First configuration

- Shopping cart, Bonus, Shipping Method (Economical, Fast)

### Buy Goods

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the checkout option.</td>
<td>Present the items in the shopping cart and the amount to be paid. The user can remove items from shopping cart.</td>
</tr>
<tr>
<td>Select the confirm option.</td>
<td>Request bonus and payment information.</td>
</tr>
<tr>
<td>Fill in the requested information and select the proceed option.</td>
<td>Request the shipping method and address.</td>
</tr>
<tr>
<td>Select one of the shipping methods (Economical, Fast), fill in the destination address and select the proceed option.</td>
<td>Calculate the shipping costs.</td>
</tr>
<tr>
<td>Confirm the purchase.</td>
<td>Execute the order and send a request to the Delivery System in order to dispatch the products</td>
</tr>
</tbody>
</table>

### Search

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the search product option.</td>
<td>Present the available search criteria.</td>
</tr>
<tr>
<td>Choose one of the available criteria and select the proceed option.</td>
<td>Request the search criteria.</td>
</tr>
<tr>
<td>Inform the search criteria.</td>
<td>Retrieve the products that satisfy the search criteria. Show a list with the resulting products.</td>
</tr>
</tbody>
</table>
### Update User Preferences, Shipping Method (Economical)

#### Buy Goods

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the buy product option.</td>
<td>Present the selected product. The user can change the quantity of items that he wants to buy....</td>
</tr>
<tr>
<td>Select the confirm option.</td>
<td>Request payment information.</td>
</tr>
<tr>
<td>Fill in the requested information and select the proceed option.</td>
<td>Request the shipping method and address.</td>
</tr>
<tr>
<td>Select one of the shipping methods (Economical), fill in the destination address and select the proceed option.</td>
<td>Calculate the shipping costs.</td>
</tr>
<tr>
<td>Confirm the purchase.</td>
<td>Execute the order and send a request to the Delivery System in order to dispatch the products</td>
</tr>
<tr>
<td></td>
<td>Register user preferences</td>
</tr>
</tbody>
</table>

#### Search

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the search product option.</td>
<td>Present the available search criteria.</td>
</tr>
<tr>
<td>Choose one of the available criteria and select the proceed option.</td>
<td>Request the search criteria.</td>
</tr>
<tr>
<td>Inform the search criteria.</td>
<td>Retrieve the products that satisfy the search criteria. Show a list with the resulting products.</td>
</tr>
<tr>
<td></td>
<td>Register user preferences</td>
</tr>
</tbody>
</table>
According to the product line approach, we should be able to generate those specifications from reusable assets.
However

Existing approaches for representing variability in textual scenarios yield to tangling and scattering.

- Feature specification scattered throughout different scenarios
- Variant behavior tangled with common behavior
- Variability management tangled with scenario specification
However

Existing approaches for representing variability in textual scenarios yield to tangling and scattering.

- Feature specification scattered throughout different scenarios
- Variant behavior tangled with common behavior
- Variability management tangled with scenario specification
Buy Products Scenario
Main Flow
01 Select the [VP1] option.
02 [VP2]
03 Select the confirm option.
04 [VP3]
05 Fill in the requested information and select the proceed option.
06 Request the shipping method and address.
07 Select the [VP4] shipping method, fill in the destination address and select the proceed option.
08 Calculate the shipping costs.
09 Confirm the purchase.
10 Execute the order and sends a request to the Delivery System in order to dispatch the products.
11 {[VP5] Register the user preferences.}

Products definition:
P = (P1, P2)

Variation points:
VP1 = if (P == P1) then (checkout)
else (buy product)
VP2 = if (P == P1)
then (Presents the items in the shopping cart...)
else (Present the selected product. The user...)
VP3 = if (P == P1)
then (Requests bonus and payment information.)
else (Requests payment information.)
VP4 = if (P == P1) then (Economical, Fast)
else (Economical)
VP5 requires (P == P1)
<table>
<thead>
<tr>
<th>Id</th>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(a)</td>
<td>Select the checkout option. [ShoppingCart]</td>
<td>Present the items in the shopping cart and the amount to be paid. The user can remove items from shopping cart.</td>
</tr>
<tr>
<td>1(b)</td>
<td>Select the buy product option. [not ShoppingCart]</td>
<td>Present the selected product. The user can change the quantity of items that he wants to buy. Calculate and show the amount to be paid.</td>
</tr>
<tr>
<td>2(a)</td>
<td>Select the confirm option. [Bonus]</td>
<td>Request bonus and payment information.</td>
</tr>
<tr>
<td>2(b)</td>
<td>Select the confirm option. [not Bonus]</td>
<td>Request payment information.</td>
</tr>
<tr>
<td>3</td>
<td>Fill in the requested information and select the proceed option.</td>
<td>Request the shipping method and address.</td>
</tr>
<tr>
<td>4</td>
<td>Select the $ShippingMethod$, fill in the destination address and select the proceed option.</td>
<td>Calculate the shipping costs.</td>
</tr>
<tr>
<td>5</td>
<td>Confirm the purchase.</td>
<td>Execute the order and send a request to the Delivery System in order to dispatch the products.</td>
</tr>
<tr>
<td>(6)</td>
<td>[Update User Preferences]</td>
<td>Update user preferences.</td>
</tr>
</tbody>
</table>
MSVCM (proposed solution)
### SC01: Buy products.

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill in the requested information and select the proceed option.</td>
<td>Request the shipping method and address. @ConfirmPurchase</td>
</tr>
<tr>
<td>Select one of the shipping methods &lt;SM&gt;, fill in the destination address and select the proceed option.</td>
<td>Calculate the shipping costs.</td>
</tr>
<tr>
<td>Confirm the purchase.</td>
<td>Execute the order and send a request to the Delivery System, in order to dispatch the products. @UpdatePreferences</td>
</tr>
</tbody>
</table>

### SC02: Search products.

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the search product option.</td>
<td>Present the available search criteria.</td>
</tr>
<tr>
<td>Inform the search criteria.</td>
<td>Retrieve the products that satisfy the search criteria. Show a list with the resulting products. @UpdatePreferences</td>
</tr>
</tbody>
</table>

### ADV01: Buy products with shopping cart and bonus.

**Pointcut: before** @ConfirmPurchase

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the checkout option.</td>
<td>Present the items in the shopping cart and the amount to be paid. The user can remove items from her shopping cart.</td>
</tr>
<tr>
<td>Select the confirm option.</td>
<td>Request bonus and payment information.</td>
</tr>
</tbody>
</table>

### ADV02: Buy products without shopping cart and bonus.

**Pointcut: before** @ConfirmPurchase

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the buy product option.</td>
<td>Present the selected product. The user can change the amount of items that she wants to buy.</td>
</tr>
<tr>
<td>Select the confirm option.</td>
<td>Request payment information.</td>
</tr>
</tbody>
</table>

### ADV03: Update user preferences.

**Pointcut: after** @UpdatePreferences

<table>
<thead>
<tr>
<th>User Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Update user preferences.</td>
</tr>
</tbody>
</table>
... and a separate configuration knowledge (CK)

<table>
<thead>
<tr>
<th>Feature expression</th>
<th>Transformations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy Goods Service</td>
<td>select Scenario SC01</td>
</tr>
<tr>
<td>Search Products Service</td>
<td>select Scenario SC02</td>
</tr>
<tr>
<td>Shopping Cart and Bonus</td>
<td>evaluate Advice ADV01</td>
</tr>
<tr>
<td>not (Shopping Cart and Bonus)</td>
<td>evaluate Advice ADV02</td>
</tr>
<tr>
<td>Update User Preferences</td>
<td>evaluate Advice ADV03</td>
</tr>
<tr>
<td>Shipping Method</td>
<td>bind Parameter SM to Shipping Method</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Execute the order and send a request to the Delivery System to dispatch the products.

Confirm the purchase.

Select one of the available shipping methods, fill in the destination address and proceed.

Calculate the shipping costs.

Feature Expression Transformations

<table>
<thead>
<tr>
<th>Feature Expression</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy</td>
<td>Add Customer ID</td>
</tr>
<tr>
<td>FT Shopping Cart and Bonus</td>
<td>Add Customer ID</td>
</tr>
<tr>
<td>Shipping Method</td>
<td>Add Customer ID</td>
</tr>
<tr>
<td>User Preferences</td>
<td>Add Customer ID</td>
</tr>
<tr>
<td>Shipping Method</td>
<td>Add FT to Shipping Method</td>
</tr>
</tbody>
</table>

User action System response

Id: ADV01
Description: Buy a specific product.
Before: [CustomerPreferences]
S3 Inform the search criteria.
satisfy the search criteria.
Retrieve the products that match the criteria.
Show a list with the resulting products.

User action Id: SC03
Description: Search for products.
after: [CustomerPreferences]
Select the confirm option.

Update User Preferences
Shipping Method
shopping cart and the
amount to be paid. The user
can remove items from the
shopping cart.

Select the checkout option.
Present the items in the
shopping cart and the
amount to be paid. The user
can remove items from the
shopping cart.

Select one of the available shipping methods, fill in the destination address and proceed.

Select the buy product option.

Calculate the shipping costs.

Select the confirm option.
Request bonus and payment information.
Select the confirm option.
Request the shipping method and address.

Select the checkout option.
shopping cart and the
amount to be paid. The user
can remove items from the
shopping cart.

Select the confirm option.
Update User Preferences.

Search Products

Product Specific Use Case Model

Weaving Process

Buy Goods

<table>
<thead>
<tr>
<th>User action</th>
<th>System response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the checkout option.</td>
<td>Present the items in the shopping cart and the amount to be paid. The user can remove items from the shopping cart.</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Select the favorite shipping method (Economical, Fast), fill in the destination address and proceed.</td>
<td>Calculate the shipping costs.</td>
</tr>
<tr>
<td>Confirm the purchase.</td>
<td>Execute the order and send a request to the Delivery System to dispatch the products.</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Update the user preferences based on the search results or purchased items.</td>
<td>Update the user preferences based on the search results or purchased items.</td>
</tr>
</tbody>
</table>

Product Configuration

Configuration Knowledge

SPL
Use Case Model

Feature Model

Weaving Process
Abstract syntax

```haskell
type ConfigurationKnowledge = [ConfigurationItem]
data ConfigurationItem = ConfigurationItem {
  expression :: FeatureExpression,
  transformations :: [Transformation]
}
type Transformation = SPL → Product → Product
```

Interpreter-based semantics

```haskell
weavingProcess fm pc ck spl = refine ts spl newProduct
  where
    ts = concat [transformations c | c ← ck, eval pc (expression c)]
    newProduct = ...
    refine [] pla product = product
    refine (t : ts) spl product = refine ts spl (t spl product)
```
Abstract syntax

```typescript
type ConfigurationKnowledge = [ConfigurationItem]
data ConfigurationItem = ConfigurationItem {
  expression :: FeatureExpression,
  transformations :: [Transformation]
}

// Interpreter-based semantics

weavingProcess fm pc ck spl = refine ts spl newProduct

where
  ts = concat [transformations c | c ← ck, eval pc (expression c)]
  newProduct = ...
  refine [] pla product = product
  refine (t : ts) spl product = refine ts spl (t spl product)
```
Each transformation implements one source of variability...

- `selectScenario` $\rightarrow$ variability in function
- `bindParameter` $\rightarrow$ variability in data
- `evaluateAdvice` $\rightarrow$ variability in control flow
Each transformation implements one source of variability...

- `selectScenario` $\mapsto$ variability in function
- `bindParameter` $\mapsto$ variability in data
- `evaluateAdvice` $\mapsto$ variability in control flow
and is formally described as a crosscutting mechanism.

[Masuhara and Kiczales, 2003].
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O$</td>
<td>Specific scenarios with extensions</td>
</tr>
<tr>
<td>$O_{jp}$</td>
<td>Scenarios and steps of scenarios</td>
</tr>
<tr>
<td>$L$</td>
<td>{UCM, CK, PC}</td>
</tr>
<tr>
<td>$UCM_{ID}$</td>
<td>Pointcuts of declared advices</td>
</tr>
<tr>
<td>$CK_{ID}$</td>
<td>Mapping of features to advices</td>
</tr>
<tr>
<td>$PC_{ID}$</td>
<td>Selected features related to advices</td>
</tr>
<tr>
<td>$UCM_{EFF}$</td>
<td>Provides declaration of advices</td>
</tr>
<tr>
<td>$CK_{EFF}$</td>
<td>Relates features to advices</td>
</tr>
<tr>
<td>$PC_{EFF}$</td>
<td>Triggers advice selection</td>
</tr>
<tr>
<td>$UCM_{MOD}$</td>
<td>Advices</td>
</tr>
<tr>
<td>$CK_{MOD}$</td>
<td>Each configuration item</td>
</tr>
<tr>
<td>$PC_{MOD}$</td>
<td>Each selected feature</td>
</tr>
</tbody>
</table>
evaluateAdvice name spl product =
    evaluateAdvice' product advice
where
    advice = head [a | a ← advices spl, (advName a) == name]
evaluateAdvice' product advice =
    if (isBefore advice)
        then composeBefore product matchedSteps flow
    else composeAfter product matchedSteps flow
where
    matchedSteps = match (product (pointCut advice))
    flow = scenarioOfAdvice (advice)
But wait, don’t get scared! We also detail MSVCM pragmatics and provide tool support for it.
Writing MSVCM assets from a family of specifications.
Hephaestus

Rodrigo Bonifácio, Paulo Borba, Cristiano Ferraz

Modeling SPL Variability in Use Case Scenarios
Thesis statement
MSVCM is effective for reducing the coupling among feature specifications and increasing scenario cohesion. Altogether, MSVCM might simplify and reduce the effort to evolve SPL specifications.
Research questions

Q1. Is it worth to specify scenario variability using MSVCM?
   1. Does MSVCM reduce scattering and tangling?
   2. Does MSVCM better support the open-closed principle?
   3. Does MSVCM reduce the size of SPL specifications?

Q2. Does MSVCM increase the time to extract SPL specifications?

Q3. Does MSVCM reduce the time to evolve SPL specifications?
Evaluation
Hypothesis

H1. MSVCM reduces scattering and tangling
H2. MSVCM better supports the open-closed principle
H3. MSVCM reduces the total size of specifications
H4. MSVCM increases the number of assets
H5. MSVCM increases the average time to extract product lines
H6. MSVCM reduces the average time to evolve product lines
Modularity’s assessment (H1)

**Metrics** (adapted from the Marc Eaddy thesis)
- Degree of scattering of features
- Degree of tangling of scenarios

**Case studies**
- The security module of a Smart Home (SmartHome),
- Conference Management (EasyChair),
- Mobile Media (MobileMedia),
- Car Crisis Product Line (CarCrisis),
- eShop Product Line (eShop)
- E-finance (HomeBanking)
Degree of scattering

<table>
<thead>
<tr>
<th></th>
<th>MobileMedia</th>
<th>SmartHome</th>
<th>eShop</th>
</tr>
</thead>
<tbody>
<tr>
<td>CarCrisis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EasyChair</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HomeBanking</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MSVC</th>
<th>PLUSS</th>
<th>MSVC</th>
<th>PLUSS</th>
<th>MSVC</th>
<th>PLUSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoS</td>
<td>0.0</td>
<td>0.8</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Degree of tangling

Modeling SPL Variability in Use Case Scenarios
First conclusions

DoS benefits depends on...

- The *crosscutting nature* of features (homogeneous x heterogeneous)
- The more heterogeneous the features the less gain on scattering reduction

DoT benefits depends on

- The number of *alternative*, *or* and *optional* features
- The number of variants for each *alternative* and *or* feature
- Little tangling implies smaller gains on DoT improvements
Adherence to the open-closed principle (H2)

Metrics
We propose and theoretically validate\(^1\) the Impact of Change metric.

\[
IC = \frac{\#\text{modifications}}{\#\text{modifications} + \#\text{extensions}}
\]

Case studies
- EasyChair
- CarCrisis
- HomeBanking

---

\(^{1}\text{see Barbara Kitchenham and others (IEEE Trans. Softw. Eng. 1995).}\)
Impact of Change results

![Bar chart showing Impact of Change results for HomeBanking, CarCrisis, and EasyChair across MSVCM, PLUSS categories.](image)
Metrics

- Number of steps
- Vocabulary size (number of assets)

Case studies

- All six studies previously introduced.
Vocabulary size

<table>
<thead>
<tr>
<th>SPL</th>
<th>MobileMedia</th>
<th>SmartHome</th>
<th>eShop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary Size</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **CarCrisis**
  - MSVCM: 10
  - PLUSS: 6

- **EasyChair**
  - MSVCM: 12
  - PLUSS: 8

- **HomeBanking**
  - MSVCM: 14
  - PLUSS: 10

- **Total**
  - MobileMedia: 26
  - SmartHome: 34
  - eShop: 22

- **PLUSS**
  - Total: 18
Controlled experiments to check H5 and H6

Three “phases” of experiments involving students

In each phase we investigated:
- Average time to extract SPL specifications
- Average time to evolve SPL specifications

Techniques: MSVCM and PLUSS
First phase—extractive situation

P1, P2, P3... PN in SPL

Rodrigo Bonifácio, Paulo Borba, Cristiano Ferraz

Modeling SPL Variability in Use Case Scenarios
Factors: 12 Students x 2 SPLs x 2 Techniques
Latin square design (6 replicas)

\[ Y_{ijlk} = \mu + \tau_i + \alpha_{li} + \beta_j + \gamma_k + \varepsilon_{ijkl} \]
Factors: 12 Students x 2 SPLs x 2 Techniques

Latin square design (6 replicas)

\[ Y_{lijk} = \mu + \tau_i + \tau \alpha_{li} + \beta_j + \gamma_k + \epsilon_{lijk} \]
Execution

- Training (two classes)
  - Introductory class about the techniques
  - Warm-up session
- Tool support: Microsoft Word
- Product lines: eShop and EasyChair
Testing hypothesis $H_{50}$.

There is no significant difference in the average time to extract a product line, by reason of the specification technique (PLUSS or MSVCM).

- $H_{50}$ was not rejected (p-value=0.0619),
- although it is suggestive that MSVCM requires more time to extract SPL specifications.
First phase—reactive situation

P1, P2, P3... PN in SPL

SPL

P(N+1)

SPL'

P1, P2, P3... PN, P(N+1) in SPL
Factors: 10 Students x 2 Sets of Change x 2 Techniques

Latin square design (5 replicas)

\[ Y_{ijk} = \mu + \tau_i + \tau\alpha_{ii} + \beta_j + \gamma_k + \varepsilon_{lijk} \]
Design

- Factors: 10 Students x 2 Sets of Change x 2 Techniques
- Latin square design (5 replicas)

\[ Y_{lijk} = \mu + \tau_i + \tau\alpha_{li} + \beta_j + \gamma_k + \varepsilon_{lijk} \]
- No additional training
- Tool support: Microsoft Word
- Two sets of six change requests
Testing hypothesis $H_{60}$.

There is no significant difference in the average time to evolve a product line, by reason of the specification technique (PLUSS or MSVCM).

- $H_{60}$ was not rejected (p-value=0.2071).
- There is no evidence that MSVCM reduces the time to evolve product lines.
These results motivated us to conduct new “rounds” of experiments.

Indeed, we should mitigate some threats:

- Insufficient training
- Lack of tool support for recording the effort to extract and evolve SPL specifications

and refine our hypothesis, considering the type of the change requests and the number of preparatory classes.
These results motivated us to conduct new “rounds” of experiments.

Indeed, we should mitigate some threats:

- Insufficient training
- Lack of tool support for recording the effort to extract and evolve SPL specifications

and refine our hypothesis, considering the type of the change requests and the number of preparatory classes.
The “Learning Process”

Second phase of experiments
Preparatory classes

- One introductory class
- One warm-up session during 4 classes (extractive situation)
- Presentation of the results (one class)
- One warm-up session regarding the evolutive situation
Subjects and tools

- 22 students involved in the extractive evaluation
- 20 students involved in the evolutive evaluation

A tool was developed in order to mitigate errors during the data collection activity.
Testing hypothesis $H_{50}$.

There is no significant difference in the time required to extract a product line, by reason of the specification technique (PLUSS or MSVCM).

- $H_{50}$ was rejected (p-value=0.0004)
- There is strong evidences that MSVCM requires more time to extract SPL specifications.
Testing the hypothesis $H_{60}$

There is no significant difference in the time required to evolve a product line, by reason of the specification technique (PLUSS or MSVCM).

- $H_{60}$ was not rejected (p-value=0.259)
- There is no evidence that MSVCM reduces the time to evolve product lines.

![Box plot showing comparison of time to evolve product lines for MSVCM and PLUSS techniques.]
Third and final phase of experiments
Preparatory classes

- One introductory class about the extractive situation
- One warm-up session during 5 classes (extractive situation)
- One introductory class about the evolutive situation
- One warm-up session during 2 classes (evolutive situation)
16 students involved in the extractive evaluation
16 students involved in the evolutive evaluation

Students use the same tool of the previous phase to specify the product lines and collect the time required to extract and evolve SPLs.
Testing hypothesis $H_{50}$.

There is no significant difference in the time required to extract a product line, by reason of the specification technique (PLUSS or MSVCM).

- $H_{50}$ was rejected (p-value=0.0009)
- Enforcing the evidence that MSVCM requires more time to extract SPL specifications.
There is no significant difference in the time required to evolve a product line, by reason of the specification technique (PLUSS or MSVCM).

- $H_{60}$ was not rejected (p-value=0.082)
- although it is suggestive that MSVCM requires less time to evolve SPL specifications.
Final remarks...
Summary of the contributions

- New constructs for representing variability in textual scenarios.
- Novel notation for representing configuration knowledge.
- Tools and Haskell libraries for variability management.
- Designs of experiments that could be reused in other studies.
Next steps

Improve our evaluation through a deeper analysis of our previous experiments (and perhaps conduct a new experiment).

Further research

- Feature and aspect interactions
- Mitigate the problem of fragile pointcuts
- New variability and composition mechanisms
Next steps

Improve our evaluation through a deeper analysis of our previous experiments (and perhaps conduct a new experiment).

Further research

- Feature and aspect interactions
- Mitigate the problem of fragile pointcuts
- New variability and composition mechanisms
Thanks!
Modeling SPL Variability in Use Case Scenarios
An Approach Based on Crosscutting Mechanisms

Rodrigo Bonifácio  Paulo Borba  Cristiano Ferraz

March 8, 2010
Metrics
### Degree of Scattering

\[
\text{DoS}(f) = 1 - \frac{|S| \sum_{s \in S} (\text{Conc}(f, s) - \frac{1}{|S|})^2}{|S| - 1}
\]

\[
\text{Conc}(f, s) = \frac{\text{number of steps in } s \text{ assigned to } f}{\text{number of steps assigned to } f}
\]

### Degree of Tangling

\[
\text{DoT}(s) = 1 - \frac{|F| \sum_{f \in F} (\text{Dedi}(s, f) - \frac{1}{|F|})^2}{|F| - 1}
\]

\[
\text{Dedi}(s, f) = \frac{\text{number of steps in } s \text{ assigned to } f}{\text{number of steps of all features in } s}
\]
Analysis Procedure
Given a model as:

\[ Y_{lijk} = \mu + \tau_l + \tau\alpha_{li} + \beta_j + \gamma_k + \varepsilon_{lijk} \]

- The error variance must be constant
- The model must be additive
- The model must present a normal distribution of residuals
The model must present a constant variance of errors

Plot *residuals x observed values*
The model must present a constant variance of errors

**Score Test for Non-Constant Error Variance**

- $H_0$: constant variance of errors
- $H_1$: $H_0$ is false

$(p_{value} = 0.62) \Rightarrow$ No evidence for rejecting $H_0$
Tukey’s test for non-additivity

- $H_0$: the model is additive
- $H_1$: $H_0$ is false

$(p_{value} = 0.50) \Rightarrow$ No evidence for rejecting $H_0$
The model must present a normal distribution of residuals

Quantil-Quantil normal

![Normal Q-Q Plot](image)