ij} \in \{ \text{sc}, \text{sa}, \text{ma}, \text{o} \}$
Example

List of Alternatives

R1.1
R1.2
R1.3
R2.1
R2.2
R3.1
R3.2
R3.3
R3.4
R4
R5.1
R5.2
R5.3

Mutual Exclusion

List of Alternatives

Option
Mobile Phone Product Line Model

\[
\begin{align*}
((R1 \land (R1.1 \land R1.2 \land R1.3)) & \land \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (T_1) \\
(R2 \land (R2.1 \oplus R2.2)) & \land \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (T_2) \\
(R3 \land (R3.1 \lor R3.2 \lor R3.3 \lor R3.4)) & \land \ldots (T_3) \\
(R4 \leftrightarrow (R5 \land (R5.1 \lor R5.2 \lor R5.3))) & \ldots \ldots (T_4)
\end{align*}
\]
Free Selection: Example 1

Suppose the selected requirements are:

(R1, R1.1, R1.2, R1.3, R2, R2.1, R3, R3.1, R3.2, R4, R5, R5.1)

The product line logical expression becomes:

(T1), (T2), (T3) and (T4) each evaluate to TRUE.

Hence T1 \land T2 \land T3 \land T4 evaluates to TRUE.
Suppose the selected requirements are:

(R1, R1.1, R1.2, R2, R2.1, R3, R3.1)

Product line logical expression is:

\[(\text{TRUE} \land (\text{TRUE} \land \text{TRUE} \land \text{FALSE})) \land \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (T_1)\]

\[(\text{TRUE} \land (\text{TRUE} \oplus \text{FALSE})) \land \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (T_2)\]

\[(\text{TRUE} \land (\text{TRUE} \lor \text{FALSE} \lor \text{FALSE} \lor \text{FALSE})) \land \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (T_3)\]

\[(\text{FALSE} \leftrightarrow (\text{FALSE} \land (\text{FALSE} \lor \text{FALSE} \lor \text{FALSE} \lor \text{FALSE}))) \ldots \ldots \ldots (T_4)\]

(T_2), (T_3) and (T_4) each evaluate to TRUE but (T_1) evaluates to FALSE because the directly reusable requirement R1.3 was not selected. Hence $T_1 \land T_2 \land T_3 \land T_4$ evaluates to FALSE.
Verification Technique

• Helps verify whether the application requirements satisfy the constraints of the product line model.

• Variation Point-based selection always evaluates to TRUE for any resulting requirements selection of a single system.

• Debugging is a matter of isolating the tree in which the wrong selection combinations have been made.

• Easy to automate.

• Invites the engineer to consider reworking the model.
Part V

Selection Decision Interdependencies
Decision-Making

- large number of variation points and inter-dependencies makes holistic view difficult to see
- inter-dependencies between variation points, between variants, and between variation points and variants
- groups of selection decisions
- groups of inter-dependencies
- different reasons for an inter-dependency
- decision makers require additional information and advice about decisions already taken to inform decisions to be taken
- the transparent and consistent application of selection methods
- different priorities of variation points and variants
- populating of variation points with default variant values before a decision model is used
- making clear the scope and impact of a selection decision i.e. there can be interacting decisions within the same set of artefacts (e.g. features) and interacting decisions between different sets of artefacts (e.g. feature and design and components).
Decision Model Construction & Decision Making Process

- Domain Analysis (Commonality & Variability Analysis)
- Decision Model Meta-Model
- Inter-Dependency Model Meta-Model
- Meta-Model Instance
- Tool: V-Define

- New Product Development (Decision Making)
- Application Model
- Tool: V-Resolve

Output:
- Requirements and inter-dependencies for a product line
- Requirements for a new product
Selection Decision Metamodel
Metal Processing Lines

PLC = Programmable Logic Controller
SCADA = Supervisory Control And Data Acquisition
Case Study: Metal Processing Lines
Mondragón Sistemas de Información

- Product line for the coordinator PLC program
- 9 person-months to develop the decision model with the tools.
- 510 requirements
- 201 common requirements
- 309 variable requirements
<table>
<thead>
<tr>
<th>Decision Model Element type</th>
<th>Number of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decisions</td>
<td>309</td>
</tr>
<tr>
<td>Unrestricted Decision</td>
<td>8</td>
</tr>
<tr>
<td>Bounded Decision</td>
<td>168</td>
</tr>
<tr>
<td>Single Choice List Decision</td>
<td>108</td>
</tr>
<tr>
<td>Multiple Choice List Decision</td>
<td>25</td>
</tr>
<tr>
<td>Group</td>
<td>59</td>
</tr>
<tr>
<td>Collection</td>
<td>1</td>
</tr>
<tr>
<td>Choices</td>
<td>464</td>
</tr>
<tr>
<td>InterDependencies</td>
<td>198</td>
</tr>
<tr>
<td>Inter-Dependency Actions</td>
<td>319</td>
</tr>
</tbody>
</table>
Cost-Benefits

- **Time spent on projects**
  - Blue bars

- **Effort spent on projects**
  - Orange bars

- **Time spent on PL infrastructure**
  - Yellow bars

- **Effort spent on PL infrastructure**
  - Green bars

### Software Development Cost

- **Percentage Distribution**
  - 0% 12% 15% 8% 8%

### Software Development Costs

- **Percentage Distribution**
  - 0% 10% 13% 10% 12%
Part VI
Tool Support
Tools

• Gears (BigLever)
• Pure:: Variants (Pure Systems)
• X-Feature (ETH-Zürich)
• RequiLine (RWTH Aachen University)
• T-REK (Thalès/Telelogic)
• DecisionKing (Christian Doppler Laboratory for Automated Software Engineering Johannes Kepler University)
• V-Define, V-Resolve (European Software Institute)
• TRAM (ESOC)
Lessons Learned

• Metamodels enable consistent information
• Rework model if it breaks during selection or if new requirements to be added
• Role of Product Line board
• Tool support essential
• Complex models – visualisation aids
• Managing Changing Variability
Summary

• Describe a product line engineering lifecycle.
• Build a product line model.
• Develop a single system model from a product line model.
• Use a product line engineering support tool.
Selected work of presenters


