Modelling and Transformation of Software Tests to Automate their Evolution. #PUBLIC 2017#

Dr Gerson Sunyé (director 40%), HDR, AtlanMod*els* team leader, LS2N laboratory
Dr Jean-Marie Mottu (supervisor 30%), AeLoS team, INRIA delegation in AtlanMod*els*, LS2N
Dr Pascal André (supervisor 30%), AeLoS team, LS2N

Context

When adopting Industry 4.0 technologies [Nan], industrial managers look for better productivity rates provided by more automation, more flexibility, and reactivity to changes. For example, the development of numerous delivery centers (e.g. Amazon, Chronodrive, Leclerc Drive, for individuals or professional, etc.). To ensure this transition, software quality must be improved by the adoption of reliable development methods, software verification, and testing. Concrete requirements focus on software upgrade bugs (e.g. the bug of the 1/09/2011 of Chronodrive[1]), the integration failures of new software components and services (e.g. the bug of Stallergenes[2] with medical consequences), which require adequate test procedures.

We consider continuously evolving software, where change occur at design time (agile processes) and all along the software lifecycle [LR00, MPP12] (by continuous improvements: new functions, extra-functional improvements, etc.) or in their ecosystem context [MD08] (when their dependencies or their deployment environments evolve). The verification of evolution is a continuous activity for the development and support teams.

Software Verification by testing is a difficult and strenuous activity when it is not sufficiently automated. The state of the art shows that existing test techniques are insufficient to capture different aspects such as: the scale and distribution of the system under test [AMS10], the code evolution or the amount of data to be processed, the uncertainty inherent to the CPS (Cyber-Physical Systems omnipresent in Industry 4.0). Typically, these test techniques rely on unit or integration tests that are dynamically executed on the application code. The tests are therefore sensitive to application evolutions. Since the evolution of tests is not currently automated, this requires expensive development effort.

In the AtlanModels and AeLoS teams, we study the software quality of different software systems such as web or distributed applications in the cloud [AMS16], as well as the development of mobile applications that can be deployed in a multitude of terminals, or connected objects (IoT) [partnership with BeApp], [KK13]. The quality of software as well as the preservation of this quality through their evolutions is a main concern for the transition to the Industry 4.0.

Problem

Decoupling tests from the deployed software is necessary for evolution [BDG07], but the main difficulty is to consider separately lazy-correlated concerns. In particular, the evolution of the software environment does not necessarily influence its functionalities: most of software tests should therefore be platform independent. Similarly, platform changes have an influence on some functionalities but mostly on non-functional concerns. Thus, it would be more efficient to have tests that consider only the functionalities of the software and others that separately consider their deployment in a given environment.

We summarize the problems to be dealt with during this thesis in two points:

---

[1]https://fr-fr.facebook.com/Chronodrive/posts/280088128671874}
[2]http://www.usinenouvelle.com/article/stallergenes-paralyse-par-un-probleme-informatique-en-france.N367862 - in french

1. Model the tests to simulate and transform them into a deployment platform such that the system code is generated [S06].
2. Make the tests evolve (semi-)automatically:
   a) Abstract the ecosystem evolutions at the model level
   b) Change the generation transformations to handle evolutions.

Opportunities

In the AtlanModels and AeLoS teams, we are interested in the Verification of Modeled Software thanks to Modeling and Model Transformation. Our experience results in numerous publications and various tools that will be a basis to begin the work of this thesis.

We study test formal modeling (Kmelia [AAA10, A16]) or semi-formal modeling (UML, MOF), as well as the evolutions to be considered (pb 1). The study of test simulation at the model level has already been undertaken (COSTOTest [AMA13, A16, AAM17]) (pb 1). Evolution can be taken into account by reverse engineering (MoDisco [BCJ10]) (pb 2.a). Capturing the changes in transformations can be considered by higher order transformations (ATL [JK06]) (pb 2.b).

Bibliography

[A16] "Costo Website." [Online]. Available: http://costo.univ-nantes. fr/, AeLoS team, major update January 2016.

[AAA10] P. André, G. Ardourel, C. Attiogbé, and A. Lanoix, "Using assertions to enhance the correctness of kmelia components and their assemblies", proceedings of FACS 2009, pp. 5 – 30, 2009.

[AAM17] Pascal André, Christian Attiogbé, and Jean-Marie Mottu. Combining techniques to verify service-based components. In Proceedings of AMARETTO@MODELSWARD 2017, Porto, Portugal. 19-21, 2017.

[AMA13] P. André, J.-M. Mottu, and G. Ardourel, "Building test harness from service-based component models," in proceedings of the Workshop MoDeVVa 2013 associated with Models2013 conference, Miami, USA, Oct. 2013, pp. 11–20.

[AMS10] Eduardo Almeida, João Eugenio Marynowski, Gerson Sunye, Yves Le Traon, and Patrick Valduriez. Efficient distributed test architectures for large-scale systems. In ICTSS 2010: 22nd IFIP Int. Conf. on Testing Software and Systems, Natal, Brazil, November 2010.

[AMS16] Albonico, M., Mottu, J.-M., Sunye, G., Controlling the Elasticity of Web Applications on Cloud Computing, Proceedings of the 31st ACM Symposium on Applied Computing, SAC2016, ACM.

[BCJ10] Bruneliere, H., Cabot, J., Jouault, F., & Madiot, F. (2010, September). MoDisco: a generic and extensible framework for model driven reverse engineering. In Proceedings of the IEEE/ACM international conference on Automated software engineering (pp. 173-174). ACM.

[BDG07] Baker, P., Dai, Z. R., Grabowski, J., Schieferdecker, I., & Williams, C. (2007). Model-driven testing: Using the UML testing profile. Springer Science & Business Media.

[JK06] Jouault, F., & Kurtev, I. (2006, January). Transforming models with ATL. In satellite events at the MoDELS 2005 Conference (pp. 128-138). Springer Berlin Heidelberg.

[KK13] KIRUBAKARAN, B. et KARTHIKEYANI, V. Mobile application testing—Challenges and solution approach through automation. In : Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on. IEEE, 2013. p. 79-84.

[LR00] Lehman, M. M., & Ramil, J. F. (2000). Software evolution in the age of component-based software engineering. IEE Proceedings-Software, 147(6), 249-255. ISO 690.

[MD08] Tom Mens, Serge Demeyer, Software Evolution, Springer, 347 p.

[MPP12] Mirzaaghaei, M., Pastore, F., & Pezze, M. (2012, April). Supporting test suite evolution through test case adaptation. In Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on (pp. 231-240). IEEE.

[Nan] Nantes. Nantes excellence trajectory. http ://next-isite.fr/

[S06] Douglas C. Schmidt. Guest editor's introduction: Model-driven engineering. IEEE Computer, 39(2):25{31, 2006.