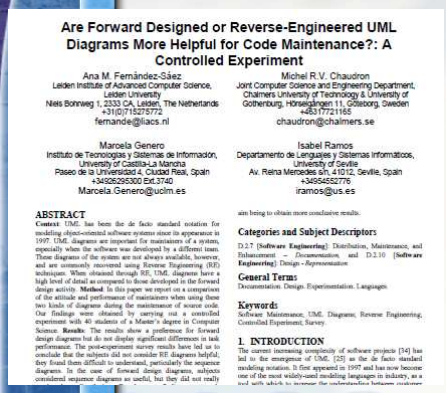
CACIC2013
XIX Congreso Argentino de Ciencias de la Computación

EASE 2013
17th International Conference on Evaluation and Assessment in Software Engineering
Porto de Galinhas, Brazil | April 14th - 16th, 2013

Are Forward Designed or Reverse-Engineered UML Diagrams More Helpful for Code Maintenance?: A Controlled Experiment

Ana M. Fernández-Sáez,
Marcela Genero, Michel R.V. Chaudron, Isabel Ramos

Fernández-Sáez, A., Chaudron, M., Genero, M., Ramos, I. (2013). Are forward designed or reverse-engineered UML diagrams more helpful for code maintenance?: a controlled experiment. EASE



Are Forward Designed or Reverse-Engineered UML Diagrams More Helpful for Code Maintenance?: A Controlled Experiment

Ana M. Fernández-Sáez
Leiden Institute of Advanced Computer Science, Leiden University
Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands
+31(0)715275722
fernande@liacs.nl

Michel R.V. Chaudron
Joint Computer Science and Engineering Department, Chalmers University of Technology & University of Gothenburg, Hörselgången 11, Göteborg, Sweden
+46317721165
chaudron@chalmers.se

Marcela Genero
Instituto de Tecnologías y Sistemas de Información, University of Castilla-La Mancha
Paseo de la Universidad 4, Ciudad Real, Spain
+34926295300 Ext.3740
Marcela.Genero@uclm.es

Isabel Ramos
Departamento de Lenguajes y Sistemas Informáticos, University of Seville
Av. Reina Mercedes s/n, 41012, Seville, Spain
+3495455276
iramos@us.es

ABSTRACT

aim being to obtain more conclusive results.

Categories and Subject Descriptors

General Terms

Keywords

1. INTRODUCTION

Material available at:
http://alarcos.esi.uclm.es/originUML maintenance/

EASE 2013
17th International Conference on Evaluation and Assessment in Software Engineering
Porto de Galinhas, Brazil | April 14th - 16th, 2013

Does the Origin of UML Diagrams Influence the Maintenance of the Code?: Results from a Controlled Experiment

Ana M. Fernández-Sáez, Marcela Genero, and Michel R.V. Chaudron

ABSTRACT

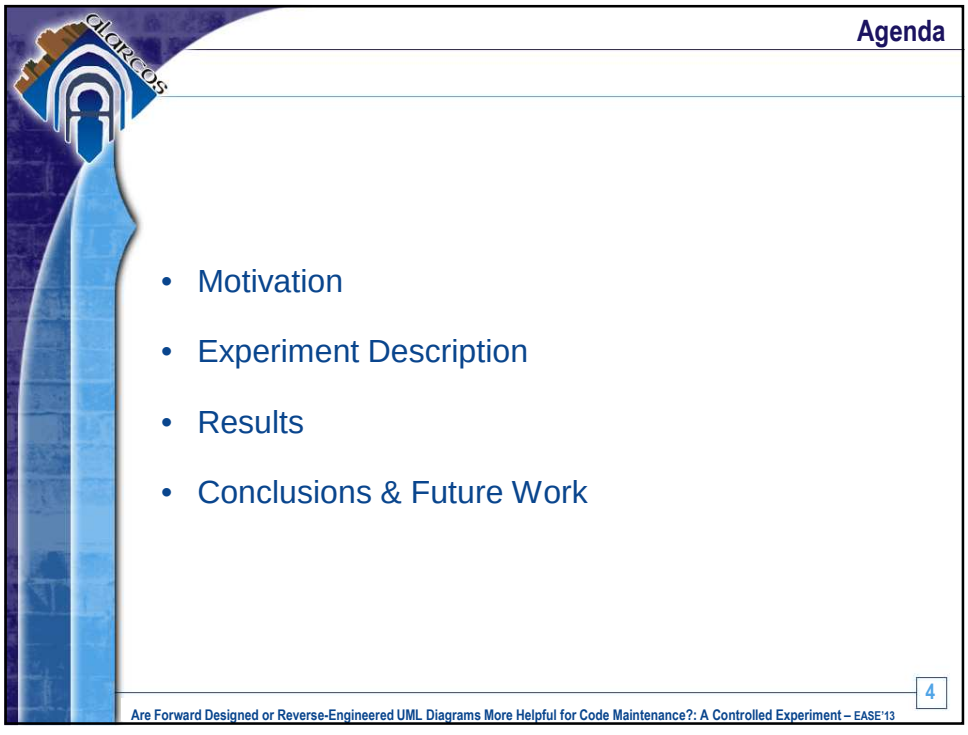
Background: UML diagrams have become an important technique with which to describe the functionality and behavior of a software system. These diagrams are particularly important for maintainers who have to maintain a software system which was developed by a different team or company. However, UML diagrams with the documentation of the software system are not always available, and have to be generated using a reverse engineering technique. UML diagrams (class diagrams, sequence diagrams) obtained through reverse engineering have a high level of detail in comparison to those developed in the design phase following a model-centric approach. Aim: We therefore wished to compare the performance of maintainers when using design UML diagrams as opposed to reverse engineered diagrams during the maintenance of source code. Method: This was achieved by carrying out a controlled experiment with 40 students in the second year of a Master's degree in Computer Science at the University of Seville (Spain). Results: The results obtained are not conclusive, but show a slight tendency towards obtaining better results when using UML diagrams obtained in the design phase. These results may encourage software developers to follow a model-centric approach, which implies beginning the development of a software system by building the corresponding UML diagrams and keeping them up-to-date in order to facilitate maintenance tasks.

Conclusions: Nevertheless, we are conscious that these results should be considered as preliminary. Further replications of this experiment are planned with students and professionals in order to obtain more conclusive results.

In the following table, you can find the experimental materials used in all replications.

Material in Spanish		Material in Italian	
TrainMat	RunMat	TrainMat	RunMat

TrainMat = Training Material; RunMat = Run Material

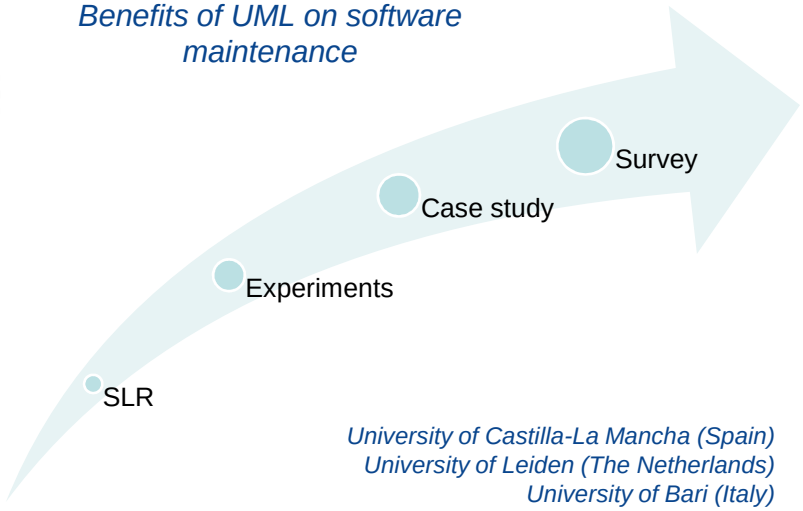


Agenda

- Motivation
- Experiment Description
- Results
- Conclusions & Future Work

Motivation

Benefits of UML on software maintenance



University of Castilla-La Mancha (Spain)
University of Leiden (The Netherlands)
University of Bari (Italy)

Motivation

- **Long-term study goal** → Investigate the benefits of using UML in software maintenance.
- **Current study goal** → Investigate if forward designed or reverse-engineered UML diagrams are more helpful for code maintenance.
  - *FD diagrams* → manually developed and maintained
  - *RE diagrams* → automatically obtained and not maintained

Impact of the results:

\+ *FD diagrams* → maintain the diagrams up-to-date

\+ *RE diagrams* → not to maintain the diagrams (saving time)

Motivation

Independent variable(s)
(factors-treatment)

Origin

(FD - RE diagrams)

Effect

Dependent variable(s)
(outcome)

Maintainability

Effectiveness
Efficiency

Other independent variables
(controlled)
(experience)

Experiment Description

Goal

- Using GQM template our goal is:
  - *Analyze* the origin of UML diagrams
  - *for the purpose of* evaluating them
  - *with respect to* their support for maintaining source code
  - *from the point of view* of the researcher
  - *in the context of* fifth year undergraduate students in Computer Science from the University of Seville.
- Used diagrams: sequence and class diagrams.
- Origin:
  - Forward Design diagrams → D
  - Reverse Engineered diagrams → RE

Experiment Description

Context selection

- 40 Computer Science students from the University of Seville (Spain) 2nd year of their Master's Degree.
- Realistic system
  - Sports center domain
  - In Spanish
  - Code in JAVA
- IBM Rational Software Architect

	#Class diagrams	#classes	#Sequence diagrams	#messages	LoC
D	4	16	21	226	5123
RE	4	21	11	191	

Experiment Description

Variables selection

- Independent variable: Origin
  - RE diagrams
  - D diagrams
- Dependent variable
  - Maintainability
- Measures
  - Maintainability Effectiveness (M_{Effec})

$$\frac{\#correct\ tasks\ -\ \#performed\ tasks}{\#tasks}$$

  - Maintainability Efficiency (M_{Effic})

$$\frac{\#correct\ tasks\ -\ \#performed\ tasks}{time\ spent}$$

- Cofactor
  - Ability

Experiment Description

Hypothesis formulation

$H_{1,0}$: *There is **no significant difference** in the subjects' maintenance effectiveness when working with UML diagrams which have originated from the design phase or with diagrams which originated from a Reverse Engineering technique.*

$H_{1,1}$: $\neg H_{1,0}$

$H_{2,0}$: *There is **no significant difference** in the subjects' maintenance efficiency when working with UML diagrams which have originated from the design phase or with those which originated from a Reverse Engineering technique.*

$H_{2,1}$: $\neg H_{2,0}$

Experiment Description

Experimental design

Between-subjects design

Origin	
RE	D
Group 1	Group 2

- *Selected design due to time restrictions*
- *Threats related to experience were alleviated through a pre- experiment questionnaire*

Object:

- Sport center software system (code + class and sequence diagrams) corresponding to the two treatments (FD, RE)

Tasks:

- **Pre-experiment tasks**
  - Background questionnaire,
  - Java
  - UML test

- Maintenance tasks
  - **5 Maintenance tasks → Randomized**
    - **3 Adaptive maintenance tasks**
    - **2 Corrective maintenance tasks**
  - **Answers sheets**

Write down the start time (HH:MM:SS) _____:_____:_____

You should add functionality for storing also the *phone number* of the customers of the system.

Note: You do NOT have to manage the phone numbers of the customers already stored in the database. This functionality is only for INTRODUCING NEW CUSTOMERS (they cannot be accessed, showed or modified). In your answer you may EXCLUDE changes to the user interface. If your solution requires changes to the database, you should explain these changes by writing comments on your answer sheet.

Write your answer on the forms

Write down the finish time (HH:MM:SS) _____:_____:_____

Experiment Description

Instrumentation

Internal validity: **Is it the treatment that really causes the effect?**

- Subjects with similar experiences
- Fatigue effect was avoided (time duration)
- Random assignment of subjects into groups

External validity: **Can the results be generalised?**

- Selection of objects and subjects

Construct validity: **Do the experimental settings actually reflect the construct under study?**

- Select well known domains
- Diagrams with similar size
- Select appropriate measures
- Take care with evaluation apprehension

Conclusion validity: **Are the results statistically valid?**

- Select the appropriate statistical tests according the design
- Check assumptions of the statistical tests

Experiment Description

Preparation

Pilot study

Why? Check the experimental material, the experiment duration.

When? One month before the experiment execution.

Who? 6 PhD students of the Alarcos research group at UCLM (Spain).

Experiment Description

Preparation/Excecution

1st session→ Training

- A training session of **two hours** took place the day before the experiment was carried out.
- The session included various tasks:
  - Collaborative exercise
  - Pre-experiment questionnaire
    - Background questionnaire, Java test, UML test

2nd session → Execution (2 hours)

- Assignment of subjects to the 2 groups located in the same room
- Explain experiment procedure (advice about time, tasks, etc.)
- Deliver the material, Receive the Material completed
- Deliver post-experiment questionnaire, Receive the questionnaire completed

Data analysis and interpratation

Descriptive statistics and exploratory analysis

Origin	N	MEffec			MEffic		
		$\bar{\chi}$	Median	SD	$\bar{\chi}$	Median	SD
RE	20	0.641	0.6818	0.165	0.00270	0. 00283	0.00079
D	20	**0.650**	0.6818	0.148	**0.00273**	0. 00303	0.00072

- Mann-Withney tests

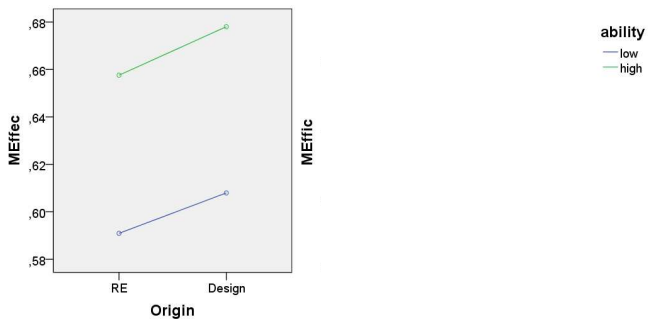
	MEffec			
	p-value	observed power	Effect size	R
Origin	0.957	0.054	0.001	NO

	MEffic			
	p-value	observed power	Effect size	R
Origin	0.534	0.051	0.0003	NO

Not significant

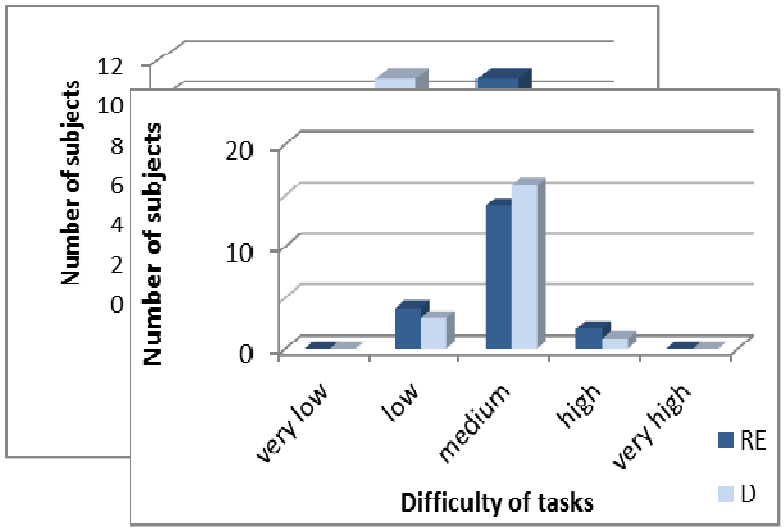
- Dividing by type of maintenance
  - Not significant

- Ability
  - p-value = 0.914 → not influenced the results
- Ability & Origin



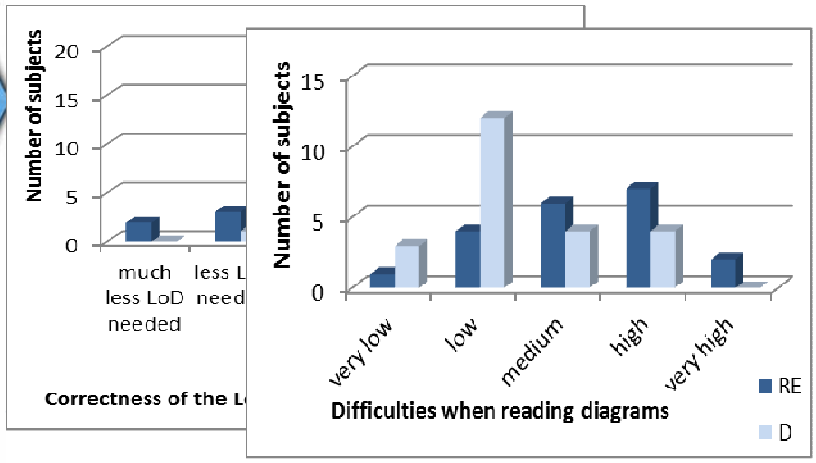
Data analysis and interpratation

Post-experiment survey



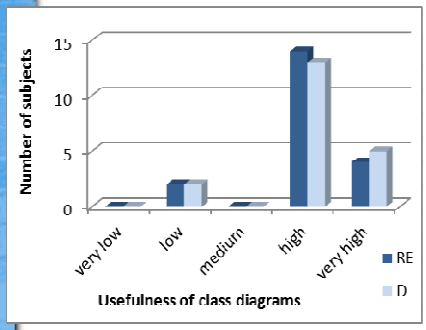
Data analysis and interpratation

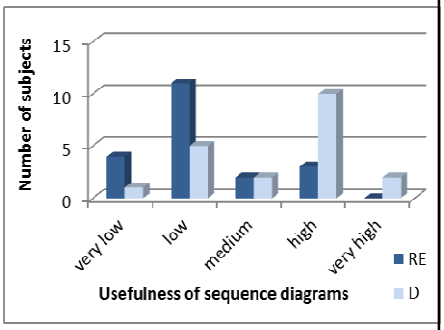
Post-experiment survey



Data analysis and interpratation

Post-experiment survey





Conclusions & Future Work

The results are not definitive

- *The null hypothesis cannot be rejected*
- *Descriptive statistics showed a slight tendency → FD diagrams*
- *Subjective preference for FD diagrams*

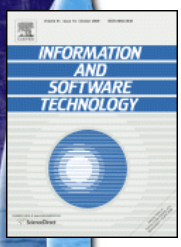
  Most used (class diagrams)

  Easier to read and understand

Results are valid in the context of simple systems related to well-known domains and novice software engineers

We carried out replications to obtain more definitive results (Seville and Bari)

Future Work (done)



Fernández-Sáez, A., Genero, M., Chaudron, M., Caivano, D., Ramos, I. (2013). **Are Forward Designed or Reverse-Engineered UML Diagrams More Helpful for Code Maintenance?: A Family of Experiments.**

Submitted to **Information and Software Technology** (Special section EASE 2013).

MAIN FINDINGS:
1) Subjects' performance is better when using FD diagrams.
2) Subjects preferred FD diagrams.
3) Subjects considered FD diagrams easier to understand.

Follow a model-based approach

CACIC2013
XIX Congreso Argentino de Ciencias de la Computación

EASE 2013
17th International Conference on Evaluation and Assessment in Software Engineering
Porto de Galinhas, Brazil | April 14th - 16th, 2013

Are Forward Designed or Reverse-Engineered UML Diagrams More Helpful for Code Maintenance?: A Controlled Experiment

Ana M. Fernández-Sáez,
Marcela Genero, Michel R.V. Chaudron,
Isabel Ramos