Variability: From Software Product Lines to Self-Adaptive Systems

Danny Weyns
www.lnu.se/adaptwise

Dynamic Software Product Lines, DSPL
Florence, September 15, 2014
Observation

Variability
“The ability of software systems or artifacts to be adjusted for different contexts”*
Mainly been studied in the context of software product lines

Variability beyond traditional software product lines
Runtime service composition
Self-adaptation
Dynamic software product lines…

* van Gurp, Bosch, Svahnberg, On the Notion of variability in software product lines, WICSA 2011
Research Objective

Addressing variability in a more holistic manner

Analyze and characterize handling of variability in software systems from researchers viewpoint

- Identify trends in research on variability handling in software systems in general
- Assess evidence of variability research, identify open problems and new opportunities
- Obtain a more rigorous understanding of variability in software engineering
Research Method

Systematic Literature Review*

Research Questions

1. What methods to handle variability in software systems exist?
   - Design time vs runtime variability?
   - Quality attributes addressed?
   - Activities in software development affected?

2. What is the evidence that motivates the adoption of existing methods?

3. What are the limitations of existing methods?
Search Strategy

Manual search of articles published at quality venues since 2000 (18 conferences, 13 journals)

Complemented with targeted automatic search

Pair-wise review of batches of papers
## Data Collection

<table>
<thead>
<tr>
<th>#</th>
<th>Field</th>
<th>Research question</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Author(s)</td>
<td>n/a</td>
</tr>
<tr>
<td>F2</td>
<td>Year</td>
<td>n/a</td>
</tr>
<tr>
<td>F3</td>
<td>Title</td>
<td>n/a</td>
</tr>
<tr>
<td>F4</td>
<td>Venue (journal or conference)</td>
<td>n/a</td>
</tr>
<tr>
<td>F5</td>
<td>Keywords</td>
<td>n/a</td>
</tr>
<tr>
<td>F6</td>
<td>Abstract</td>
<td>n/a</td>
</tr>
<tr>
<td>F7</td>
<td>Citation count</td>
<td>RQ2</td>
</tr>
<tr>
<td>F8</td>
<td>Quality score</td>
<td>RQ2</td>
</tr>
<tr>
<td>F9</td>
<td>Runtime quality attributes</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>F10</td>
<td>Design time quality attributes</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>F11</td>
<td>Level of tool support</td>
<td>RQ3</td>
</tr>
<tr>
<td>F12</td>
<td>Evidence level</td>
<td>RQ2</td>
</tr>
<tr>
<td>F13</td>
<td>Activities addressed by approach</td>
<td>RQ1.2</td>
</tr>
<tr>
<td>F14</td>
<td>Limitations</td>
<td>RQ3</td>
</tr>
<tr>
<td>F15</td>
<td>Types of variability</td>
<td>RQ1.1</td>
</tr>
<tr>
<td>F16</td>
<td>Type of study</td>
<td>RQ2</td>
</tr>
</tbody>
</table>
Main Results

Demographic data
Design time vs runtime variability
Engineering activities addressed
Level of evidence
Reported limitations
Demographic Data

Manually searched 15430 papers
Resulted in 196 papers for data collection
Design vs Runtime Variability

128 studies focus on design time variability
  Top: flexibility, reusability, modifiability

50 studies focus on runtime variability
  Top: performance, flexibility, availability

18 remaining studies no clear differentiation
Engineering Activities Addressed

<table>
<thead>
<tr>
<th></th>
<th>Absolute</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement engineering</td>
<td>32</td>
<td>13.0</td>
</tr>
<tr>
<td>Architecture</td>
<td>72</td>
<td>29.2</td>
</tr>
<tr>
<td>Design</td>
<td>86</td>
<td>34.8</td>
</tr>
<tr>
<td>Implementation</td>
<td>25</td>
<td>10.1</td>
</tr>
<tr>
<td>Testing and verification</td>
<td>5</td>
<td>2.0</td>
</tr>
<tr>
<td>Maintenance</td>
<td>27</td>
<td>10.9</td>
</tr>
</tbody>
</table>

130 studies focus on one activity
22 studies focus on two activities
One study focuses on 6 activities
Level of Evidence

Level 0: no evidence
Level 1: demonstration or toy example
Level 2: expert opinions
Level 3: academic studies (experiments, etc.)
Level 4: industrial studies (e.g., case studies)
Level 5: evidence from industrial practice
Observations Related to Evidence (1/2)

Quality score of publications
Q1: rationale
Q2: context of the research
Q3: research design
Q4: data available
Q5: role of researchers
Q6: discussion credibility results
Observations Related to Evidence (1/2)

Quality score of publications
Q1: rationale
Q2: context of the research
Q3: research design
Q4: data available
Q5: role of researchers
Q6: discussion credibility results

Average of empirical papers = 3.8
None empirical papers = 2.8
Observations Related to Evidence (2/2)

<table>
<thead>
<tr>
<th>Study ID</th>
<th>Citation count</th>
<th>Average count</th>
<th>citation count</th>
<th>Total score</th>
<th>quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>428</td>
<td>71</td>
<td>6.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>198</td>
<td>15</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>130</td>
<td>12</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>168</td>
<td>17</td>
<td>3.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>229</td>
<td>25</td>
<td>5.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>126</td>
<td>14</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>173</td>
<td>35</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>491</td>
<td>49</td>
<td>3.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>192</td>
<td>21</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>197</td>
<td>20</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>107</td>
<td>13</td>
<td>3.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>150</td>
<td>30</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>202</td>
<td>29</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>114</td>
<td>19</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>193</td>
<td>21</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>128</td>
<td>11</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>108</td>
<td>12</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>116</td>
<td>11</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>124</td>
<td>11</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>171</td>
<td>21</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>218</td>
<td>44</td>
<td>3.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>101</td>
<td>34</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>197</td>
<td>20</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>380</td>
<td>48</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>345</td>
<td>43</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>681</td>
<td>52</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>167</td>
<td>28</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>225</td>
<td>25</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>106</td>
<td>35</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No correlation between number of citations and total quality score of the papers
Reported Limitations

**Correctness and consistency** are difficult to guarantee; need for verification techniques

**Impact on quality attributes**, in particular performance overhead, support for security needs, dynamic features

**Poor user-friendliness**, lack of flexibility, support for users (2 on 3 studies provide some kind of tool support)

**Identification of variability and variants**, need for advanced techniques such as data mining
Dimensions of Variability

Aim to make a step forward in establishing a more holistic characterization of variability in software systems

Capture key facets of variability ("dimensions")
Baseline to identify variability needs and compare approaches

We identified initial set of dimensions from a number of existing approaches documented in the literature

Pilot (20% papers with best quality score) to test dimensions and identify different options ("domain") for each dimension
## Dimensions of Variability: Two Clusters

<table>
<thead>
<tr>
<th>Type</th>
<th>Introduction and specification of variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>Domain (number of studies in pilot)</td>
</tr>
</tbody>
</table>
| Requirement type | Functional (15)  
Quality (17) |
| Representation | Feature model (4)  
Rules/conditions (15)  
Variant labels/annotations (7)  
Profiles (4)  
Change scenarios (2) |
| Artifact | Scenario (5)  
Business process (1)  
Architecture (14)  
Component (14)  
Code fragment (5)  
Variable (2) |
| Orthogonality | Separated (15)  
Integrated (7) |

<table>
<thead>
<tr>
<th>Mechanisms</th>
<th>How does variability take place or is brought about</th>
</tr>
</thead>
</table>
| Trigger | Stakeholder (14)  
Business process (1)  
System (5)  
Environment (8) |
| Realization technique | Reorganization (7)  
Selection (18)  
Value assignment (2)  
Code generation (2) |
| Time of binding | Software construction/evolution (10)  
Runtime (13) |
| Automation | Manual (3)  
Semi-automatic (10)  
Automatic (9) |
# Dimensions of Variability: Two Clusters

<table>
<thead>
<tr>
<th>Type</th>
<th>Introduction and specification of variability</th>
<th>Mechanisms</th>
<th>How does variability take place or is brought about</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension</strong></td>
<td><strong>Domain (number of studies in pilot)</strong></td>
<td><strong>Trigger</strong></td>
<td>Stakeholder (14) Business process (1) System (5) Environment (8)</td>
</tr>
<tr>
<td>Requirement type</td>
<td>Functional (15) Quality (17)</td>
<td><strong>Realization technique</strong></td>
<td>Reorganization (7) Selection (18) Value assignment (2) Code generation (2)</td>
</tr>
<tr>
<td>Representation</td>
<td>Feature model (4) Rules/conditions (15) Variant labels/annotations (7) Profiles (4) Change scenarios (2)</td>
<td><strong>Time of binding</strong></td>
<td>Software construction/evolution (10) Runtime (13)</td>
</tr>
<tr>
<td>Orthogonality</td>
<td>Separated (15) Integrated (7)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Requirement Type

Functional
   E.g. support for different types of missions in a command and controller simulator provided by optional components

Quality
   E.g. an alternative service is selected and invoked when the original service is not available
Representation

Feature model
  E.g., feature graph with links between features and requirement specifications to represent variability

Rules/conditions
  E.g., expressions that refer to elements of design patterns to express variability of patterns

Variant labels/annotations
  E.g., component annotated with properties (e.g., required resources) to discriminate between alternatives enabling dynamic selection of variants

Profiles
  E.g., use of meta-data (composition policy) together with context data to dynamically select and add service (e.g., authentication)

Change scenarios
  E.g., scenarios that specify a specific set of choices supporting decision making for variability of a product line
Trigger

**Stakeholder**
E.g., user commands trigger the selection of variants of GUI forms that guide the user through the process of generating legal documents

**Business process**
E.g., firing conditions of a transition in a business process trigger the execution of a variant of a region of the process

**System**
E.g., a service system optimizes its workflow by replacing services with services of better quality when the current quality differs from the prediction by more than a give threshold

**Environment**
E.g., changes in the context, such as quality of the network or physical location trigger the system to replace components by variants that guarantee better utility
Time of Binding

Software construction/evolution
   E.g., an analyst selects the best architecture among a set of variants during system development based on the impact analysis for a set of modifiability scenarios

Runtime
   E.g., dynamic content adaptations of mobile devices based on device profiles and users’ service level agreements by selecting matching services from a repository of variants
Conclusions

Variability is a key issue of many software systems today, if not all

SLR aimed at studying variability handing in software systems beyond SPL

Resulting classification offers vocabulary to support specification of variability properties under consideration
  - Enables comparison of variability solutions across different systems/domains
  - Potential asset for identifying related work in specific areas of variability

Step towards a unifying and integrated perspective on variability in software systems
  - Spanning currently disparate or loosely connected fields/communities
  - Potential to accelerate exchange of ideas and solutions between fields
Examples in Our Planned & Ongoing Research

Variability as a mechanism to support self-adaptation

Focus on multiple concerns

MAPE-K feedback loop(s):
- Maintain(s) models of different quality properties
- Reason(s) about tradeoffs of architectural configurations
- Select(s) most suitable variant to adapt system to the given context
Examples in Our Planned & Ongoing Research

Self-adaptation as a variability mechanism

Autonomic software product lines*
- Separate domain logic from adaptation logic in SPL
- Adaptation logic = reusable building blocks of feedback loops
- Complemented with supporting artifacts
  Domain quality attribute scenarios
  Responsibility structures
  Reasoning framework
- Supporting systematic reuse for quality properties of interest

* Abbas, Andersson, Löwe. Autonomic Software Product Lines (ASPL), ECSA 2010
Abbas, Andersson, Weyns, Modeling variability in product lines using domain quality attribute scenarios,, WICS/ECSA Companion Volume, 2011
Thank you!
Questions, comments?

http://homepage.lnu.se/staff/daweea/papers/2013TSE.pdf

Technical report with all SLR material:
Bibliography

van Gurp, Bosch, Svahnberg, On the Notion of variability in software product lines, WICSA 2011


Abbas, Andersson, Löwe, Autonomic Software Product Lines (ASPL), ECSA 2010

Abbas, Andersson, Weyns, Modeling variability in product lines using domain quality attribute scenarios, WICS/ECSA Companion Volume, 2011

Nadeem Abbas, Andersson, Weyns Knowledge evolution in autonomic software product lines, DSPL, 2011