The Role of models@run.time in Autonomic Systems: Keynote

Nelly Bencomo
ALICE, School of Engineering and Applied Science
Aston University
B4 7ET, Birmingham, UK
Email: nelly@acm.org

***Abstract*—Autonomic systems manage their own behaviour in accordance with high-level goals. This paper presents a brief outline of challenges related to Autonomic Computing due to uncertainty in the operational environments, and the role that models@run.time play in meeting them. We argue that the existing progress in Autonomic Computing can be further exploited with the support of runtime models. We briefly discuss our ideas related to the need to understand the extent to which the high-level goals of the autonomic system are being satisfied to support decision-making based on runtime evidence and, the need to support self-explanation.**

1. Background

As systems become more interconnected and diverse, software designers and architects are less able to anticipate and design interactions among components, and therefore they are not able to offer an *a priori* model to specify the system's dynamic behaviour and architecture. Such issues are left to be solved at runtime [1]. The expected result is that systems will become too large and complex for even the most skilled professionals to install, configure, optimize, and maintain. It will be virtually impossible to make timely, decisive responses to the rapid stream of conflicting and changing demands. Systems are increasingly expected to change themselves and self-react to continue to ensure their expected behaviour.

Autonomic Computing has emerged as the solution to the situation described above. An autonomic system is capable of self-management and able to monitor and analyze its runtime behaviour to make decisions on its own, and manage its behaviour according to high-level goals [2]. However, before end users and system administrators can take the benefits of autonomic computing for granted, researchers need to overcome different obstacles in designing them and understanding their behaviour.

2. Synergy between Autonomic Computing and models@run.time: experiences

Traditional software automation and adaptation techniques usually require an *a priori* model for a system's dynamic behaviour. Under the uncertainty present in current and future scenarios, this model is difficult to define and labour-intensive to maintain, and tends to get out of date due to architecture decay. Modern approaches, such as "models@runtime" [1], do not necessarily require defining the system's behaviour model beforehand. Instead, it can involve different techniques such as machine learning, or mining software component interactions from system execution traces to build a model which is in turn used to analyze, plan, and execute adaptations [3], and synthesize emergent software on the fly [4]. Autonomic Computing and runtime models can be used together to support the new paradigm needed to break the boundary between design time and runtime [5]. Models would not be just design artefacts but would continue to live and evolve at runtime according to changes while the system is running. Autonomic Computing can provide the intelligent support needed during runtime to update and evolve the runtime models.

Uncertainty will inevitably provoke emergent behaviour that is not expected as it has not been foreseen previously. A crucial issue to be tackled by an autonomic system is its ability to continuously quantify the deviation between the behaviour exposed and the behaviour expected, which is dictated by its high-level goals, based on collection and evaluation of new evidence [6]. The system would therefore be goal-aware. According to how large the deviation gap is, the autonomic system should decide to take corrective actions or, to flag that an abnormal situation is happening. Appraisal of new evidence by the running system will improve its judgement while performing decision-making.

In [6], [7], [8], we have proposed ways to use techniques such as Bayesian learning to collect runtime evidence, and therefore inform and update runtime models accordingly, enhancing the judgement and decision-making process of the system. We have used the concept of Bayesian surprise [9] to measure how observed data modify, during runtime, previous assumptions of the world. The notion of Bayesian surprise specifically measures the divergence between prior and posterior distributions given evidence observed, and which will be used to quantify the size of the gap between the behaviour targeted by the goals of the autonomic system and the behaviour exposed. The final result is a better-informed decision making by enabling the re-appraisal and

update of the runtime models according to evidence gathered from the operational environment. We have shown how new evidence could imply that design-time assumptions are not valid anymore. The role of Bayesian surprises is to offer support to flag these situations. Among others, the techniques we are developing will enable the autonomic system to (i) temporarily and autonomously "relax" requirements and face unanticipated but transient environmental conditions which could trigger unnecessary actions [10], (ii) disclose conflicts between non-functional requirements and support reasoning about these conflicts based on the new knowledge obtained during execution. The newly acquired knowledge, which may have been impossible to know before runtime, provides a better understanding of the operating environment by the running system [6].

Another well-known problem with autonomic systems is that users may not understand them due to the emergent behaviour. The difficulty in predicting the system's behaviour means that the system may surprise its customers and/or developers. Such a lack of understanding compromises the trust by end users and can end in situations where they cease to use a system [11]. Because its behaviour is emergent, an autonomic system needs to promote confidence in its end users and it needs to resolve any surprise. The latter can only be attained if an autonomic system is also capable of self-explanation [12]. In the context of goal-awareness, the technique for quantifying the gap is based on a runtime goal model and qualitative and quantitative reasoning about how the organisation of the goal-based model changes over time [13] and its impact on the architecture of the system and viceversa. Additional research challenges worth exploring includes how best to present explanations, which essentially consist of a trace of system behaviour, in our case a sequence of operations applied to the runtime goal models and/or architecture models [14]. We need to incorporate techniques such as artificial intelligence, machine learning, optimization, planning, decision theory, emergent behaviour analysis, and bio-inspired computing into our systems while retaining the ability to reason about system behaviour and provide explanation with respect to the goals and requirements of the system. To produce these required techniques, there are plenty of lessons to learn from research communities such as ICAC and models@run.time.

3. Concluding Remarks

This short paper highlights synergies between autonomic systems and models@run.time.This short paper is also a call for more mutual awareness and recognition. The communities behind runtime models, software engineering, and autonomic systems should work in a cooperative way to further investigate means for conceiving autonomic software systems that users trust and understand. How do we put in contact all these communities to work together towards a common goal? This is a big challenge!

Acknowledgments

The author would like to thank Amel Belaggoun and Luis Garcia-Paucar as the postgraduate students supporting the research described in this paper.

References

[1] G. Blair, N. Bencomo, and R. B. France, "Models@ run. time," *Computer*, vol. 42, no. 10, pp. 22–27, 2009.

[2] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan 2003.

[3] N. Esfahani, E. Yuan, K. R. Canavera, and S. Malek, "Inferring software component interaction dependencies for adaptation support," pp. 26:1–26:32, 2016.

[4] N. Bencomo, A. Bennaceur, P. Grace, G. Blair, and V. Issarny, "The role of models@run.time in supporting on-the-fly interoperability," *Computing*, vol. 95, no. 3, pp. 167–190, 2012.

[5] L. Baresi and C. Ghezzi, "The disappearing boundary between development-time and run-time," in *Proceedings of the Workshop on Future of Software Engineering Research, FoSER 2010, at the 18th International Symposium on Foundations of Software Engineering, 2010, USA*, 2010, pp. 17–22.

[6] N. Bencomo, "Quantun: Quantification of uncertainty for the reassessment of requirements," in *23rd IEEE International Requirements Engineering Conference, RE*, 2015, pp. 236–240.

[7] N. Bencomo, A. Belaggoun, and V. Issarny, "Dynamic decision networks to support decision-making for self-adaptive systems," in *(SEAMS)*, 2013.

[8] L. Garcia-Paucar, N. Bencomo, and K. Yuen, "Juggling preferences in a world of uncertainty," in *Proceedings of the Reqirements Engineering Conference 2017, RE-NEXT Track*, 2017.

[9] N. Bencomo and A. Belaggoun, "A world full of surprises: bayesian theory of surprise to quantify degrees of uncertainty," in *ICSE*, 2014, pp. 460–463.

[10] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J. M. Bruel, "RELAX: A language to address uncertainty in self-adaptive systems requirement," *Requirements Engineering*, vol. 15, no. 2, pp. 177–196, 2010.

[11] B. Muir, "Trust in automation: Part i," *Theoretical Issues in the Study of Trust and Human Intervention in Automated Systems. Ergonomics*, vol. 37, no. 11, pp. 1905–1922, 1994.

[12] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. Finkelstein, "Requirements-aware systems: A research agenda for RE for self-adaptive systems," in *Proceedings of RE*, ser. RE '10. Washington, DC, USA: IEEE, 2010.

[13] K. Welsh, P. Sawyer, and N. Bencomo, "Run-time resolution of uncertainty," in *RE 2011, 19th IEEE International Requirements Engineering Conference, Trento, Italy, August 29 2011 - September 2, 2011*, 2011, C, pp. 355–356.

[14] N. Bencomo, K. Welsh, P. Sawyer, and J. Whittle, "Self-explanation in adaptive systems," *Transactions on Computational Collective Intelligence*, 2014.