PhD in Information Technology and Electrical Engineering

Università degli Studi di Napoli Federico II

PhD Student: Vincenzo Riccio

XXXI Cycle

Training and Research Activities Report – Second Year

Tutor: Anna Rita Fasolino

Information

I am Vincenzo Riccio, I obtained a Ms.Sc. Degree (Laurea Magistrale) cum laude in Computer Engineering (Ingegneria Informatica) at the Università degli Studi di Napoli Federico II in April 2015. I am a Second Year PhD Student attending the XXXI Cycle of the Information Technology and Electrical Engineering (ITEE) PhD program of Università degli Studi di Napoli Federico II. My fellowship is financed by a PhD student grant. I am carrying out my research activity within the RevERSE research group of Software Engineering under the tutorship of Prof. Anna Rita Fasolino and collaborating with Domenico Amalfitano, Porfirio Tramontana, Nicola Amatucci, and Vincenzo De Simone.

Study and Training activities

In the second year of PhD program (1.11.2016-31.10.2017), I attended the PhD courses and carried out the training activities reported in tables below; these activities include Modules, Seminars, and a Workshop. I also attended two Doctoral Summer Schools at University of Salerno and Università degli Studi di Napoli Federico II, respectively.

Modules

Name	Type	Provided by	Credits
Formal Methods in Artificial Intelligence	Occasionally Provided	Prof. Aniello Murano	2,5
Le imprese e la ricerca: gestione strategica dell'informazione	Enhancement of the Research Skills	Prof. Francesco Bellucci	4
Progettazione e Sviluppo dei Sistemi Software	M.Sc. module	Prof.ssa Anna Rita Fasolino	9

Doctoral Schools

Name	Type	Provided by	Credits
13th International Summer School on Software Engineering (ISSSE 2017)	Doctoral School	Università di Salerno	3
13th Training And Research On Testing Summer School (TAROT 2017)	Doctoral School	Università degli Studi di Napoli "Federico II"	3

Seminars

Name	Type	Speaker	Credits
Verifica e Validazione di sistemi Safety Critical	Seminar	L. Porzio	0,4
Exploiting Machine Learning Techniques in Software Development Processes	Seminar	D. Amalfitano	0,2
IBM Cognitive Computing: Challenges and Opportunities in Building an Artificial Intelligence Platform for Business	Seminar	P. Leo	0,4
Cognitive Computing and da Vinci robot: Research Proposals and Discussion	Seminar	P. Maresca	0,2
DataFlow SuperComputing for BigData	Seminar	V. Milutinovic	0,6
Embedded Systems: Development and Testing in Automotive	Seminar	B. Noviello, C. D'Avino, A. Crispino, M. Rega	0,2
Mobile Memories	Seminar	C. Leonetti, M. Iaculo	0,2

Research Activity

In my PhD I am deepening my knowledge in the subject of Software Engineering. My main research topic is Software Testing, a practice that allows to evaluate and improve the software quality. In highly competitive sectors where software is a key component of the product, there is a constant pressure to improve the software quality and to provide quantitative evidence of the improvement. I focused on software testing in automotive, mobile and Internet of Things domains.

Model Driven Engineering

This research activity concerns the novel approaches in software development that focus on implementing models of software systems that are automatically transformed into target code. They are emerging as a relevant research topic in both industrial and scientific communities as they shift the focus of software development from writing code to modeling the behavior of the system [1]. Testing is a common practice to assess and improve software quality. In Model Driven approaches, models can be tested before the target code generation; this allows detecting earlier software faults and reducing the costs of software development processes.

In this second PhD year, I applied the knowledge gained during my Master Thesis in the Model Driven Verification and Validation of Embedded Software [C1] towards the Internet of Things domain.

In recent years the terms Smart Devices, Smartphones, Smart Homes, Smart Cities are increasingly being used. Advances in technology have made possible a smart vision of the world: a world overlaid with sensors, actuators and computing capabilities embedded in everyday objects always able to interact with each other in order to reach common goals. This context is often referred to as Internet of Things (IoT).

Although there is an increasing interest among several research communities to IoT themes, there are still few contributions related to approaches for the Verification and Validation of IoT systems. The testing of a Thing can be compared to the testing of an embedded system. An embedded control system makes decisions on the basis of the feedbacks it receives from the hardware under control, i.e. the plant. A Thing interacts analogously with its surrounding Context. This motivated me to collaborate on the preparation of the paper **"Towards a Thing-in-the-Loop approach for Verification and Validation of IoT systems"** accepted at the 1st ACM Workshop on the Internet of Safe Things (SafeThings 2017) [C2], co-located with the ACM Conference on Embedded Networked Sensor Systems (SenSys 2017). This manuscripts proposes a novel Verification and Validation approach for the IoT domain that inherits the advantages of the Model Driven approaches adopted in multiple embedded systems domains and tackles the main challenges of the IoT world relying on: (1) an abstract representation of the test cases and of the context of the entire IoT system; (2) the generation of concrete test cases for the specific languages, hardware platforms and communication protocols of the Thing to be tested, and their execution; (3) the simulation of the context in which the IoT system is immersed during the test execution.

Automated Testing Techniques for Android Applications

The major research activity of my PhD concerns Mobile Software Testing Automation, one of the main topics addressed by the RevERSE research group. Software testing is a well-known approach for assuring the quality of mobile applications.

Mobile applications present new challenges in contrast to other types of applications due to their peculiarities [2]. An Android app is composed by one or more Activities. Each Activity represents a single screen. The Android Framework defines a peculiar lifecycle for Activity instances in order to manage them transparently to the user who can navigate through an app and switch between apps without losing his progress and data. A systematic mapping study of mobile application testing techniques performed by Zein *et al.*[3] emphasized the need for specific testing techniques targeting Activity lifecycle conformance.

A relevant part of existing techniques and tools focus on testing the functionality of a system through its Graphical User Interface (GUI) [4]. These techniques are event-based but often neglect mobile-specific events [5]. Among the mobile-specific events, particularly relevant are the ones the exercise the lifecycle of the Activity app components, such as putting an application in background and resuming it or changing the orientation of the device. In fact, Android apps suffer from several issues that can be attributed to Activity lifecycle mishandling and affect their quality.

Android Testing emerged as a common research ground with Prof. Ana C. R. Paiva from the Informatics Engineering Department of the Faculty of Engineering of University of Porto (FEUP) who held the seminars *"Model based and Pattern based GUI Testing"* in November 2016, organized by Prof. Anna Rita Fasolino and Prof. Porfirio Tramontana as part of the ITEE PhD program. This meeting was an opportunity for the RevERSE research group to start working together with the Prof. Paiva's research group.

We focused on GUI failures, a relevant class of failures that may disrupt the mobile user experience. They consist in the manifestation of an unexpected GUI state [6]. Our efforts aimed at investigating GUI failures exposed in mobile apps by the mobile-specific event of changing the screen orientation. This event is able to exercise the Activity lifecycle.

We propose a classification framework that distinguishes three main classes of GUI failures due to orientation changes and exploit it in two studies that investigate the impact of such failures in Android apps. The studies involved both open-source and apps from Google Play that were specifically tested exposing them to orientation change events. The results showed that this problem is widespread in the context of Android mobile apps, some classes of GUI failures were more common than others, and some GUI objects were more frequently involved. The app source code analysis allowed us to identify classes of common faults causing specific GUI failures.

This study has been described in collaboration with D. Amalfitano, A. R. Fasolino, and A.C.R. Paiva in the paper **"Why does the orientation change mess up my Android application? From GUI failures to code faults"**, accepted by the Software Testing, Verification and Reliability (STVR) journal, published by Wiley [J1].

The results of this work motivated me to further investigate the issues that can be attributed to Activity lifecycle mishandling in Android apps. I supported the design and implementation of ALARic (Activity Lifecycle Android Ripper): a Fully Automated Black-box Testing Technique for Android Activities. This technique aims at detecting issues in Android apps that are tied to the Activity lifecycle. It is able to automatically explore the app under test, to systematically exercise the lifecycle of its Activities, and to detect both GUI failures and crashes. It has been defined by extending the generic online GUI testing algorithm described by the framework proposed in [7]. An Empirical Evaluation on 15 real Android apps showed that ALARic has been effective in detecting issues tied to the Activity lifecycle in all the analyzed apps. Moreover, ALARic outperformed Monkey, the state-of-the-practice tool, in

detecting issues tied to the Activity lifecycle**.** This work has been described in collaboration with D. Amalfitano, A. R. Fasolino, N. Amatucci, and V. De Simone in the paper **"Is This the Lifecycle We Really Want? An Automated Black-Box Testing Approach for Android Activities",** submitted to the 11th IEEE Conference on Software Testing, Validation and Verification (ICST 2018) [C3].

This activity is open to future works as (1) extension of the ALARic tool by implementing a set of oracles able to detect other issues tied to the Activity lifecycle, such as memory leaks and threading issues; (2) extension of the proposed approaches to test the lifecycle of other Android app components, such as Services, Fragments and Content Providers. Moreover, we could focus on designing fault localization techniques aimed at detecting source code bugs that may cause failures tied to the Activity Lifecycle.

Products

Conference Papers

[C1] D. Amalfitano, V. De Simone, A. R. Fasolino and V. Riccio, **Comparing Model Coverage and Code Coverage in Model Driven Testing: An Exploratory Study**, 2015 30th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW), Lincoln, NE, 2015, pp. 70-73. doi: 10.1109/ASEW.2015.1830.

[C2] D. Amalfitano, N. Amatucci, V. De Simone, V. Riccio, and A. R. Fasolino, **Towards a Thing-in-the-Loop approach for Verification and Validation of IoT systems,** 1st ACM Workshop on the Internet of Safe Things (SafeThings 2017), Delft, The Netherlands, 2017.

Journal Papers

[J1] Amalfitano D., Riccio V., Paiva ACR., Fasolino AR. **Why does the orientation change mess up my Android application? From GUI failures to code faults.** Softw Test Verif Reliab Journal , Wiley, 2017. doi: https://doi.org/10.1002/stvr.1654.

Submitted Papers

[C3] Amalfitano D., Amatucci N., De Simone V., Riccio V., Fasolino AR. **Is This the Lifecycle We Really Want? An Automated Black-Box Testing Approach for Android Activities.** Submitted to the 11th IEEE Conference on Software Testing, Validation and Verification (ICST 2018).

Conferences and Seminars

[S1] V. Riccio, “The Android Activity Lifecycle Testing: Overview and Challenges”, Internal Seminar for the RevERSE group.

[S2] V. Riccio, “Debug Your Android app”, Internal Seminar for the RevERSE group.

Activity abroad

I do not have carried out any activity abroad in my second PhD year

Tutorship

[T1] Attività di Tutorato e supporto a studenti iscritti al primo anno dei Corsi di Laurea incardinati nel Dipartimento di Ingegneria Elettrica e delle Tecnologie dell’Informazione (DIETI), Analisi I e Elementi di Informatica, 50 ore.

[T2] Presentazione dei Corsi di Studio e dei Laboratori agli studenti delle scuole superiori nell’ambito dell’Evento “Porte Aperte 2017”, 9 ore

[T3] Assistenza in Aula, Corso di Informatica per Non Vedenti “C. Savy”, 4 ore.

Credits Summary

Student: Name Surname									Tutor: Name Surname								Cycle XXXI		
vincenzo.riccio@unina.it									annarita.fasolino@unina.it										
	Credits year 1								Credits year 2								year 3		
		1	2	3	4	5	6			1	2	3	4	5	6				
	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Estimated	Total	Check
Modules	20	0	6	3	8	0	0	**17**	15	0	2,5	4	3	0	9	**18,5**	0	35,5	30-70
Seminars	10	6,8	1,4	0,7	0,8	0	0,5	**10,2**	5	0,4	0,8	0,6	3,4	0	0	**5,2**	0	15,4	10-30
Research	30	4	4	5	3	8	10	**34**	45	8	7	8	8	5	10	**46**	60	80	80-140
	60	10,8	11,4	8,7	11,8	8	10,5	**61,2**	65	8,4	10,3	12,6	14,4	5	19	**69,7**	60	130,9	180

References

[1] T. Meservy and K. Fenstermacher, “Transforming software development: an mda road map,” Computer, vol. 38, no. 9, pp. 52–58, Sept 2005.

[2] Muccini H, di Francesco A, Esposito P. Software testing of mobile applications: Challenges and future research directions. Automation of Software Test (AST), 2012 7th International Workshop on, IEEE: Zurich, Switzerland, 2012; 29–35, doi:10.1109/IWAST.2012.6228987.

[3] Zein S., Salleh N., Grundy J. A Systematic Mapping Study of Mobile Application Testing Techniques. J. Syst. Softw. Jul 2016; 117(C):334–356. https://doi.org/10.1016/j.jss.2016.03.065

[4] Issa A, Sillito J, Garousi V. Visual testing of graphical user interfaces: An exploratory study towards systematic definitions and approaches. Proceedings of the 2012 IEEE 14th International Symposium on Web Systems Evolution (WSE), WSE ’12, IEEE Computer Society: Washington, DC, USA, 2012; 11–15, doi:10.1109/WSE.2012.6320526.

[5] Zaeem RN, Prasad MR, Khurshid S. Automated generation of oracles for testing user-interaction features of mobile apps. Proceedings of the 2014 IEEE International Conference on Software Testing, Verification, and Validation, ICST ’14, IEEE Computer Society: Washington, DC, USA, 2014; 183–192, doi:10.1109/ICST.2014.31.

[6] Lelli V, Blouin A, Baudry B. Classifying and qualifying gui defects. 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST), 2015; 1–10, doi:10.1109/ICST.2015.7102582.

[7] Amalfitano D, Amatucci N, Memon AM, Tramontana P, Fasolino AR. A general framework for comparing automatic testing techniques of Android mobile apps. Journal of Systems and Software 2017; 125:322–343. doi:10.1016/j.jss.2016.12.017