From Hindsight to Foresight: Enhancing Design Artifacts for Business Logic Flaw Discovery

Carmen Cheh\textsuperscript{1}, Nicholas Tay\textsuperscript{2}, and Binbin Chen\textsuperscript{2}

\textsuperscript{1}Advanced Digital Sciences Center (ADSC), Singapore
\textsuperscript{2}Singapore University of Technology and Design (SUTD), Singapore
Motivation

High cost for fixing security issues in later stages (source: NIST)

Realizing quality improvement through test driven development: results and experiences of four industrial teams

Nachiappan Nagappan · E. Michael Maximilien · Thirumalesh Bhat · Laurie Williams

(Microsoft)
Background: Use Case Scenario Testing

**Scenario: Registering an account**

*Given* I am a new user

*When* I want to register a new account

*And* I specify the name as “Saul Goodman”

*And* I specify the password as “Heisenberg”

*And* I confirm the password as “Heisenberg”

*Then* I should be notified that my account is created

**Test code**

```php
/* @When I want to register a new account */
public function iWantToRegisterNewAccount(): void
{
    $this->fillContent();
}

/* @When I specify the password as :password */
public function iSpecifyThePasswordAs(string $password = ''): void
{
    $this->content['password'] = $password;
}

/* @When I register this account */
public function iRegisterThisAccount(): void
{
    $this->client->request(
        'POST', '/api/v2/shop/customers', [], [],
        ['HTTP_ACCEPT' => 'application/ld+json', 'CONTENT_TYPE' => 'application/ld+json'],
        json_encode($this->content, JSON_THROW_ON_ERROR));
    $this->content = [];
}
```

1 scenario (1 passed)
7 steps (7 passed)
Our Vision: Create High-quality Misuse Case Scenarios

Use case scenario

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as “Saul Goodman”
And I specify the password as “Heisenberg”
And I confirm the password as “Heisenberg”
Then I should be notified that my account is created

Test results

1 scenario (1 failed)
7 steps (6 passed, 1 failed)

Misuse case scenarios

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as “%&#^@&”
And I specify the password as “Heisenberg”
And I confirm the password as “Heisenberg”
And I register this account
Then I should be notified that my account is created

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as “%&#^@&”
And I specify the password as “Heisenberg”
And I confirm the password as “Heisenberg”
And I register this account
Then I should be notified that my account is created

Scenario: Registering an account
Given I am a new user
When I want to register a new account
And I specify the name as “%&#^@&”
And I specify the password as “Heisenberg”
And I confirm the password as “Heisenberg”
And I register this account
Then I should be notified that my account is created

1 scenario (1 failed)
7 steps (the third step failed)
Design Constraints Can Help

Use case scenario

**Scenario: Registering an account**

**Given** I am a new user  
**When** I want to register a new account  
**And** I specify the name as “Saul Goodman”  
**And** I specify the password as “Heisenberg”  
**And** I confirm the password as “Heisenberg”  
**And** I register this account  
**Then** I should be notified that my account is created

Misuse case scenarios

**Scenario: Registering an account**

**Given** I am a new user  
**When** I want to register a new account  
**And** I specify the name as “%&#^@&”  
**And** I specify the password as “Heisenberg”  
**And** I confirm the password as “Heisenberg”  
**And** I register this account  
**Then** I should be notified that my account is created

Test results

**Scenario: Registering an account**

**Given** I am a new user  
**When** I want to register a new account  
**And** I specify the name as “%&#^@&”  
**And** I specify the password as “Heisenberg”  
**And** I confirm the password as “Heisenberg”  
**And** I register this account  
**Then** I should be notified that my account is created

1 scenario (1 failed)  
7 steps (the third step failed)

Design constraints

1. Only new users can register an account  
2. Name must not have any special character  
3. Password must follow company’s password policy  
4. Confirmed password must be the same as the specified password  
5. All fields must be filled in  
6. Name and password can be specified in any order
What Kind of Design Constraints?

Use case scenario

Scenario: Registering an account
(1) Given I am a (a): new user
(2) When I want to register a new account
(3) And I specify the name as (b): “Saul Goodman”
(4) And I specify the password as (c): “Heisenberg”
(5) And I confirm the password as (d): “Heisenberg”
(6) And I register this account
(7) Then I should be notified that my account is created

Data Constraints
Access
Only new users can register an account
Range
Name must not have any special character
Correlation
Confirmed password must be the same as the specified password

Action Constraints
Prerequisite
All fields must be filled in
Ordering
Name and password can be specified in any order
Repetition
A user cannot register the same account more than once
### From Design Constraints to Misuse Case Scenario

<table>
<thead>
<tr>
<th>Constraints</th>
<th>From Constraints to MUCS</th>
<th>Example MUCS Snippets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Constraints</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access</td>
<td>Replace input parameter with invalid value.</td>
<td><strong>Given</strong> I am an <strong>existing user</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I specify the name as “!#$@#$”</td>
</tr>
<tr>
<td>Range</td>
<td></td>
<td><strong>And</strong> I specify the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I confirm the password as “hackingthis”</td>
</tr>
<tr>
<td>Correlation</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Action Constraints</strong></td>
<td>Swap ordering of steps in a way that does not satisfy the constraint</td>
<td><strong>And</strong> I confirm the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I specify the password as “Heisenberg”</td>
</tr>
<tr>
<td>Prerequisite</td>
<td>Remove each prerequisite step</td>
<td><strong>And</strong> I specify the name as “Saul Goodman”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I specify the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I confirm the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I register this account</td>
</tr>
<tr>
<td>Repetition</td>
<td>Duplicate the group of steps</td>
<td><strong>When</strong> I want to register a new account</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I specify the name as “Saul Goodman”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I specify the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I confirm the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I register this account</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I want to register a new account</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I specify the name as “Saul Goodman”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I specify the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I confirm the password as “Heisenberg”</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>And</strong> I register this account</td>
</tr>
</tbody>
</table>
Our User Study: Objective & Approach

• Main research question: Can developers easily specify useful design constraints?

• Baseline: Developers will manually enumerate misuse case scenarios that they would like to test.

• Project used: Ecommerce open source project, Sylius, and file-sharing open source project, ownCloud
<table>
<thead>
<tr>
<th>Constraints</th>
<th>Examples specified by participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>Only administrators can change the tax rate, add a product, and create a new promotion</td>
</tr>
<tr>
<td>Range</td>
<td>Shipping amount, product price, tax rate, and coupons’ per customer usage limit should not be negative. Passwords should follow the format set by the application’s password policy. HTTP method for accessing remote files must be POST or GET</td>
</tr>
<tr>
<td>Correlation</td>
<td>New password should be the same as re-confirmed password during resetting password. Only pre-defined variants can be used when adding product to cart</td>
</tr>
<tr>
<td>Pre-requisite</td>
<td>User must login first before doing any action. User must add a product to cart to continue with checkout. User must select shipping method to continue with checkout</td>
</tr>
<tr>
<td>Ordering</td>
<td>Checkout steps must be performed in strict order from addressing to shipping, and then to payment. Administrator must log in first before performing administrative duties</td>
</tr>
<tr>
<td>Repetition</td>
<td>Following same reset password link twice should not allow user to change password. Using a promotion multiple times should result in usage being decreased by equal amount</td>
</tr>
</tbody>
</table>

User Study Results: Constraints Specified
Evaluation

- Ran misuse case scenarios on:
  - Old version of Sylius (2016)
  - Latest version of Sylius (as of the writing of the paper)

- Misuse case scenarios and results of running them have been released as artifacts\(^1\)

\(^1\) https://github.com/cheh2/ACSAC-Artifacts
Turning Hindsight into Foresight

- **Context:** A security issue was reported by someone. After the Sylius developers fix the flaws, they add a misuse case scenario which is used to validate that the flaws are mitigated.

- **Findings:** The generated misuse case scenarios are similar to those created by Sylius developers.
**Use case scenario**

@ui
Scenario: Adding a new shipping method
  Given I am logged in as an administrator
  When I want to create a new shipping method
  And I specify its code as “Fedex”
  And I specify its amount as 50
  And I add it
  Then I should be notified that it has been successfully created

**Range Constraint:**

**Amount > 0**

**Our generated misuse case scenario**

@ui
Scenario: Adding a new shipping method
  Given I am logged in as an administrator
  When I want to create a new shipping method
  And I specify its code as “Fedex”
  And I specify its amount as -50
  And I add it
  Then I should be notified that it has been successfully created

**Expected Result:**

FAIL

**Sylius misuse case scenario**

@ui
Scenario: Adding a new shipping method
  Given I am logged in as an administrator
  When I want to create a new shipping method
  And I specify its code as “Fedex”
  And I specify its amount as -50
  And I try to add it
  Then I should be notified that shipping cannot be below 0
  And shipping method code “Fedex” should not be added

**PASS**

**Expected Result:**

FAIL

**PASS**

**Expected Result:**

FAIL
Discovery of New Flaws

• Findings: 5 new flaws are discovered and brought to Sylius developers’ attention. Flaws are currently (being) fixed.

• Our results show that we generate a larger coverage of potential misuse case scenarios for a given constraint

Scenario: Adding a new shipping method

Given I am logged in as an administrator
When I want to create a new shipping method
And I specify its code as “Fedex”
And I specify its amount as -50
And I add it
Then I should be notified that it has been successfully created

Expected Result: FAIL

Observations
• No shipping method created
• No error message thrown
### Discovery of New Flaws: Examples

**Scenario:** Modify tax rate

*Given* the store has “US Sales tax” tax rate of 20%
*And* I am logged in as an administrator
*And* I want to modify the tax rate “US Sales tax”
*When* I want to specify its amount as “-16%”
*And* I save my changes
*Then* this tax rate amount should be changed

**Expected Result:** FAIL

**PASS**

**Scenario:** Purchase product

*Given* the store has a “US Sales tax” tax rate of -10%
*And* the store has a product “T-Shirt” at $100
*And* I am logged in as “John”
*When* I add product “T-Shirt” to the cart
*Then* my cart total should be $110

**Expected Result:** FAIL

**FAIL**

**Range Constraint:**
- Amount > 0%
- Tax rate > 0%
Discovery of New Flaws: Examples

Our generated misuse case scenario

@ui
Scenario: Completing shipping step for checkout
  Given the store has “Post” shipping method with “$10” fee
  And I have product “TShirt” in the cart
  And I specify the shipping address as “908 LA”
  When I select “Post” shipping
  And I complete the shipping step
  Then I should be on the payment step

PASS

Expected Result:
FAIL

Prerequisite Constraint:
User must select shipping method

Sylius misuse case scenario

@ui
Scenario: Completing shipping step for checkout
  Given the store has “Post” shipping method with “$10” fee
  And I have product “TShirt” in the cart
  And I specify the shipping address as “908 LA”
  When I do not select a shipping method
  And I try to complete the shipping step
  Then I should still be on the shipping step

FAIL

Before the flaw was fixed
After the flaw was fixed

@ui
Scenario: Completing shipping step for checkout
  Given the store has “Post” shipping method with “$10” fee
  And I have product “TShirt” in the cart
  And I specify the shipping address as “908 LA”
  When I select “Post” shipping
  And I complete the shipping step
  Then I should be on the payment step

PASS

Test code has not been completed yet
Discovery of New Flaws: Examples

Use case scenario

Scenario: Counting promotion usage
Given there is a promotion “Limit” limited to 2 usage
And there is a customer “A” that bought a “Tshirt”
And the customer used “Limit” promotion
And the customer cancelled this order
When the administrator browses promotions
Then the “Limit” promotion should be used 0 times

Repetition Constraint:
Buying a product using the promotion twice should reduce the promotion usage to 1

Expected Result:
FAIL

Promotion is not even applied to the order!
Missing promotion definition step:
And the promotion gives $10 discount to every order.

Our generated misuse case scenario

Scenario: Counting promotion usage
Given there is a promotion “Limit” limited to 2 usage
And there is a customer “A” that bought a “Tshirt”
And the customer used “Limit” promotion
And the customer cancelled this order
When the administrator browses promotions
Then the “Limit” promotion should be used 0 times

PASS

Expected Result:
FAIL
Challenges

• Making it easier for developers to specify constraints
  • Intuitive syntax vs. expressiveness

• Generating misuse case scenarios for data constraints
  • Fuzzing parameters in a “smart” way
  • Maintaining certain syntax to match regex expression in code annotation

• Specifying detailed expected results when constraints are violated
  • Failure of misuse case scenario can be due to improper formulation of scenario/test code (which is not a design flaw)
  • Failure of misuse case scenario can happen at different steps in scenario
Related Work

State-of-the-Practice Tools

- BDD-Security

State-of-the-Art Research

- Model-based security testing and Design-by-Contract are two applicable concepts that can be introduced early on in design phase

- Types of design documents or models used for creating tests
  - Formal language models [1], [2]
  - Graphical models [3], [4]
  - Misuse case specification [6]

References:


Our Framework

Our Program

DESIGN PHASE

Specify

New Inputs

Contraints

Expected results

Use case scenarios

Test code

Generate misuse case scenarios

Develop code

CODING PHASE

Test use case scenarios

Testing Framework

CURRENT SDLC

Parse Use Case Scenario

Generate Misuse Case Scenario

Expected results

Analyze Results

List of bugs

OUR APPROACH
This research/project is supported in part by the National Research Foundation, Singapore and Infocomm Media Development Authority, and in part by the SUTD Start-up Research Grant. The authors would like to thank Ms. Alice Kham and Mr. En Low for their contributions to performing the experimental evaluation.