Research Statement

Nelly Bencomo - www.nellybencomo.me

Keywords: distributed self-adaptive systems, software engineering, models@run.time

Modern software systems are highly distributed and software engineering (SE) is inherently interdisciplinary, comprising both technical and human concerns. SE provides me with my preferred style of research, which is **collaborative** and **interdisciplinary**. I seek to apply SE on real-world applications and that has taken me to work in research labs that are not inherently related to SE, such as the ARLES team in Inria- France (a lab on software architecture and distributed systems in French), the reflective middleware group in Lancaster University, a math models Lab at the UCV in Venezuela and more recently the research lab, which I co-founded, ALICE at Aston (the Aston Lab for Intelligent Collectives Engineering). I have built a portfolio of successful collaborative and innovative research in areas such as decision making under uncertainty and distributed, self-adaptive and autonomous systems where I have successfully applied innovative SE techniques.

Current Work: Decision-making Under Uncertainty

As software systems become more pervasive and mobile, there is growing **uncertainty** about their environment. Therefore, how the system should behave under different contexts cannot be fully predicted at design-time [23; 10]. It is considerations such as these that have led to the development of self-adaptive systems (SAS) [25], which have the ability to dynamically and autonomously reconfigure their behavior to respond to changing external conditions.

The self-adaptation and self-management capabilities required involve decision making under uncertainty - that is, choosing actions based on often imperfect observations, with unknown outcomes. Designers of these distributed and self-adaptive and software systems must take into account the various sources of uncertainty while balancing the multiple objectives of the system [24; 11]. A key argument in my research is that current systems engineering methods do not well support the kind of dynamic appraisal of how well the SAS is meeting its target behaviour. Furthermore, a considerable part of current research efforts in the research area of SAS have focused on answering the question *when to adapt* by defining the events (known or partially known) and the actions the system needs to take accordingly in order to perform the adaptation [25]. The approach I have taken is rather different. I argue that **a dynamically adaptive system should be able to autonomously assess deviations from its specified behaviour and use these deviations to trigger adaptation**. This is a different way to tackle the question *when to adapt* and allows the running system to deal with uncertainty in a more explicit way. The underlying hypothesis here is that explicit treatment of uncertainty by the running system, based on evaluation of evidence, improves its judgment to make decisions. Crucially, allowing the system to exhibit explicit treatment of uncertainty by evaluation of new evidence requires the use of techniques that traditionally have not been used to support requirements, such as Bayesian learning and other AI techniques [21; 14; 15].

QuantUn: Quantification of Uncertainty. I am working on the development of novel techniques to explicitly quantify uncertainty to support decision-making in self-adaptive systems. **An aim of *QuantUn* is to investigate techniques to explicitly quantify the deviation from the original specified behaviour based on information and evidence gathered during runtime**. A technique I have developed is based on the novel idea of the definition of Bayesian surprise [1] as the basis for quantitative analysis to measure degrees of uncertainty and deviations of self- adaptive systems from the expected behaviour. A surprise measures how observed data affects the models or assumptions of the world during runtime. The key idea is that a "surprising" event can be defined as one that causes a large divergence between the belief distributions prior to and posterior to the event occurring. Crucially, a surprise implies that the system may decide either to adapt according to the new situation found or to flag that an abnormal situation is happening, and therefore the need for adaptation may be delayed. Further, surprises associated with different non-functional requirements (a.k.a. QoS properties) can be used to uncover unknown relationships between non-functional requirements to, therefore, allow the re-assessment of their tradeoff due to new knowledge acquired during runtime and that may have been impossible to know before.

In [12] with my collaborators in Birmingham University, I presented a method that allows designers to make explicit links between the possible emergence of surprises, risks and design trade-offs during design time. The method can be used to explore the design decisions to support self-adaptation to, therefore, choose among decisions that satisfice non-functional requirements in a better way and also address their trade-offs.

Future Research

I plan to leverage my developed research skills to build and enhance my current portfolio with new, but allied and complementary, lines of research, potentially collaborating to solve real-world problems relevant to society. Models@run.time is a crucial device in the future research directions.

Uncertainty. I plan to continue working on decision-making under uncertainty. One of the benefits of being able to characterise if a surprise is small is that the surprises could be used as a way of providing an implementation of the RELAX language [10]. Furthermore, small surprises could be used to tolerate evidence of unanticipated but transient environmental conditions that can trigger unnecessary adaptations. Doing so would be an alternative solution to the one offered in [7] where the validity of design assumptions are assessed at run time while tolerating minor and unanticipated environmental conditions that can trigger adaptations. I am also interested in the definition of new quantification techniques alternative to Bayesian surprises.

Requirements Reflection. Requirements reflection is a new SE paradigm (a.k.a. requirements-awareness) [11; 17], in which requirements are reified as runtime entities [8]. This allows systems to dynamically reason about themselves at the level of the requirements. Requirements reflection supports the development of SAS because it raises the level of discourse at which a software system is able to reflect upon itself [9]. With other colleagues [11], I have presented a research agenda to realise the vision of requirements reflection to support the development and operation of self-adaptive systems. Five challenges for this vision were identified. During my Marie Curie fellowship I worked towards all these research challenges with main results towards challenges 1, 2 and 4. Challenge 1, Uncertainty, came up as my main current research topic developed earlier. Bellow, I list the challenges and make reference to the publications associated. However, Challenges 3, 4, and 5 offer relevant paths that can be built upon my current research and with collaboration with academics in Lancaster.
-Challenge 1: Dealing with Uncertainty. Uncertainties arise because of the stochastic nature of events in the environment, limited sensor capabilities, and difficulties in predicting how the modification of system services will affect behaviors and the system goals. [5; 6; 7]
-Challenge 2: Runtime representation of requirements. The first challenge is the runtime representation of requirements in a form suitable for introspection and adaptation. Introspection implies the ability of a runtime entity to reveal information about itself. Explicit runtime representation of systems' requirements and goals has a key role when endowing systems with self-awareness capabilities [9; 8; 15].
-Challenge 3 : Multi-objective decision-making. Because of the nature of conflicting requirements, run-time resolutions of uncertainty inherently involve multi-objective decision making. In SE, multi-objective decision making techniques most often rely on constructing a utility function, defined as the weighted sum of the different objectives. However, this approach suffers from a number of drawbacks. Firstly, it is well known that correctly identifying the weight of each goal is a major difficulty. Secondly, the approach hides conflicts between multiple goals under a single aggregate objective function rather than truly exposing the conflicts and reasoning about them.
-Challenge 4: Self-explanation. A problem with self-adaptive systems is that users may not understand or trust them. Such a lack of intelligibility can mean that users may cease to use a system. I propose the use of goal-based models during runtime to offer self-explanation of how a system is meeting its requirements, and why the means of meeting these were chosen. [2; 3]
-Challenge 5: Synchronization between goals and architecture. The above has the subsequent need to maintain the relationship (i.e. synchronization) at runtime between goals and underlying system structures. I therefore see a major challenge to maintain this synchronization during execution as either the requirements are changed from above or the architecture is changed from below. [16]

Wrt *Challenge 3*, Multi-objective decision-making, I believe that Bayesian surprises can also be used to explore the operating environment to therefore improve its understanding. we have envisioned [1] the use of Bayesian surprises to support a review process of sensitivity analysis to agree on consistent utility functions. As hinted by the initial experiments shown in [1], preferences and weights to certain QoS properties given by experts during the sensitivity analysis process may not be ideal for some specific cases. Badly-chosen preferences and weights can either suggest unnecessary adaptations or make the system miss adaptations that may degrade the behaviour of the system due to contexts that were not fully understood during the requirements elicitation and design of the decision-making process. Furthermore, Bayesian surprises do not make use of preferences or weights. Based on the above and using the application scenarios, Bayesian surprises will be used (i) as a way to review and improve the sensitivity analysis to agree on consistent utility functions during simulations of the system or while having the running system enabled with Bayesian surprise technique, and (ii) to uncover conflicts between non-functional requirements and support reasoning about these conflicts and therefore, allow the re-appraisal of their tradeoff due to evidence found during runtime. In [13] with my PhD student at Aston we show first experiments using techniques to reassess the preferences of quality attributes (soft-goals or non-functional requirements) to correct unwanted effects due to initial preferences and assumptions that may not agree with the current environmental conditions and context monitored by the system.

Challenge 4: Self-explanation. The behaviour of distributed and self adaptive systems can be emergent, which means that the system's behaviour may be seen as unexpected by its customers and its developers. Therefore, a distributed self-adaptive system needs to garner confidence in its customers and it also needs to resolve any surprise on the part of the developer during testing and maintenance. I believe that these two functions can only be achieved if the system is also capable of self-explanation. I argue a self-adaptive system?s behaviour needs to be explained in terms of satisfaction of its requirements to be understood by stake holders such as final users or operators. Since new expected behaviour may emerge, I propose the use of runtime requirements models to offer self-explanation of how a system is meeting its new target behaviour. Also, it is needed to raise the level of abstraction while developing distributed self-adaptive applications to support reasoning and match domain-specific knowledge. In [2] we present initial results and demonstrate the analysis of run-time requirements models to yield a self-explanation codified in a domain specific language.

Challenge 5: Synchronisation between goals and architecture. To support eternal software systems, the focus of software development should shift away from a traditional approach where environmental conditions are foreseen and behaviour of the system is coded accordingly. New dynamic approaches are required where components and/or services are dynamically discovered and then composed together to recreate the system according to the current requirements and environmental contexts. This dynamic composition requires the synthesis of software on-the fly [16]. I believe that the use of runtime models will play an important role [4].

I am attracted to apply the above to a range of distributed systems that fall within different areas of interest, such as IoT, cloud computing. I am also attracted by problem domains such as environmental IoT deployments, mitigation and adaptation to environmental change, sustainability.

References

[1] N. Bencomo and A. Belaggoun. A world full of surprises: bayesian theory of surprise to quantify degrees of uncertainty. In *ICSE Companion*, pages 460–463, 2014.

[2] K. Welsh, N. Bencomo, P. Sawyer, and J. Whittle. Self-explanation in adaptive systems based on runtime goal-based models. In *Transactions on Computational Collective Intelligence*, december 2014.

[3] N. Bencomo, K. Welsh, P. Sawyer, and J. Whittle. Self-explanation in Adaptive Systems Assumptions. In *17th IEEE International Conference International Conference on Engineering of Complex Computer Systems ICECCS 2012*, Paris, France, 2012.

[4] H. Giese, N. Bencomo, L. Pasquale, A. Ramirez, P. Inverardi, S. Watzoldt, and S. Clarke. Living with uncertainty in the age of runtime models. *Models@run.time*, *LNCS*, Springer International Publishing, 2014.

[5] N. Bencomo, A. Belaggoun, and V. Issarny. Dynamic decision networks to support decision-making for self-adaptive systems. In *SEAMS*, 2013.

[6] N. Bencomo and A. Belaggoun. Supporting decision-making for self-adaptive systems: From goal models to dynamic decision networks. In *REFSQ - Best Paper Award*, 2013.

[7] A. Ramirez, B. Cheng, N. Bencomo, and P. Sawyer. Relaxing claims: Coping with uncertainty while evaluating assumptions at run time. *MODELS*, 2012.

[8] K. Welsh, P. Sawyer, and N. Bencomo. Towards requirements aware systems: Run-time resolution of design-time assumptions. In *ASE*, pages 560–563, 2011.

[9] N. Bencomo, J. Whittle, P. Sawyer, A. Finkelstein, and E. Letier. Requirements reflection: requirements as runtime entities. In *ICSE (2)*, pages 199–202, 2010.

[10] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J. Bruel. RELAX: a language to address uncertainty in self-adaptive systems requirement. *Requir. Eng.*, 15(2):177–196, 2010.

[11] 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*, RE '10, 2010.

[12] S. Hassan, N. Bencomo, and R. Bahsoon. Minimize nasty surprises with better informed decision-making in self-adaptive systems. In *SEAMS*, 2015.

[13] L. Garcia, N. Bencomo, and A. Belaggoun. The Reassessment of Preferences of Quality Attributes for Better Informed Decision-making in Self-Adaptation. Submitted to *MODELS*, 2016.

[14] N. Bencomo, A. Belaggoun, and V. Issarny.. Bayesian Artificial Intelligence for Tackling Uncertainty in Self-adaptive Systems: the Case of Dynamic Decision Networks. In *2nd International NSF sponsored Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, RAISE at ICSE* , 2013.

[15] R. Torres,N. Bencomo, H. Astudillo. Addressing the QoS drift in specification models of self-adaptive service-based systems. In *2nd International NSF sponsored Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, RAISE at ICSE* , 2013.

[16] N. Bencomo, A. Bennaceur, P. Grace, G. Blair, and V. Issarny.. The role of models@run.time in supporting on-the-fly interoperability. In *Computing* , 2012.

[17] N. Bencomo. Marie-Curie Fellowship: Requirements-aware Systems (Requirements@run.time). http://www.nellybencomo.me/ReqAwareSys.html.

[18] N. Bencomo, R. B. France, B. H. C. Cheng, and U. Aßmann, editors. *Models@run.time - Foundations, Applications, and Roadmaps [Dagstuhl Seminar 11481, 2011], LNCS*. Springer, 2014.

[19] G. Blair, N. Bencomo, and R. France. Editor's introduction: Models@run.time. *IEEE Software*, 2009.

[20] R. Calinescu, R. B. France, and C. Ghezzi. Editorial special issue on models@runtime. *Computing*, 95(3):165–166, 2013.

[21] N. Bencomo, J. Cleland-Huang, J. Guo, and R. Harrison. First international workshop on artificial intelligence for requirements engineering (aire 2014). In *RE 2014*. IEEE, 2014.

[22] N. Bencomo. QuantUn: Quantification of Uncertainty for the Reassessment of Requirements. In *RE: Next! track, RE 2015*. IEEE, 2015.

[23] B. H. Cheng, P. Sawyer, N. Bencomo, and J. Whittle. Goal-based modeling approach to develop requirements for adaptive systems with environmental uncertainty. In *MODELS Conf.*, 2009.

[24] B. H. C. Cheng and J. M. Atlee. Current and future research directions in requirements engineering. *Design Requirements Engineering: A Ten-year Perspective*, Springer, 2009.

[25] B. Cheng, R. De Lemos, H. Giese, H. A Mller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M Villegas, T. Vogel, D. Weyns, L. Baresi, B. Becker, N. Bencomo, Y. Brun, B. Cukic, R. Desmarais, S. Dustdar, G. Engels, K. Geihs, K Gschka, A. Gorla, V. Grassi, P. Inverardi, G. Karsai, J. Kramer, A. Lopes, J. Magee, S. Malek, S. Mankovskii, R. Mirandola, J. Mylopoulos, O. Nierstrasz, M. Pezz, C. Prehofer, W. Schfer, R. Schlichting, D. B Smith, J. Pedro Sousa, L. Tahvildari, K. Wong, J. Wuttke *Software engineering for self-adaptive systems: A second research roadmap. In* Software Engineering for Self-Adaptive Systems II.*, 2013.*

[26] EPSRC. Engineering grand challenges, report on outcomes of a retreat, ettington chase, stratford-upon-avon, 07 and 08 may,. Technical report, EPSRC, 2014.

[27] H. Goldsby, P. Sawyer, N. Bencomo, D. Hughes, B.H.C. Cheng. Goal-Based Modelling of Dynamically Adaptive System Requirements*. In* ECBS Conf.*, 2008.*