

An Empirical Study on the Impact of Inconsistency Feedback during Model and Code Co-changing

15th European Conference on Modeling Foundations and Applications ECMFA 2019

Georgios Kanakis, Djamel Eddine Khelladi, Stefan Fischer, Michael Troels, Alexander Egyed
Johannes Kepler University, Linz, Austria

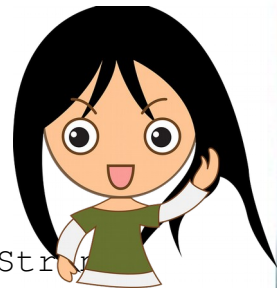
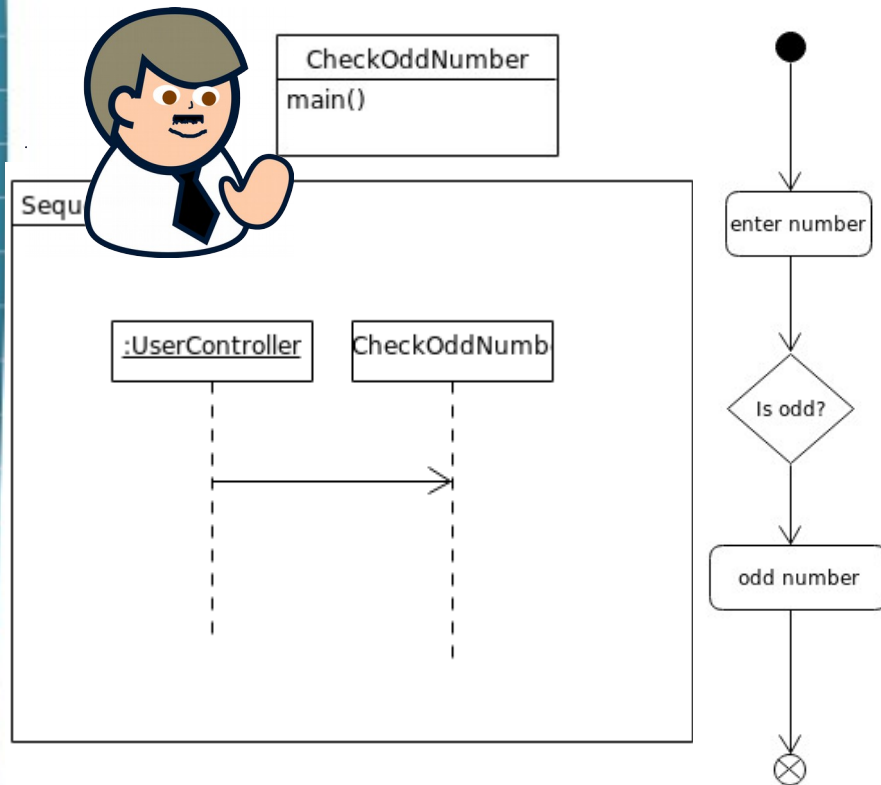
18 July 2019

The research leading to these results was funded by the Land of Upper Austria - Joint International PhD Program in Informatics under the grant number Wi-2014-202477/6, the JKU Linz Institute of Technology (LIT) the state of Upper Austria, grant no. LIT-2016-2-SEE-019 and Pro2Future, a COMET K1-Center of the Austrian Research Promotion Agency (FPG), grant no. 854184.

Introduction

Model Driven Development

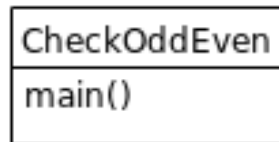
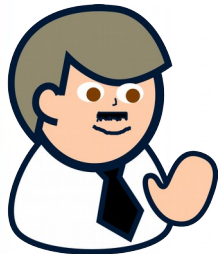
Engineers working with model and code using usually different tools



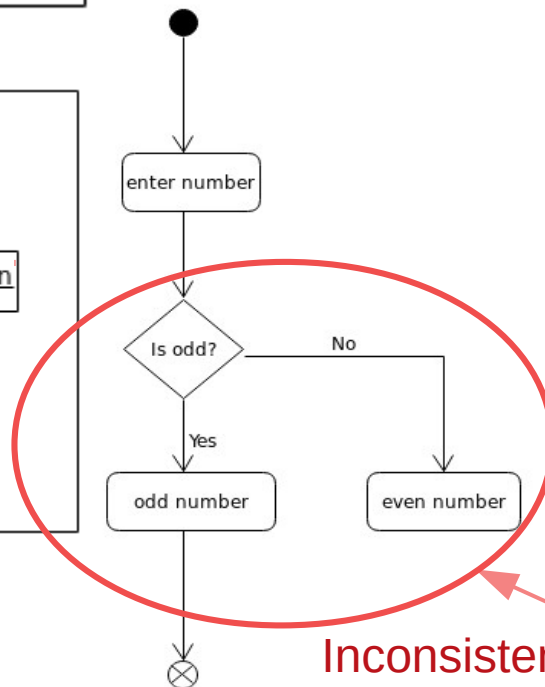
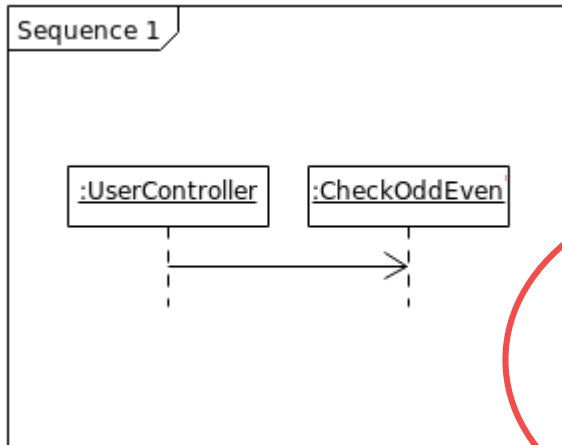
```
1 import java.util.Scanner;
2
3 class CheckOddNumber
4 {
5     public static void main(String
6     args[])
7     {
8         int num;
9         System.out.println("Enter an
10        Integer number:");
11
12        //The input provided by user is
13        stored in num
14        Scanner input = new
15        Scanner(System.in);
16        num = input.nextInt();
17
18        /* If number is divisible by 2
19        then it's an even number
20        * else odd number*/
21        if ( num % 2 != 0 )
22            System.out.println("Entered
23            number is odd");
24    }
25 }
```

Introduction

Co-evolution of model and code by the different engineers



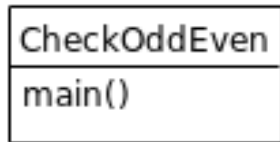
Inconsistency



Inconsistency

```
1 import java.util.Scanner;
2
3 class CheckOddNumber
4 {
5     public static void main(String
6     args[])
7     {
8         int num;
9         System.out.println("Enter an
10        Integer number:");
11
12        //The input provided by user is
13        stored in num
14        Scanner input = new
15        Scanner(System.in);
16        num = input.nextInt();
17
18        /* If number is divisible by 2
19        then it's an even number
20        * else odd number*/
21        if ( num % 2 == 0 )
22            System.out.println("Entered
23            number is even");
24        else if (num == 0)
25            System.out.println("Entered
26            number is odd");
27        else
28            System.out.println("Entered
29            number is odd");
30    }
31 }
```

Introduction



Inconsistency

3 `class` CheckOddNumber

Class Name does not exist in model

Is the inconsistency feedback beneficial?

`number is odd");`

State 0 number is not present in the model

Controlled Experiment

A yellow pencil and a pink eraser are positioned in the top right corner of the slide, appearing to be on a white sheet of paper against a blue grid background.

- 36 computer science students.
- A set of consistency rules between UML diagrams and Java code.
- 10 different tasks violating the consistency rules.
- 3 hour time.

Variables



- Independent variables

- Inconsistency feedback

- Provided

Class name does not match in code class

- Not provided

Class name does not match in code class

- Task complexity

- Simple tasks affecting 1 UML diagram Type
 - Complex tasks affecting more than 1 UML diagram Types

Variables

- Independent variables

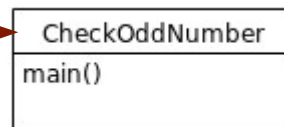
- Project

- Matador ≈5KLoC
 - Calendarium ≈21KLoC

- The direction of the change

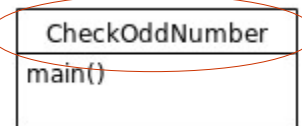
- Code to Model change propagation

```
1 import java.util.Scanner;  
2  
3 class CheckEvenOdd
```



- Model to Code change propagation

```
1 import java.util.Scanner;  
2  
3 class CheckEvenOdd
```



Variables



- Dependent variables

Time

The time in minutes a subject request to complete a task

Correctness


The correctness level of each completed task:

- I. Correct
- II. Partially correct
- III. Incorrect

Research Questions



- RQ1: “What is the effect of inconsistency feedback on correctness when engineers are presented with inconsistent models and code changes?”
- RQ2: “What is the effect of inconsistency feedback on time (effort)?”
- RQ3: “Does the complexity of the tasks affect time and correctness?”



RQ1: “What is the effect of inconsistency feedback on correctness when engineers are presented with inconsistent models and code changes?”


Research Question 1



Correctness Level	Inconsistency Feedback	No Inconsistency Feedback
Correct tasks	59 (42%)	22(16%)
Partially Correct	49(35%)	50(36%)
Incorrect	33(23%)	68(48%)

We had 226% increase at the correctly resolved inconsistent tasks when inconsistency feedback was provided

Research Question 1



Matador Project	Inconsistency Feedback	No Inconsistency Feedback
Correct Tasks	34(45%)	15(20%)
Partially Correct	18(24%)	24(32%)
Incorrect	24(31%)	36(48%)

Callendarium Project	Inconsistency Feedback	No Inconsistency Feedback
Correct Tasks	24(37%)	7(11%)
Partially Correct	26(40%)	26(40%)
Incorrect	15(23%)	32(49%)

Research Question 1

Direction	Correctness	Inconsistency Feedback	No Inconsistency Feedback
Model-to-code	Correct Tasks	40(47%)	14(23%)
	Partially Correct	30(35%)	23(38%)
	Incorrect	15(18%)	23(38%)
Code-to-model	Correct Tasks	19(34%)	8(10%)
	Partially Correct	19(34%)	27(34%)
	Incorrect	18(32%)	45(56%)

Subjects seems to benefit more when fixing Model-to-Code changes when inconsistency feedback is provided



RQ2: “What is the effect of inconsistency feedback on time (effort)?”

Research Question 2



No clear benefit in time was observed when inconsistency feedback was provided.

	Inconsistency Feedback	No Inconsistency Feedback
Correct & Partially Correct tasks	23 min	21 min
Correct Tasks	22 min	22 min
Incorrect tasks	N.A.	N.A.

Research Question 2

No clear benefit between the 2 projects



Project	Inconsistency Feedback	No Inconsistency Feedback
Matador	23 min	21 min
Callendarium	22 min	22 min

Project	Inconsistency Feedback	No Inconsistency Feedback
Model-to-Code	20 min	24 min
Code-to-Model	29 min	17 min



RQ3: “Does the complexity of the tasks affect time and correctness?”

Research Question 3

Split the correct tasks by complexity

Simple tasks

- Class diagram only
- Sequence diagram only
- State diagram only

Complex Tasks

- Class – Sequence diagrams
- Class – States diagrams
- Sequence – States diagrams
- Class – Sequence – State diagrams

Research Question 3

Affected UML Diagrams	Inconsistency Feedback	No Inconsistency Feedback
Simple Tasks		
Class	9	3
Sequence	12	6
State	7	3
Complex Tasks		
Class - Sequence	10	2
Class - State	3	2
Sequence - State	5	2
Class - Sequence - State	13	4

Research Question 3

Affected UML Diagrams	Inconsistency Feedback (Mean Time)	No Inconsistency Feedback (Mean Time)
Simple Tasks		
Class	23 min	18 min
Sequence	17.5 min	12.6 min
State	25 min	28 min
Complex Tasks		
Class - Sequence	22 min	68 min
Class - State	23 min	23.5 min
Sequence - State	23 min	15.5 min
Class - Sequence - State	21 min	20 min

Conclusion



- Subjects' correctness was significantly improved by 2.5x times when subjects were provided with inconsistency feedback.
- Subjects do not seem to have benefit when provided with feedback compared to subjects not provided with inconsistency feedback.
- Interestingly in the change direction variable, subjects have improved performance when provided with inconsistency feedback for the Model-to-Code direction.

Overall, the availability of inconsistency feedback provided support to the subjects in resolving correctly the inconsistencies between model and code.

Future Work



- Replicate the experiment
 - with experienced participants
 - Add complexity to the projects
 - Create project related inconsistencies
- More qualitative analysis on the subjects when solving these tasks must be made on the replicated experiment.

Questions??

