Product-Line Requirements Specification (PRS)

an Approach and Case Study

Stuart R. Faulk
Acronyms

- DARE
- FODA
- FORM
- MPP
- PRS
- SRS
- CoRE
- SCR
Message

• *if developers can re-use* rather than re-do requirements for families of similar systems, we can improve *productivity* while significantly reducing the opportunity for requirements *errors*.
Goal

• Developing a systematic approach to specifying product lines requirements (PRS), then rapidly generating correct requirements specifications (SRS) for applications in the product line.
Scope

• Mission-critical and safety systems
• Requirements errors represent a major source of development problems.
• Requirements are sufficiently similar
• The high up-front verification and validation costs (time, money, manpower)
Requirements Documents in PLs

• Requirements documents in PL:

1. (PRS) specifies the requirements for a program family.
2. (SRS) describes the requirements for a single family member.
Commonalities vs. Variabilities

Ford PRS

SRS1

SRS2

SRS3

SRS4
Approach

• PRS provides the basis for rapidly creating the SRS for a given family member

• Commonalities and variability information

Figure 1: Member SRS derived from PRS
Commonalties in PRS

• Provides a single document characterizing the family as a whole.
• How the family is likely to evolve over time
• The order in which the ability to generate different family members should be implemented.
• How to generate SRSs
Variability information in PRC

- Which requirements may vary from one family member to the next
- The range of variation
- Constraints among variations
- combinations of variations (Dependency analysis).
PRS

- PRS is created in the *domain engineering phase* of product-line development process
- After commonality analysis
Systematic approach

<table>
<thead>
<tr>
<th>Input</th>
<th>Process</th>
<th>Output</th>
</tr>
</thead>
</table>
SCR & CoRE

- SCR: **Software Cost Reduction**
- Practical formal methods for embedded system requirements
- Specifies the behavioral requirements of a system as a set of relations between the quantities monitored by the system and the quantities controlled by the system.
• Describe the required behavior of a software system in terms of a 4 Variable model

Intermediate variables are introduced to simplify writing the relations.
1. conditions (predicates over system state),
2. events (predicates over successive states)
3. terms (expressions over one or more variables)
4. modes (state machines capturing history).
CoRE

• **Consortium Requirements Engineering**
• Provides facilities for modularization and encapsulation of OO requirements without unnecessarily constraining the design.
SRC using CoRE

Figure 2: a simple CoRE Class Structure
Why SRC & CoRE

• Encapsulation and Abstraction:
• It helps put the requirements that vary together in one place so they can be specified, understood or changed relatively independently.
Case Study

• Rockwell Collins Commercial Flight Control System (FCS) product-line
The Collins FCS Example

- Validating the effectiveness of the approach
- Applying it to a portion of a commercial avionics product line
- The Collins Flight Control System
- Concentrated on a part of the flight guidance system, the mode control logic.
Systematic approach

<table>
<thead>
<tr>
<th>Input</th>
<th>Process</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Definition</td>
<td>1. Organize the Domain Definition</td>
<td>PRC in SCR &amp; CoRE</td>
</tr>
<tr>
<td>1. PL’s terminology</td>
<td>2. Create a Decision Model</td>
<td></td>
</tr>
<tr>
<td>2. Commonalities</td>
<td>3. Encapsulate variations in CoRE</td>
<td></td>
</tr>
<tr>
<td>3. Variabilities,</td>
<td>4. Define variations</td>
<td></td>
</tr>
<tr>
<td>4. Dependencies</td>
<td>5. Provide Traceability</td>
<td></td>
</tr>
</tbody>
</table>

Diagram:
- Input: Domain Definition, PL’s terminology, Commonalities, Variabilities, Dependencies
- Process: Organize the Domain Definition, Create a Decision Model, Encapsulate variations in CoRE, Define variations, Provide Traceability
- Output: PRC in SCR & CoRE
1- Organize Domain Definition

- Structure the information in the Domain Definition so such requirements can be found in one part of the document.
- Most common first, subordinate dependencies and preserve traceability

4.7.2 Lateral Modes

4.7.2 C 1 Every FCS has one or more lateral modes exactly one of which must be active at any time.
4.7.2 C 2 Every FCS has a default (basic) lateral mode
4.7.2 V 3 The basic lateral mode varies from aircraft to aircraft.
4.7.2 V 4 The FCS may or may not select the basic lateral mode upon transfer of flight guidance computations.
4.7.2 V 5 The set of lateral modes varies from aircraft to aircraft.
   4.7.2 V 5.1 An FCS may or may not have a Roll Mode.
      4.7.2 C 5.1.1 Every FCS that has a roll mode uses a roll reference.
      4.7.2 C 5.1.2 The roll reference is synchronized when the SYNC switch is enabled and pressed with the flight director on.
      4.7.2 V 5.1.3 There may or may not be a roll knob to adjust the roll reference.
         4.7.2 V 5.1.3.1 The roll knob may have a detent angle of 0, 5, or 6 degrees.
      4.7.2 V 5.1.4 Roll/Heading transition angle can assume values of 5 or 6 deg.

Figure 3: Example of organized domain definition from Collins Flight Control System mode requirements
2- Create Decision Model

- Represents the set of choices that distinguish the members of a family

4.7.2. V 5. I An FCS may or may not have a Roll Mode.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Decision</th>
<th>Values</th>
<th>Traceability</th>
</tr>
</thead>
<tbody>
<tr>
<td>LatBasic</td>
<td>What is the basic lateral mode?</td>
<td>Roll, Heading</td>
<td>4.7.2 V 3</td>
</tr>
<tr>
<td>RollMode</td>
<td>Does the FCS have a Roll Mode?</td>
<td>Yes, No</td>
<td>4.7.2.V 5.1</td>
</tr>
<tr>
<td>RollKnob</td>
<td>Does the FCS have a Roll Knob?</td>
<td>Yes, No</td>
<td>4.7.2 V 5.1.3</td>
</tr>
<tr>
<td>DetentAngle</td>
<td>What is the detent angle of the Roll Knob?</td>
<td>0, 5, 6</td>
<td>4.7.2 V 5.1.3.1</td>
</tr>
<tr>
<td>RollHdgAngle</td>
<td>What is the Roll Heading Transition Angle?</td>
<td>5, 6</td>
<td>4.7.2 V 5.1.4</td>
</tr>
<tr>
<td>RollLimit</td>
<td>What is the Bank Limit in Roll Mode?</td>
<td>31.5, 32</td>
<td>4.7.2 V 5.1.5</td>
</tr>
</tbody>
</table>

Figure 4: Decision table for FCS Roll Mode
3- Encapsulating Variations in CoRE

- Input to this stage includes the Domain Definition and the decision model

Figure 5: Illustration of a PRS class structure
4- Define Variation

Table 1 - mode_Active_Lateral Transition Table

<table>
<thead>
<tr>
<th>From</th>
<th>Events</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>@T(Flight_Director_On) OR @Basic_Selected</td>
<td>mode _&lt;&lt;LatBasic&gt;&gt;</td>
</tr>
<tr>
<td>Any</td>
<td>@T(Nav_Guidance.term_Active)</td>
<td>mode_NAV</td>
</tr>
</tbody>
</table>

<<if RollMode=Yes then "

Any          | @T(Roll_Guidance.term_Active) | mode_ROLL     

">>

Instantiated with the values: LatBasic = ROLL and RollMode = Yes results in the following table:

<table>
<thead>
<tr>
<th>From</th>
<th>Events</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>@T(Flight_Director_On) OR @Basic_Selected</td>
<td>mode_ROLL</td>
</tr>
<tr>
<td>Any</td>
<td>@T(Nav_Guidance.term_Active)</td>
<td>mode_NAV</td>
</tr>
<tr>
<td>Any</td>
<td>@T(Roll_Guidance.term_Active)</td>
<td>mode_ROLL</td>
</tr>
</tbody>
</table>

Figure 6: Generating a table for an FCS family member
Questions

• Discussion: Article hard to understand without reading the authors related work on SCR and CoRE since these were used throughout examples in the article.

• What is meta-text constructs applied to the PRS?
Questions

• How feasible this approach is for the systems that are not safety- or mission-control systems?
• What is a practical formal method? Where exactly the approach applies practical formal method?
• The approach is demonstrated with the help of an existing product family where the domain is well-understood by all the stakeholders. How well it will work for a totally new product family?
• The decision table looks confusing to me. Is not it possible to create decision diagram instead of a decision table?
Questions

• Can PRS be used along with the requirements-based development method?
• Has this process changed in current literature?
• Are there any tools that exist to help automate step 4 (Defining Variations)?
• Can we use cases to capture the requirements from an external point of view?
• How can we use (CoRE) and implement it in Real time environment by using both SRS and PRS?
Questions

• What is the significance of a decision model?
• Do you agree with the idea to have both PRS and SRS?