A New Family of Software Anti-Patterns: Linguistic Anti-Patterns

Venera Arnaoudova¹, Massimiliano Di Penta², Giuliano Antoniol¹, Yann-Gaël Guéhéneuc¹

¹ École Polytechnique de Montréal, Canada
² University of Sannio, Benevento, Italy
Families of Patterns/Anti-Patterns

- Patterns, e.g., Design

- Anti-Patterns, e.g., Software Development
Source code lexicon

i.e. vocabulary of your code

• What do we know?
  • We should strive for good quality lexicon

• How is quality of the lexicon measured?
  • English words, domain terms
  • Abbreviations, acronyms

[Takang et al., ’96], [Deissenbock and Pizka, ’05], [Lawrie et al., ’07], [Abebe et al., ’09]
What does it do?

• Consider method:

```java
public ??? setBreadth(Dimension target, int source){ ??? }
```

• What is its return type?

• What is its implementation?

**Expecting:**
```
public void setBreadth(Dimension target, int source){
    target.setBreadth(source);
}
```

**Receiving:**
```
public Dimension setBreadth(Dimension target, int source) {
    if (orientation == VERTICAL)
        return new Dimension(source, (int)target.getHeight());
    else return new Dimension((int)target.getWidth(), source);
}
```
Does more than it says

• Consider method:

```java
public ??? setBreadth(Dimension target, int source){ ??? }
```

• What is it’s return type?

• What is it’s implementation?

Expecting:

```java
public void setBreadth(Dimension target, int source){
    target.setBreadth(source);
}
```

Receiving:

```java
public Dimension setBreadth(Dimension target, int source) {
    if (orientation == VERTICAL)
        return new Dimension(source, (int)target.getHeight());
    else return new Dimension((int)target.getWidth(), source);
}
```
What does it do?

Consider method:

```java
public ??? isClassPathCorrect(wellKnownType, compUnitDecl){ ??? }
```

- What is it’s return type?
- What is it’s implementation?

**Expecting:**

```java
public boolean isClassPathCorrect(..){
    if(..){return true; else {return false;}
}
```

**Receiving:**

```java
public void isClassPathCorrect(wellKnownType, compUnitDecl) {
    referenceContext = compUnitDecl;
    this.handle(..);
}
```
Consider method:

```java
public ??? isClassPathCorrect(wellKnownType, compUnitDecl) {
  ???
}
```

- What is its return type?
- What is its implementation?

**Expecting:**

```java
public boolean isClassPathCorrect(..) {
  if(..) {return true; else {return false;}
}
```

**Receiving:**

```java
public void isClassPathCorrect(wellKnownType, compUnitDecl) {
  referenceContext = compUnitDecl;
  this.handle(..);
}
```

*Says more than it does*
What does it contain?

• Consider attribute:
  ```java
  private static ??? stats
  ```

• What is it’s declared type?

• What is it’s purpose?

Expecting:  ```java
private static int[] stats // some statistics..
``` 

Receiving:  ```java
private static boolean stats = true;
```
Says more than it contains

- Consider attribute:
  ```java
  private static ??? stats
  ```

- What is its declared type?

- What is its purpose?

Expecting:  private static int[] stats // some statistics..

Receiving:  private static boolean stats = true;
Linguistic Anti-Patterns

• LAs: Recurring poor practices in the naming, documentation and choice of identifiers in the implementation.

• Different families

  • One such family is related to inconsistencies: Inconsistency LAs
Inconsistency LAs

- Inconsistency between a program entity’s
  - documentation (i.e., comments)
  - name
  - behaviour (type, implementation)
Inconsistency LAs

Does more than it says

Contains more than it says

Does the opposite than it says

Says more than it does

Says more than it contains

Contains the opposite than it says

Dimension setBreadth(..)

void isClassPathCorrect(..)

ControlEnableState disable(..)

int[] isReached

boolean _stats

//.. default exclude pattern.. INCLUDE_NAME_DEFAULT

Others trust what you say!
Inconsistency LAs

Others trust what you say!

**Behaviour**

- **Does more than it says**
  - “Set” method returns
  - “Get” - more than an accessor
  - “Is” returns more than a Boolean
  - Expecting but not getting a single instance

- **Says more than it does**

- **Does the opposite**
Inconsistency LAs

Others trust what you say!

Behaviour

Does more than it says

Says more than it does

- Not implemented condition
- Validation method does not confirm
- “Get” method does not return
- Not answered question
- Transform method does not return
- Expecting but not getting a collection

Does the opposite
Inconsistency LAs

Others trust what you say!

Behaviour

- Does more than it says
- Says more than it does
- Does the opposite

Method name and return type are opposite
Method signature and comment are opposite
State

- **Contains more than it says**
  - Says one but contains many
  - Name suggests Boolean but type does not

- **Says more than it contains**
  - Says many but contains one

- **Contains the opposite**
  - Attribute name and type are opposite
  - Attribute signature and comment are opposite
Study

• Prototype detector: LAPD

• RQ: To what extent LAs exist?

• Context:

<table>
<thead>
<tr>
<th>System</th>
<th>Version</th>
<th>Methods</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgoUML</td>
<td>0.10.1</td>
<td>5 K</td>
<td>3 K</td>
</tr>
<tr>
<td></td>
<td>0.34</td>
<td>11 K</td>
<td>6 K</td>
</tr>
<tr>
<td>Cocoon</td>
<td>2.2.0</td>
<td>4 K</td>
<td>3 K</td>
</tr>
<tr>
<td>Eclipse</td>
<td>1.0</td>
<td>36 K</td>
<td>22 K</td>
</tr>
</tbody>
</table>
Inconsistency LAs

<table>
<thead>
<tr>
<th>Behaviour</th>
<th>Detected</th>
<th>wrt the population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does more than it says</td>
<td>194</td>
<td>0.35%</td>
</tr>
<tr>
<td>Says more than it does</td>
<td>1016</td>
<td>1.82%</td>
</tr>
<tr>
<td>Does the opposite than it says</td>
<td>288</td>
<td>0.52%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Detected</th>
<th>wrt the population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains more than it says</td>
<td>438</td>
<td>1.3%</td>
</tr>
<tr>
<td>Says more than it contains</td>
<td>302</td>
<td>0.89%</td>
</tr>
<tr>
<td>Contains the opposite than it says</td>
<td>24</td>
<td>0.07%</td>
</tr>
</tbody>
</table>

Do they exist?
### Inconsistency LAs

<table>
<thead>
<tr>
<th>Behaviour</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does more than it says</td>
<td>88%</td>
</tr>
<tr>
<td>Says more than it does</td>
<td>85%</td>
</tr>
<tr>
<td>Does the opposite than it says</td>
<td>12%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains more than it says</td>
<td>57%</td>
</tr>
<tr>
<td>Says more than it contains</td>
<td>75%</td>
</tr>
<tr>
<td>Contains the opposite than it says</td>
<td>13%</td>
</tr>
</tbody>
</table>
To summarize

- Defined Inconsistency LAs
- Prototype detection tool - LAPD
  - 72% precision
- Inconsistency LAs represent 5% of the studied systems
Why do we care?

• What can go wrong with LAs:
  • Useless time and effort spent to understand source code
  • Wrong assumptions
  • Being aware they exist is the first step...
The next step

- Opinion of developers
- Study the impact
- Solutions
Inconsistency LAs

**Behaviour**
- Does more than it says
- Says more than it does
- Does the opposite than it says

**State**
- Contains more than it says
- Says more than it contains
- Contains the opposite than it says

What do you think?