QuantUn: Quantification of Uncertainty for the Reassessment of Requirements

Nelly Bencomo
ALICE: The Aston Lab for Intelligent Collectives Engineering
School of Engineering and Applied Science
Aston University, UK
nelly@acm.org

***Abstract*—Self-adaptive systems (SASs) should be able to adapt to new environmental contexts dynamically. The uncertainty that demands this runtime self-adaptive capability makes it hard to formulate, validate and manage their requirements. QuantUn is part of our longer-term vision of requirements reflection, that is, the ability of a system to dynamically observe and reason about its own requirements. QuantUn's contribution to the achievement of this vision is the development of novel techniques to explicitly quantify uncertainty to support dynamic re-assessment of requirements and therefore improve decision-making for self-adaption. This short paper discusses the research gap we want to fill, present partial results and also the plan we propose to fill the gap.**

***Index Terms*—uncertainty, self-adaptation, requirements reflection, requirements assessment, Bayesian Surprise**

I. Motivation and Background

In this section we present the context, identify the gap to bridge and the roadmap of QuantUn.

A. Context

The growing pervasiveness and mobility of modern software systems contributes towards the increment of **uncertainty** these systems have about their environment. A consequence is that requirements changes cannot be fully predicted at design-time [1], [2], [3], [4], [5]. It is considerations such as these that have led to the development of self-adaptive systems (SASs) [6], which have the ability to dynamically and autonomously reconfigure their behavior to respond to changing external conditions.

A key argument in QuantUn is that current software engineering (SE) techniques do not well support the kind of dynamic appraisal of requirements needed by SASs. Furthermore, a considerable part of current research efforts in the area of SAS have focused on answering the question *when to adapt* [7] by defining the events (known or partially known) [8] and the actions the system needs to take accordingly in order to adapt [6]. The approach taken in QuantUn is rather different. We argue that a dynamically adaptive system should be able to autonomously asses deviations from its specified behaviour and use these deviation to trigger adaptation. This is a different way to tackle the question *when to adapt*, which allows the running system to be able to deal with uncertainty in a more explicit way. **Our high-level research hypothesis is that explicit treatment of uncertainty by the running system improves its judgment, i.e. evaluation of evidence, 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 machine learning [9].

> *Our aim in QuantUn is to facilitate the development of novel techniques to explicitly quantify the deviation gap from the original specified behaviour of a SAS. Therefore, depending on how large the deviation gap is, the running system may decide either to adapt accordingly or, to flag that an abnormal situation is happening.*

QuantUn is part of our long-term vision based on requirements reflection (also known as requirements-awareness) [10], in which requirements are reified as runtime entities [11]. Requirements-awareness allows systems to dynamically reason about themselves at the level of the requirements. Several inter-related key challenges have been identified to achieve the requirements reflection vision [10]: (i) runtime representation of requirements, (ii) dynamic synchronization between goals and the architecture, (iii) multi-objective decision making, (iv) self-explanation and (v) quantification of uncertainty. We have partial results related to challenges i, ii, and iii [11], [12], [13], [14]. Crucially, challenge (v) partly causes the other challenges [15] and is the focus of this paper. As a convenient side effect, our approach also takes into consideration challenge iv, i.e. multi-objective decision making.

B. The Need to Quantify Uncertainty

Uncertainty arises due to different reasons [5], [16], the stochastic nature of events in the environment, limited sensor capabilities, and difficulties in predicting how the modification of system services will affect agents' behaviors and the system goals [1], [17]. For instance, the introduction of new capabilities into the system may produce unintended effects.

We argue that there is the need to disclose uncertainty and make its treatment explicit instead of implicit, during the application of methods for the estimation of impacts on the running system due to decision-making during execution. Requirements-awareness encompasses a consideration of how to reason about uncertainty at runtime and how to reflect this reasoning by manipulating the requirements and architecture of the running system.

C. State-of-the-Art

Numerous mathematical and logical frameworks exist for reasoning about uncertainty [18] in SE in general. For example, probabilistic model checkers have been used to specify and analyse properties of probabilistic transition systems [19] and Bayesian networks to enable reasoning over probabilistic causal models [20]. However, more recently researchers have started to show attention to the treatment of uncertainty focusing on RE models [21] including our own work [3], [22].

In [23] Letier et al. tackle decision-making about alternative system designs during requirements and design engineering. Specifically, Letier et al. show how to specify partial degrees of goal satisfaction and quantify the impact of different design alternatives of the systems on high-level goals, which are used to guide requirements elaboration and design support for decision-making. Objective functions on quality variables are used to model the degree of satisfaction of these goals. The NFRs are formally specified using a probabilistic model. After, they are interpreted using application-specific measures.

More recent work by Letier et al. [21] focus on the lack of support for assessing uncertainty and its corresponding impact on risk. The authors argue for the value of reducing uncertainty before making critical decisions, and propose the application of decision analysis and multi-objective optimisation techniques to provide the needed support. The authors provide software architects with a method to describe uncertainty about the impact of alternatives on stakeholders' goals; to calculate the consequences of uncertainty; to pre-selected architecture candidates and to assess the value of gaining additional information before decision-making. The work of Letier et al. is focus on the support for decision-making during design time.

Uncertainty in adaptive systems has also been tackled using RELAX [3], a formal requirements language that explicitly addresses uncertainty inherent in self-adaptive systems. RELAX uses fuzzy logic to specify more flexible requirements to handle the uncertainty. Another approach is POISED [24] by Esfahani et al., which is based on possibility theory [25] and fuzzy mathematics to assess the impact of uncertainty on the system. As RELAX, POISED is based on fuzzy mathematics. While RELAX targets the specification of requirements, POISED aims at supporting decision-making at runtime as in our case. Welsh et al. [12] presented REAssuRE to use goal models and *Claims* (i.e., assumptions made during the requirements specification) to support decision-making and drive self-adaptation. In REAssuRE, an assumption made when selecting a reconfiguration strategy is made explicit and is recorded in the goal-based models which are made accessible during runtime. At runtime, *Claims* are monitored and, if there is evidence that a *Claim* is not valid anymore, an adaptation can be triggered in order to reach a more suitable system configuration for the new operating context signalled by the change of the veracity of the *Claim*. Ramirez et al. [13] adopt the use of *Claims* in conjunction with RELAX and introduced an approach for RELAXing *Claims* that focuses on how uncertainty can affect the validity of assumptions at runtime due to noise or lack of confidence in the monitoring infrastructure.

Finally, dynamic configuration of service-based systems was investigated by Filieri et al. in [26]. The authors presented KAMI, a framework for runtime modeling of service-based systems. KAMI focuses on non-functional properties that can be specified quantitatively in a probabilistic way and targets the challenge of making adaptation decisions under uncertainty using Markov models. Their focus is on verification, dependability with special emphasis on reliability and performance properties.

D. The Gap We Want to Fill

The approaches described above, among others, have advanced the state-of-the-art of the treatment and reasoning about uncertainty. However, these approaches have rather limited capabilities while solving uncertainty based on new evidence or information gathered at runtime. Furthermore, none of the approaches described above employ machine learning techniques. We argue that solving uncertainty at runtime needs new techniques to allow the use of new evidence found during execution to therefore reassess the requirements specifications dynamically. Moreover, the use of machine learning can prove to be useful to build the needed techniques to support knowledge acquisition and gain a better understanding of the operating environment.

As stated before, in previous research, we mainly focused on related challenges to pursue our long-term vision such as run-time representation of requirements, dynamic synchronisation between the system's goal and the architecture. More recently, we started working in the adoption of AI techniques in a novel way to quantify uncertainty [22], [27], [28] and to consequently support decision-making for self-adaptation. Specifically, we have presented possible applications of the Bayesian theory of surprise for the case of self-adaptive systems using specifically Bayesian dynamic decision networks with no focus on the impact of dynamic assessment of requirements as we do in this paper.

E. Our Plan to Fill the Gap

QuantUn aims at using RE and mathematical techniques (such as Bayesian reasoning and learning, and Fuzzy logic) to develop a novel technique to explicitly quantify uncertainty to support decision-making in self-adaptive systems. Specifically, QuantUn is based on the novel idea of the definition of **Bayesian surprise** [22] as the basis for quantitative analysis to measure degrees of uncertainty and deviations of self-adaptive systems from the expected behaviour. During requirements specification and design time models, design-time assumptions and preferences for specific decisions are specified. In [27], [28], we specify assumptions using Bayesian networks (a.k.a.beliefs networks) and Bayesian dynamic decision networks DDNs [29]. A surprise measures how monitored 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. In such a case, and based on evidence found during execution, the system may decide either to adapt accordingly or to flag that an abnormal situation is happening. A unit of surprise called *wow* has been applied [22]. The size of a surprise is expressed using the *wow* unit.

One of the benefits of being able to characterise a surprise as small is that the surprises could be used as a way of providing an implementation of temporal relaxation of requirements using techniques such as those based on the RELAX language. Essentially, small surprises could be used as suggestions that a set of non-functional requirements can be relaxed temporarily to tolerate evidence of unanticipated but transient environmental conditions potentially avoiding unnecessary adaptations.

Because of the nature of conflicting requirements (usually between non-functional requirements or soft goals), runtime quantification of uncertainty inherently involves multi-objective decision making [21] . In SE, multi-objective decision making techniques most often rely on constructing a utility function, defined as the weighted sum of the different objectives associated with non-functional requirements. However, this approach suffers from a number of drawbacks. Firstly, (i) it is well known that correctly identifying the weight of each goal is a major difficulty. Secondly, (ii) the approach hides conflicts between multiple goals under a single aggregate objective function rather than truly exposing the conflicts and reasoning about them. In [22], we have made very initial progress to tackle both drawbacks (i) and (ii).

The high-level hypothesis described earlier is too vague to be investigated and validated directly. Instead, we require a more specific formulation. The specific research **hypothesis** is that techniques such as Bayesian surprises, which facilitate dynamic assessment of requirements, can tackle these two drawbacks and be used to study and correct unwanted effects of initial preferences and also allow the disclosure of conflicts between non-functional requirements to therefore support reasoning about these conflicts. Specifically, with respect to the drawback (i), initial tests [22] have suggested that Bayesian surprises could be used to flag up situations where biased preferences (i.e. weights) set up during requirements specification can either mistakingly hide the need to perform an adaptation or create the need of unnecessary adaptations. Precisely, and for the case of DDNs, the Bayesian surprise technique can support improvement of the sensitivity analysis to agree on consistent utility functions to support decision-making provided by the decision networks.

Regarding the drawback (ii) and different from cases where just one utility function is used, Bayesian surprises, and specifically the use of dynamic decision networks, can facilitate further interpretations and analysis between discordances between non-functional requirements. This is possible because two different surprises can be associated with two different non-functional requirements what therefore, supports reasoning about conflicts between the non-functional requirements involved and based on the results associated with the two surprises found during the application of surprises.

II. Aim and Research Objectives

The **main aim of QuantUn** is to develop novel techniques to support informed decision-making by self-adaptive systems under uncertainty. The techniques explicitly represent and quantify uncertainty based on information gathered during both design time and runtime. Specifically, the techniques will include the concept of Bayesian Surprise to support quantitative analysis of uncertainty to therefore measure uncertainty degrees and deviations of self-adaptive systems from normal behaviour and consequently prompt decisions to be made by the running system. Three key research **objectives** have been identifies and are summarised as follows:

- RO1 *To develop techniques to measure degrees of uncertainty to support decision-making of SAS based on the concept of Surprises*. We aim to offer a formal definition of Surprise to measure the surprise factor associated with an observation based on the divergence between the belief distributions prior and posterior to the observation. Using Bayesian Surprises, we will develop an approach to design and implement the decision making of a SAS based on the trade-off between non-functional requirements taking into account the effect of new observations on the current model of assumptions during runtime.

- RO2 *To investigate techniques to implement temporal relaxation of requirements based on Surprises*. The techniques to be developed will enable a SAS to temporarily ”relax” requirements and face unanticipated but transient environmental conditions which could, otherwise, trigger unnecessary adaptations.

- RO3 *To disclose conflicts between non-functional requirements and support reasoning about these conflicts*. The goal spot is to use Bayesian surprises associated with different non-functional requirements to uncover relationships between these non-functional requirements and therefore, allow the re-assessment of their tradeoff based on the new knowledge acquired during runtime. The new acquired knowledge, which may have been impossible to know before execution time, will be translated to a better understanding of the operating environment that could prove to be useful in the conception of a future improved version of the SAS with respect to the current context and environmental conditions.

III. Ongoing Work

Results of ongoing work in QuantUn are partially presented in [28], [22], [30] and have focused so far on the use of DDNs to support the calculation of the surprises and support for decison-making.

Wow Unit: Quantifications of the size of Bayesian surprises will be studied and compared. A priority is to get a solid understanding of the meaning and possible uses of the unit *wow* [22]. On the one hand, positive *wow* is acknowledged when evidence is observed which double belief in the current model of assumptions. On the other hand, a negative *wow* when the evidence halve that belief. If evidence is observe which does not provoke changes the beliefs about which assumptions are expected, the evidence does not produce surprise no matter how improbable or informative it may be. On contrary, evidence that provokes a mayor redistribution of belief over the model of assumptions conveys surprise. Surprise is always computable numerically [31].

Further investigation is needed wrt relative entropy (i.e. to identify different ways to measure the divergence between two probability distributions and sensitivity analysis to agree on consistent utility functions.

Bayesian surprises to provide an implementation of RELAX: Also, we are investigating how the RELAX language [3] and Bayesian surprises can be applied to provide self-adaptive systems with the run-time flexibility to temporarily suspend or 'relax' some requirements in favour of others. We envisage run-time trade-offs of requirements being made as the environment changes. We foresee that the identification of a small surprise will be used as a way of providing an implementation of the RELAX language. The idea behind this is that small surprises can be interpreted as a sign that a given requirement can be RELAXed to therefore tolerate evidences of unanticipated but transient environmental conditions that could trigger unnecessary adaptations. Comparison with other alternative different implementations of RELAX will be performed in order to report evaluations and comparisons. Crucially, such an evaluation will provide a vehicle to find further meanings and uses of Bayesian surprises.

Reasoning about conflicts between non-functional requirements: Bayesian surprise can also be used to explore the operating environment to therefore improve its understanding. we have envisioned [22] the use of the technique Bayesian surprises to support a review process of sensitivity analysis to agree on consistent utility functions. As hinted by the initial experiments shown in [22], 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.

Bayesian surprise at design time : In [30] we have 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.

The results presented above are positive and reassuring however, more work needs to be developed to make reality the vision of QuantUn.

IV. Expectation from the RE-NEXT Forum

QuantUn presents an interdisciplinary research challenge and will combine appropriate approaches from the requirements specification of SASs including RE techniques and machine learning. Its scope is in the area of RE and the development of techniques to quantify uncertainty to improve decision making using machine learning and other AI techniques.

We believe that explicit treatment of uncertainty by the running system improves its judgment to make decisions supported by evaluation of evidence found during runtime. So far, we have introduced QuantUn, our ongoing research about the role of quantification of uncertainty, and specifically the Bayesian surprise-based techniques, to provide dynamic reassessment of requirements and support better decision-making for self-adaptation. In the long run, other AI techniques can be used to improve the current results.

We believe that QuantUn is novel and has potentially significant implications for RE research in other areas such as self-adaptive, autonomous and self-aware systems. We argue that it is critically important to solicit feedback from the RE community and create awareness of the importance of the research topic for future editions of the RE Conference. The RE-NEXT track of RE 2015 is an ideal venue for this. In a more general sense, we are eager to share experiences and ideas with RE researchers who are working on the use of artificial intelligence techniques in conjunction with RE techniques. We hope that by promoting QuantUn at RE 2015, we may be able to identify collaborators for our future work in this area.

References

[1] 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.

[2] D. Garlan, "Software engineering in an uncertain world," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, ser. FoSER '10. New York, NY, USA: ACM, 2010, pp. 125–128. [Online]. Available: http://doi.acm.org/10.1145/1882362.1882389

[3] J. Whittle, P. Sawyer, N. Bencomo, B. H. Cheng, and J.-M. Bruel, "Relax: Incorporating uncertainty into the specification of self-adaptive systems"," in *RE Conf.*, 2009.
[4] L. Baresi and C. Ghezzi, "The disappearing boundary between development-time and run-time," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, 2010.
[5] N. Esfahani and S. Malek, "Uncertainty in self-adaptive software systems," in *Software Engineering for Self-Adaptive Systems 2 (SEfSAS 2)*. Springer-Verlag, 2012.
[6] B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Eds., *Software Engineering for Self-Adaptive Systems [outcome of a Dagstuhl Seminar]*, ser. LNCS, vol. 5525. Springer, 2009. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02161-9
[7] K. Welsh and P. Sawyer, "When to adapt? identification of problem domains for adaptive systems," in *Proc. 14th International Working Conference on Requirements Engineering: Foundations of Software Quality (REFSQ08)*, Montpelier, France, 2008.
[8] J. Andersson, R. Lemos, S. Malek, and D. Weyns, "Software engineering for self-adaptive systems," B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, ch. Modeling Dimensions of Self-Adaptive Software Systems, pp. 27–47.
[9] 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.
[10] 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, pp. 95–103.
[11] N. Bencomo, J. Whittle, P. Sawyer, A. Finkelstein, and E. Letier, "Requirements reflection: requirements as runtime entities," in *ICSE (2)*, 2010, pp. 199–202.
[12] K. Welsh, P. Sawyer, and N. Bencomo, "Towards requirements aware systems: Run-time resolution of design-time assumptions," in *ASE*, 2011, pp. 560–563.
[13] A. Ramirez, B. Cheng, N. Bencomo, and P. Sawyer, "Relaxing claims: Coping with uncertainty while evaluating assumptions at run time," *MODELS*, 2012.
[14] N. Bencomo, K. Welsh, P. Sawyer, and J. Whittle, "Self-explanation in adaptive systems based on runtime goal-based models," in *Transactions on Computational Collective Intelligence*, december 2014.
[15] H. Giese, N. Bencomo, L. Pasquale, A. Ramirez, P. Inverardi, S. Wtzoldt, and S. Clarke, "Living with uncertainty in the age of runtime models," in *Models@run.time*, ser. LNCS, N. Bencomo, R. France, B. Cheng, and U. Assmann, Eds. Springer International Publishing, 2014, vol. 8378, pp. 47–100.
[16] A. J. Ramirez, A. C. Jensen, and B. H. C. Cheng, "A taxonomy of uncertainty for dynamically adaptive systems," in *7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2012, Zurich, Switzerland, June 4-5, 2012*, 2012, pp. 99–108. [Online]. Available: http://dx.doi.org/10.1109/SEAMS.2012.6224396
[17] B. H. C. Cheng and J. M. Atlee, "Current and future research directions in requirements engineering," in *Design Requirements Engineering: A Ten-year Perspective*, K. Lyytinen, P. Loucopulos, J. Mylopoulos, and B. Robinson, Eds. Springer, 2009, pp. 11–43.
[18] J. Y. Halpern, *Reasoning about Uncertainty*. The MIT Press, October 2003.
[19] M. Z. Kwiatkowska, G. Norman, and D. Parker, "Probabilistic symbolic model checking with prism: A hybrid approach," in *TACAS*, 2002, pp. 52–66.
[20] N. Fenton and M. Neil, "Making decisions: using bayesian nets and mcda," *Knowl.-Based Syst.*, vol. 14, no. 7, pp. 307–325, 2001.
[21] E. Letier, D. Stefan, and E. T. Barr, "Uncertainty, risk, and information value in software requirements and architecture," in *Proceedings of ICSE*, ser. ICSE 2014. New York, NY, USA: ACM, 2014, pp. 883–894.
[22] N. Bencomo and A. Belaggoun, "A world full of surprises: bayesian theory of surprise to quantify degrees of uncertainty," in *ICSE Companion*, 2014, pp. 460–463.
[23] E. Letier and A. van Lamsweerde, "Reasoning about partial goal satisfaction for requirements and design engineering," *SIGSOFT Softw. Eng. Notes*, vol. 26, 2004.
[24] N. Esfahani, E. Kouroshfar, and S. Malek, "Taming uncertainty in self-adaptive software," in *Proc.of the 19th ACM SIGSOFT Symp FSE*, no. 11, 2011.
[25] L. Zadeh, "Fuzzy sets as a basis for theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.
[26] A. Filieri, C. Ghezzi, and G. Tamburrelli, "A formal approach to adaptive software: continuous assurance of non-functional requirements," *Formal Asp. Comput.*, vol. 24, no. 2, pp. 163–186, 2012.
[27] 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.
[28] N. Bencomo, A. Belaggoun, and V. Issarny, "Dynamic decision networks to support decision-making for self-adaptive systems," in *8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2013.
[29] S. J. Russell and P. Norvig, *Artificial intelligence - a modern approach: the intelligent agent book*, ser. Prentice Hall series in artificial intelligence. Prentice Hall, 1995.
[30] S. Hassan, N. Bencomo, and R. Bahsoon, "Minimize nasty surprises with better informed decision-making in self-adaptive systems," in *10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2015.
[31] P. Baldi and L. Itti, "Of bits and wows: A bayesian theory of surprise with applications to attention," *Neural Networks*, vol. 23, no. 5, pp. 649–666, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2009.12.007