

The Reassessment of Preferences of Non-Functional Requirements for Better Informed Decision-making in Self-Adaptation

Luis H. Garcia Paucar
ALICE, Aston University, UK
Email: garciapl@aston.ac.uk

Nelly Bencomo
ALICE, Aston University, UK
Email: nelly@acm.org

Abstract—Decision-making requires the quantification and trade-off of multiple non-functional requirements (NFRs) and the analysis of costs and benefits between alternative solutions. Different techniques have been used to specify utility preferences for NFRs and decision-making strategies of self-adaptive systems (SAS). These preferences are defined during design-time. It is well known that correctly identifying the weight of the NFRs is a major difficulty. In this paper we present initial results of a novel approach that provides a set of criteria to re-assess NFRs preferences given new evidence found at runtime using dynamic decision networks (DDNs). The approach uses both conditional probabilities provided by DDNs and the concept of Bayesian surprise. The results show that our approach supports better informed decisions under uncertainty by identifying new situations where the current SAS preferences may need to be re-evaluated to improve the levels of satisfaction of NFRs.

Index Terms—Self-adaptation; decision making; non-functional requirements trade-off, uncertainty.

I. INTRODUCTION

Decision-making requires the quantification and trade-off of multiple NFRs and the analysis of the costs and benefits between alternative solutions. Decision-making is at the core of self-adaptation. An important issue is the specification of an utility function including the utility preferences (a.k.a. weights) associated with NFRs and defined in the decision-making strategies to support self-adaptation. Priorities associated with NFRs described in the requirements specification may vary from stakeholder to stakeholder and from one envisaged situation to another. Different priorities may imply different decisions by the system. Further, in self-adaptive systems (SAS), the assumptions made at design time may change during runtime and those changes can cause changes on priorities and therefore on utility preferences. Modelling and reasoning with prioritization and preferences is a research field that needs more attention [1]. Different studies that approach these issues have been performed [2], [3], [4], [5], [6]. They have produced different approaches but also critical challenges that must be further explored. Most of the current approaches tend to focus on design time issues and are effective in specific domains but unlikely to be generalizable. Further, the need of updating utility preferences or uncovering relationships between NFRs during runtime has been neglected [6]. The steps of monitoring the environment,

detecting the need of (self-) adaptation and deciding how to react are challenges identified for SAS [2] and these challenges need to involve the role of preferences and the re-prioritization of NFRs due to new evidence found at runtime.

The main contribution of this paper is the combination of conditional probabilities (using Bayesian inference) based on models of DDNs with Bayesian surprises to allow the reassessment of NFRs preferences during the operation or simulation of the system. This represents a clear advance over the existing literature and our previous work. The paper is organized as follows: In Section 2, we provide a review of related work and identify the research gap. In Section 3, we show and discuss preliminary results to fill the identified research gap. In Section 4, we show and explain the experiment performed. Finally, in Section 5, we conclude with respect to our findings, and identify and discuss future research challenges.

II. DECISION MAKING AND UTILITY PREFERENCES

When we make decisions, a natural approach is to evaluate our different alternatives and choose one based on some criteria [7]. In SAS we must build intelligent systems able to apply this natural way of reasoning under environmental uncertain conditions. How to ensure a reliable behaviour (i.e., optimize the system's behaviour) trading-off multiple goals competing among them and being constantly affected by external changing conditions is the field of action of a well known set of methods: Multi Criteria Decision Analysis Methods (MCDA) [8]. MCDA methods are currently applied in different fields but more than ever in self-adaptation. In the context of self-adaptation and autonomic computing, multi-criteria utility functions provide the objective function for self-optimization mapping each possible state of the system into a real scalar value to allow the trade-off [9]. Different MCDA techniques are being used for both, decision-making and preferences specification in SAS. Some popular MCDA techniques such as Pareto Optimal [10] are being used to reason at runtime and discover a set of optimal adaptation alternatives. The final alternative selection could demand the user intervention or be part of a fully autonomic system behaviour. Other MCDA approaches such as Analytic Hierarchical Process (AHP) [11] are also used for specifying quality attribute preferences at

design time collected from system’s stakeholders following a more formal and objective approach. In the rest of this section we explore different approaches for decision-making in SAS, focusing on aspects related to elicitation and dynamic update of preferences.

A. State-of-the-Art

This section briefly overviews different approaches for decision-making and utility preferences for SAS and provides a context to define the research gap to be tackled. A more complete survey of these approaches are shown in the report [12].

1) *Preference elicitation*: The most widespread method for preference elicitation is with user intervention. In [13], [14], [15], [16], [1] preferences are elicited initially with information from system stakeholders. Authors in [17], [16], [1] complement this approach by using multi criteria decision analysis methods (MCDA). The most common MCDA method for preference elicitation is AHP [11]. For example in [1] weights are assigned to preference goals using a quantitative requirement prioritization scheme: AHP. Elahi et. al. [16] incorporate Stakeholders preferences by using a MCDA-based method: the Even Swaps Method [16], to avoid the elicitation of numerical weights. Garcia Galan et. al. [18] defined the systems initial preference by using initial configurations defined by the stakeholders. While it is not common, some authors, define preferences in an autonomous way, for example, Ramirez et. al. [19] use predefined models to derive utility functions.

2) *Preference updating*: This is a field that needs more exploration. Several approaches [1], [15], [19] work only with the initial preferences at runtime and they do not support preference updating, some others support it but not in an autonomous way, as they require user intervention: [13] shows that if users do not agree with the final solution (after each round of adaptation), they can revise the configuration values using an interface. The users preferences elicited in this revision allow either tuning the weights of existing goals or the generation of new ones. In [20] a user interface is used to manipulate thresholds on preferences at runtime. However, Peng et. al. [21] shows first experiences of autonomous preference updating by monitoring the environment at runtime and using a preference tuning algorithm. In [9], Walsh et. al. use machine learning techniques to update the systems preferences.

3) *Decision making*: Preferences defined or updated, are used by SAS for their decision making process. Several trends have been identified. The most common approaches for decision making are utility functions and Pareto Optimal. Ramirez et. al. [19] perform a goal model transformation into utility functions for monitoring software requirements at runtime. For Peng et. al. [21] the decision process involves three main components: (i) a Proportional Integral Derivative (PID) Controller, (ii) a Preference-driven Goal Reasoner (GR) and (iii) an Architecture Configurator (AC). In this approach, the NFR preferences are updated autonomously by the PID controller, establishing a balance at runtime between the

earned business value and quality measurement. The reasoning approach uses a Pareto Optimal algorithm that yields a set of alternative configurations. A satisfiability solver (SAT) obtains the optimal solution. Some approaches have scalability issues. For example, authors in [?] are still investigating how to deal with scalabilities issues associated with the number of constraints. Garcia-Galan et. al. [18] have scalability problems with the size of its configuration Space. Approaches like Sousa et. al. [20], use forecast and learning elements for the decision making process.

B. Discussion

The study shows that so far, even if scarce, there have been important research efforts towards decision-making for SAS taking into account NFRs. However, relevant results about dynamic update of utility preference is still a challenge. The study also shows that preferences specification is still a task executed primarily at design time. The approaches studied show that different MCDA techniques stand out as common techniques used for reasoning optimization. Some approaches use ad-hoc methods for collecting users’ preferences, while others use techniques such as MCDA [15], [16], [21]. In [16], [13], [9] the support for preferences update exists but requires user intervention. Some approaches offer potential to support autonomic preference updating. For example, authors of [18] propose an approach for mining users’ behaviour while authors of [21] use an autonomic preference tuning algorithm. Ramirez et. al. [19] show an autonomic generation of utility functions. Authors in [9], [22] highlight the relevance of using models that need to be learned and refine at runtime during the operation of the system. A more thorough analysis of the state-of-the-art is shown in [12].

III. REASSESSING UTILITY PREFERENCES AT RUNTIME

In this section we briefly offer a background about the DDN-based approach to support decision-making under uncertainty and Bayesian surprise [14] to therefore set the basis to explain the new ideas about reassessment of utility preferences during runtime in subsection III-C.

A. DDNs Model for Decision-Making in SAS

[22], [23] show how dynamic-decision networks (DDNs) are abstractions that serve the purpose of modelling beliefs about the world, linking preferences and observation models (to obtain evidence from the operational environment) with states of the world in order to make informed decisions. DDNs have been used as a mechanism which allows SASs to keep track of the current state and trade-off of NFRs [22], [23]. They are abstractions for reasoning about the world over time [24]. DDNs provide a set of random variables that represent the NFRs. Fig. 1 shows a DDN during two time slices where X_t denote a set of state variables (i.e. NFRs) at time t , which are unobservable, and E_t to denote the observable evidence variables. A DDN links decision maker preferences U_t (i.e. utility nodes), state and evidence variables to make informed decisions D_t (i.e. decision nodes).

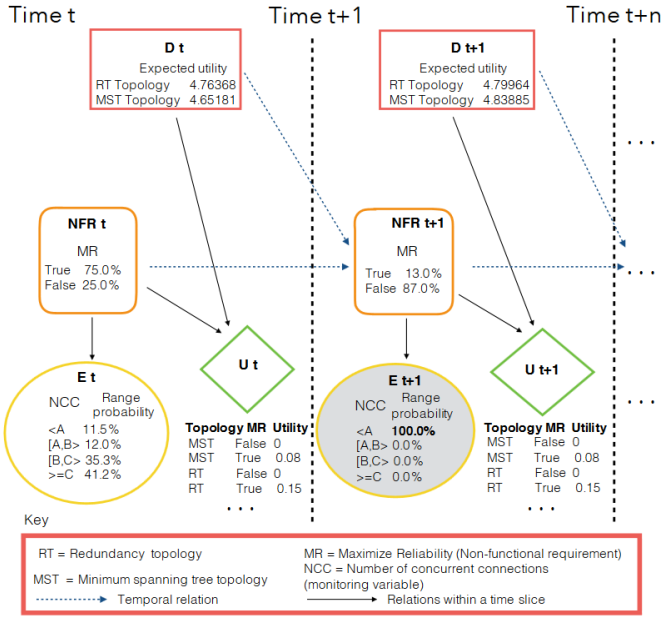


Fig. 1. Example of DDN structure.

The expected utility (EU) is computed using the equation 1 as follows:

$$EU(d_j|e) = \sum_{x_i \in X} U(x_i, d_j) \times P(x_i | e, d_j) \quad (1)$$

In equation 1 above, $P(x_i | e, d_j)$ is the conditional probability of $X = x_i$ given the evidence e and the decision d_j . The random variables X_i (i.e. state nodes in the DDN) correspond to the levels of satisfaction of the NFRs. Solving a decision network (DN) refers to finding the decision that maximizes EU.

B. Bayesian Surprises to Quantify Deviations from Expected Behaviour

A surprise value means that the evidence provided from the environment has caused a difference between the prior and posterior probabilities of an event. A Bayesian surprise measures how observed data affects the models or assumptions of the world during runtime [14]. The surprise represents the divergence between the prior and posterior distributions of a NFR and is calculated by using the Kullback-Leibler divergence (KL) [25]. Let us have a non-functional requirement NFR_i , and E representing the evidence provided by the properties monitored as variables in the execution environment. $P(NFR_i)$ is the prior probability of the non-functional requirement NFR_i being partially satisfied and $P(NFR_i|E)$ is the posterior probability of the NFR_i being partially satisfied given the evidence E .

$$S(NFR_i, E) = KL(P(NFR_i|E), P(NFR_i)) = \sum_i P(NFR_i|E) \log \frac{P(NFR_i|E)}{P(NFR_i)} \quad (2)$$

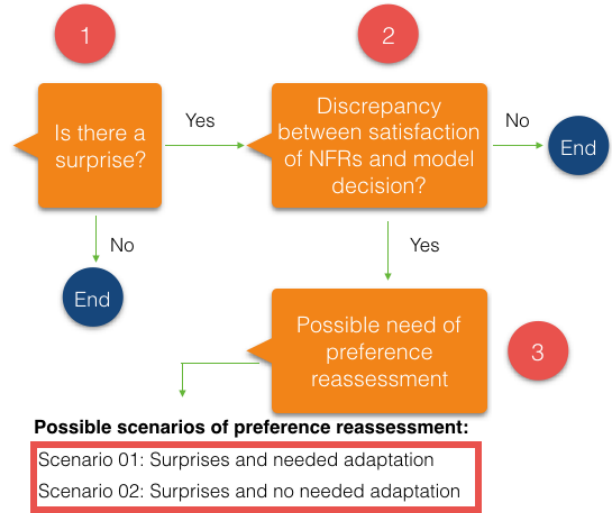


Fig. 2. Approach for preference reassessment at runtime

C. Towards Reassessment of Utility Preferences

Bayesian surprises have been exploited during runtime to improved better informed decision-making at runtime [26]. The approach supports the quantification of uncertainty over different time slices at runtime and helps the system improve its behaviour based on learning during the operation of the system. This learning process has shown to be memory intensive and therefore have presented scalability and memory issues in the past [23]. In this paper, in addition to our novel approach, we also have improved the DDN models used in the past to therefore improve the scalability issues. Currently, the experiments can be run during a bigger number of time slices.

Our method aims to improve the decision making allowing the access to new information and evidence about possible adverse effects of the utility preferences during execution by:

- Allowing the identification of a range of scenarios during the execution of the system and the corresponding effects they have on the satisfaction of relevant NFRs.
- Highlighting the environmental properties in the execution environment which have highest, and possible unknown effects at design time on the satisfaction of the NFRs.

The method involves the following steps:

- At runtime, per each time slice, a Bayesian Surprise is computed for each state variable (i.e., each NFR).
- If a surprise is detected, the next step is to evaluate the current level of satisfaction of the NFRs (by using Bayesian Inference) to compare it with the decision suggested by the model (i.e., the decision to adapt or not suggested by the DDN). It is important to highlight that the probability distribution of each NFR is not influenced by the utility nodes of the model (i.e., user preferences).
- If the decision taken by the model (which is influenced by the utility nodes) is not contributing to the satisfaction of the NFRs, the detected situation is highlighted as a possible scenario needing preference reassessment.

Fig. 2 shows a graphic representation of the process. By using surprises and conditional probabilities provided by the DDNs to revising the initial utility preferences during runtime, the approach contributes to support better understanding of the execution environment while assessing the corresponding responses of the running system.

IV. EXPERIMENTS

The experiments are based on the application of our approach to the case study of a Remote Data Mirroring (RDM) application. An RDM is a technique with the goal of protecting data against inaccessibility and to provide further resistance to data loss [27]. An RDM can be configured in different ways, for example in terms of the network's topology, minimum spanning tree (MST) vs. redundant topology (RT) or in terms of data distribution (e.g., synchronous vs. asynchronous propagation).

Let us focus on the network's topology. Different network topologies can be used to implement this technique to offer different costs and benefits that would need to be traded-off. A RT topology offers a higher level of reliability than a MST topology. However, the costs of maintaining a non-stop redundant topology may be prohibitive. An assessment of the trade-off between these two choices need to be made at design-time and revisited at runtime under the light of new evidence found.

For the experiments of this paper, a DDN for the application of RDM has been designed according to two alternatives network topologies: MST and RT as described above. Each configuration provides different levels of data protection and costs which are the quality attributes Minimize Operational Cost (MO) and Maximize Reliability (MR).

The scenario that has been used to perform the experiments is described as follows: the states of two monitored variables NCC="Number of Concurrent Connection" and C="Fewer active network links reduce operational costs" are monitored during runtime. The value of C can be either true or false and the values for NCC are different possible ranges represented by the following expressions: $NCC < A$, NCC in $[A, B >$, NCC in $[B, C >$, and $NCC \geq C$. At design time, C have been considered valid (true) and $NCC \geq C$.

In order to evaluate the DDN shown in Fig. 1, we have considered the following initial conditional probabilities:

- $P(MO = true|MST)=0.7$,
- $P(MO = false|MST)=0.3$,
- $P(MO = true|RT)=0.35$,
- $P(MO = false|RT)=0.65$,
- $P(MR = true|MST)=0.49$,
- $P(MR = false|MST)=0.51$,
- $P(MR = true|RT)=0.75$,
- $P(MR = false|RT)=0.25$,
- $P(C = true|MO = true)=0.999$,
- $P(C = false|MO = true)=0.001$,
- $P(NCC < A|MR = true)=0.02$,
- $P(NCC \text{ in } [A, B > |MR = true)=0.04$,
- $P(NCC \text{ in } [B, C > |MR = true)=0.42$,

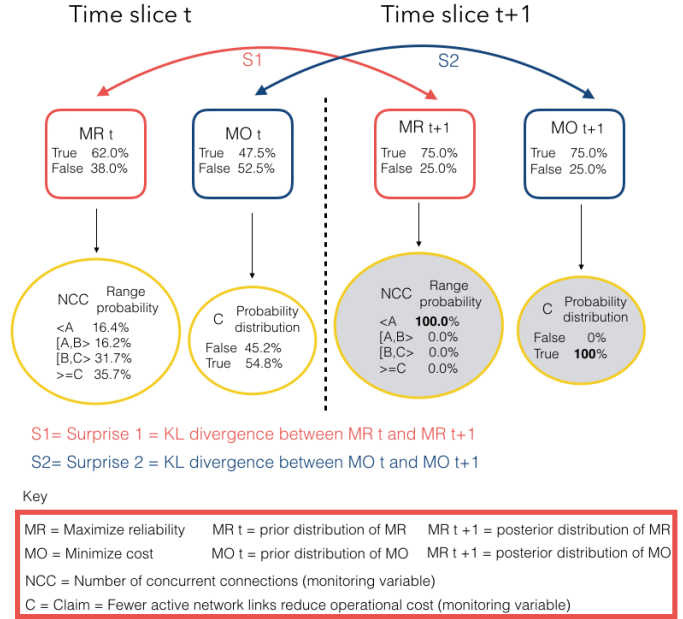


Fig. 3. Structure for Computing Surprises - Exp.01 and Exp. 02

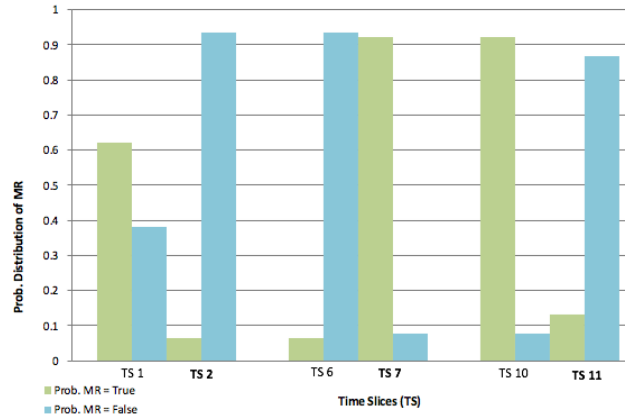


Fig. 4. Prob. distribution of NFR Maximize Reliability - Exp. 1

- $P(NCC \geq C | MR = true)=0.52$

Two experiments have been implemented and for each one Surprises have been applied. Consider the situation where the prior models for surprise computation are $P(MR_t)$ and $P(MO_t)$ and the posterior models when an evidence has been observed over the time are $P(MR_{t+1}|NCC)$ and $P(MO_{t+1}|C)$ (see Fig. 3). We have computed surprises based on the KL-divergence between the prior and the posterior probabilities during 13 time slices.

A. Experiment 1

Surprises take place in several time slices where different specific situations have been identified. Fig. 6 shows the observed values for NCC and C variables and the surprises S1 and S2. S1 and S2 are the divergence between the prior and posterior distributions for the quality attribute MR and MO respectively. Both, S1 and S2, are computed for each time slice during the experiment.

1) *Surprises and adaptation:* In time slice 2 we can observe two surprises and that an adaptation is suggested by

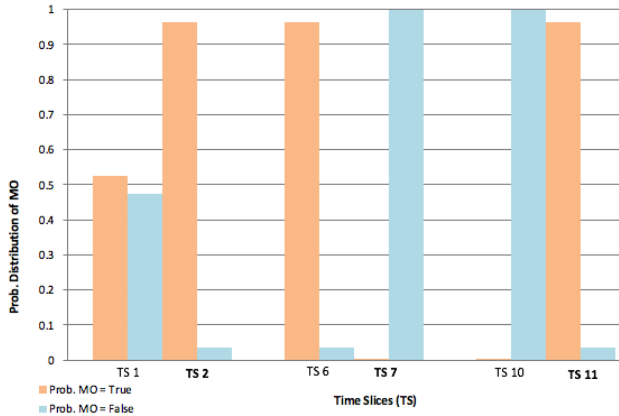


Fig. 5. Prob. distribution of NFR Minimize Costs - Exp. 1

the DDN (see Fig. 6). Studying the conditional probabilities provided by the DDN under the current conditions: $P(MR = true|NCC < A, C = true) = 6.47\%$ (see Fig. 4, time slice 2) and $P(MO = true|NCC < A, C = true) = 96.5\%$ (see Fig. 5, time slice 2), we can observe that while the probability for Maximize Reliability is low the probability for Minimize Operational Cost is high. The selected choice, i.e. to adapt from RT to MST, certainly sounds like a good selection given the current situation: low probability for Maximize Reliability and high probability for Minimize Operational Cost. Using MST would avoid unnecessary costs as the complementary information provided by the conditional probabilities suggest to use the less costly topology MST. The surprises and the conditional probabilities help us to identify up this situation. This situation is an example when surprises are generated, the conditional probabilities and the adaptation performed by the system agree to support the same behaviour by the system improving confidence.

In time slice 7 we can observe two surprises and that an adaptation is suggested by the DDN (see Fig. 6). Studying the conditional probabilities provided by the DDN under the current conditions: $P(MR = true|NCC \geq C, C = false) = 94.5\%$ (see Fig. 4, time slice 7) and $P(MO = true|NCC \geq C, C = false) = 0.057\%$, (see Fig. 5, time slice 7) we can observe that the probability for Maximize Reliability is high, however on the other hand, the probability for Minimize Operational Cost is very low. The selected choice, i.e. to adapt from MST to RT, certainly may be a good selection for the current situation: high probability for Maximize Reliability and low probability for Minimize Operational Cost. The complementary information provided by the conditional probabilities suggest to use the topology MST. The surprises and the conditional probabilities help us in flagging up this situation. Again, this situation is an example when surprises generated, the conditional probabilities and the adaption performed by the system agree.

2) *Surprises and needed adaptations:* We can observe that in time slice 11 there are surprises however, the DDN has not suggested any adaptation (see Fig. 6). Studying the conditional probabilities provided by the DDN under the current conditions: i.e. $P(MR = true|NCC \text{ in } [A, B], C =$

Time slice (TS)	Adaptation	S1	S2	NCC Monitored values	C Monitored values	Current Topology
1	-	0	0	$NCC \geq C$	True	RT
2	Yes	1.003584235223648	0.7150925003192315	$NCC < A$	True	MST
3		0.0	0.0	$NCC < A$	True	MST
4		0.0	0.0	$NCC < A$	True	MST
5		0.0	0.0	$NCC < A$	True	MST
6		0.0	0.0	$NCC < A$	True	MST
7	Yes	3.246411633405382	4.808007459687037	$NCC \geq C$	False	RT
8		0.0	0.0	$NCC \geq C$	False	RT
9		0.0	0.0	$NCC \geq C$	False	RT
10		0.0	0.0	$NCC \geq C$	False	RT
11		2.6203981427168355	9.382088854787371	$NCC \text{ in } [A, B]$	True	RT
12		0.0	0.0	$NCC \text{ in } [A, B]$	True	RT
13		0.0	0.0	$NCC \text{ in } [A, B]$	True	RT

Fig. 6. Surprises and monitored values - Exp. 1

$true) = 13.3\%$ (see Fig. 4, time slice 11) and $P(MO = true|NCC \text{ in } [A, B], C = true) = 96.4\%$ (see Fig. 5, time slice 11), we can observe that the probability for Maximize Reliability is low. However, on the other hand, the probability for Minimize Operational Cost is high. The selected choice, i.e. not to adapt, certainly may not be the best choice given the current situation: low probability for Maximize Reliability and High probability for Minimize Operational Cost. Continuing using RT as the configuration would create unnecessary costs as the complementary information provided by the conditional probabilities suggest the use of the less costly topology MST. The surprises and the conditional probabilities, which crucially are not influenced by the stakeholders' preferences, help us to flag up this situation. The situation identified is an example of how surprises and the conditional probabilities of the DDN can flag up the need of adaptation. Crucially, the above imply the need to revisit the preferences defined by the stakeholders previously providing the opportunity to improve the behaviour of the system.

B. Experiment 2

The observed values for NCC and C variables and the surprises S1 and S2 are shown in Fig. 7.

1) *Surprises and adaptation:* In time slice 2 we can observe surprises and that an adaptation is suggested by the DDN (see Fig. 7). Studying the conditional probabilities provided by the DDN under the current conditions: $P(MR = true|NCC < A, C = true) = 6.47\%$ and $P(MO = true|NCC < A, C = true) = 96.5\%$, we can observe that the probability for Maximize Reliability is low. On the other hand, the probability for Minimize Operational Cost is high. The selected choice, i.e. to adapt from RT to MST, certainly looks to be a good selection given the current situation: low probability for Maximize Reliability and high probability for Minimize Operational Cost. Crucially, the complementary information provided by the conditional probabilities suggest to use the topology MST. The surprises and the conditional probabilities help us in identifying this situation. The situation is therefore an example of agreement

Time slice (TS)	Adaptation	S1	S2	NCC Monitored values	C Monitored values	Current Topology
1	-	0	0	NCC >= C	True	RT
2	Yes	1.003584235223648	0.7150925003192315	NCC < A	True	MS
3		0.04381918744008068	1.6949044399769875E-4	NCC in [A,B]	True	RT
4		0.0	0.0	NCC in [A,B]	True	RT
5		0.0	0.0	NCC in [A,B]	True	RT
6		0.01424282209204613	4.76049740064592	NCC in [A,B]	False	RT
7		0.0	0.0	NCC in [A,B]	False	RT
8		0.0	0.0	NCC in [A,B]	False	RT
9		1.2757774188794533	8.762523955312945	NCC in [B,C]	True	RT
10		0.0	0.0	NCC in [B,C]	True	RT
11		0.0	0.0	NCC in [B,C]	True	RT
12		0.0	0.0	NCC in [B,C]	True	RT
13		0.0	0.0	NCC in [B,C]	True	RT

Fig. 7. Surprises and monitored values - Exp. 2

behavior between the surprises generated, the conditional probabilities and the adaption performed by the system.

2) *Surprises and unneeded adaptation:* We can see that in time slice 3 there are surprises and an adaptation is suggested by the DDN (see Fig. 7). Studying the conditional probabilities provided by the DDN under the current conditions: $P(MR = true | NCCin[A, B >, C = true) = 13.3\%$ and $P(MO = true | NCCin[A, B >, C = true) = 96.4\%$, we can see that the probability for Maximize Reliability is low. On the other hand, the probability for Minimize Operational Cost is high. The selected choice, i.e. to adapt, certainly may not be a good selection for the current situation: low probability for Maximize Reliability and high probability for Minimize Operational Cost. Using RT would create unnecessary costs as the complementary information provided by the conditional probabilities suggest to use the less costly topology MST. The surprises and the conditional probabilities supported flagging up the situation. The situation is an example of how surprises and conditional probabilities can highlight the need of avoiding unnecessary adaptations. The previous findings imply the need to reassess the quality preferences defined by the stakeholders during design-time.

3) *Surprises as a false positive:* In time slice 6 we can observe surprises and the fact that there is no adaptation recommended by the DDN (see Fig. 7). Studying the conditional probabilities provided by the DDN under the current conditions: $P(MR = true | NCCin[A, B >, C = false) = 18.3\%$ and $P(MO = true | NCCin[A, B >, C = false) = 0.14\%$, we can see that the probability for Maximize Reliability is low. On the other hand, the probability for Minimize Operational Cost is very low. The selected choice, i.e. not to adapt, certainly looks to be a good selection for the current situation: low probability for Maximize Reliability and very low probability for Minimize Operational Cost. The complementary information provided by the conditional probabilities suggest that using RT is a better option than using MST. This situation is an example of a false positive, there are surprises but is not needed any adaptation. However, the conditional probabilities help us flagging up this

situation providing a better informed decision making.

4) *Surprises and needed adaptation:* In time slice 9 there are surprises however, the DDN has not suggested any adaptation (see Fig. 7). This situations and its interpretation is equivalent to Experiment 1, time slice 11, i.e., is an example of how surprises and the conditional probabilities can flag up the need of adaptation.

C. Analysis of Results

During the previous experiments, using the approach we were able to identify four scenarios at runtime with more opportunities to enhance the decision making of the system:

- Scenario 01 - surprises and needed adaptation. There are surprises, there is no adaptation, and the conditional probabilities suggest to make an adaptation.
- Scenario 02 - surprises and no needed adaptation. There are surprises, there is adaptation, and the conditional probabilities suggest not to make an adaptation.
- Scenario 03 - surprises and adaptation. There are surprises, there is adaptation, and the conditional probabilities suggest to make an adaptation.
- Scenario 04 - surprises as a false positive. There are surprises, there is no adaptation, however the conditional probabilities suggest no adaptation.

Using the approach, scenarios 01 and 02 have been identify to flag up the need for revisiting the quality attribute preferences defined by the stakeholders previously and provide an opportunity to improve the decision making and behaviour of the system. Scenario 03 shows an agreement between the suggested adaptation and surprises providing more confidence in the decision making of the SAS. Scenario 04 is a false positive for surprises, however the conditional probabilities allow us to highlight the fact that the DDN was triggering the correct behaviour allowing a better informed decision making and the possibility of providing a system with self-explanation capabilities [28].

It was possible to explore all these scenarios only by using surprises and Bayesian inference (conditional probabilities) at runtime. The new implemented model is an improved version of previous experiments that allows us to implement experiments with more complexity and covering more time slices. Now that we can evaluate NFR preferences at runtime, the next possible step will be to explore mechanisms to use this information for autonomic NFR preferences updating. Different from previous initial experiments, we have used monitorables (i.e. evidence nodes) with major level of granularity to allow us the exploration of further potential situations that suggest the need for reassessment of NFR preferences. These new experiments showed how the values monitored as evidence provide different impacts on the satisfaction level of the NFRs allowing better reasoning. The new implemented model is an improved version of previous experiments that provides better scalability.

V. CONCLUDING REMARKS

As shown by experiments discussed in this paper, utility preferences associated with NFRs initially provided by domain

experts during the sensitivity analysis at design time may not be ideal for specific cases to be found at runtime. Badly chosen preferences can either make the system miss adaptations or suggest unnecessary adaptations that may degrade the behavior of the running system. To our knowledge, currently there is no related work to this specific issue. We have shown the power of runtime abstractions [29] based on DDN models (i.e. Bayesian surprises and prior and posterior probabilities given runtime evidence) to allow further study of contexts that were not fully understood during the requirements elicitation. A novel contribution is that the approach takes advantage of machine learning to collect evidence to improve the understanding of the environment and the decision making process by the running system [22]. Challenges for future work cover research questions such as how to optimize and get scalable reasoning techniques including the dynamic updating for NFR preferences once we have identified NFR utilities that are not suitable to the new contexts found.

REFERENCES

- [1] S. Liaskos, S. A. McIlraith, S. Sohrabi, and J. Mylopoulos, "Representing and reasoning about preferences in requirements engineering," *Requir. Eng.*, vol. 16, no. 3, pp. 227–249, Sep. 2011. [Online]. Available: <http://dx.doi.org/10.1007/s00766-011-0129-9>
- [2] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auton. Adapt. Syst.*, vol. 4, no. 2, pp. 14:1–14:42, May 2009. [Online]. Available: <http://doi.acm.org/10.1145/1516533.1516538>
- [3] B. H. Cheng and et al., "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. Software Engineering for Self-Adaptive Systems: A Research Roadmap, pp. 1–26.
- [4] E. Yuan, N. Esfahani, and S. Malek, "A systematic survey of self-protecting software systems," *ACM Trans. Auton. Adapt. Syst.*, vol. 8, no. 4, pp. 17:1–17:41, Jan. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2555611>
- [5] M. Salama, R. Bahsoon, and N. Bencomo, "Managing trade-offs in self-adaptive software architectures: A systematic mapping study," in *Managing trade-offs in adaptable software architectures*, I. Mistrk, N. Ali, J. Grundy, R. Kazman, and B. Schmerl, Eds. Elsevier, 2016.
- [6] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, "A survey on engineering approaches for self-adaptive systems," *Pervasive and Mobile Computing*, vol. 17, Part B, pp. 184 – 206, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S157411921400162X>
- [7] A. Ishizaka and P. Nemery, *Multi-criteria decision analysis : methods and software*. Chichester: J. Wiley & Sons, 2013. [Online]. Available: <http://opac.inria.fr/record=b1135342>
- [8] J. Figueira, S. Greco, and M. Ehrgott, *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer, 2005.
- [9] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das, "Utility functions in autonomic systems," in *Autonomic Computing, 2004. Proceedings. International Conference on*, May 2004, pp. 70–77.
- [10] M. Harman, P. McMinn, J. T. de Souza, and S. Yoo, "Empirical software engineering and verification," B. Meyer and M. Nordio, Eds. Berlin, Heidelberg: Springer-Verlag, 2012, ch. Search Based Software Engineering: Techniques, Taxonomy, Tutorial, pp. 1–59. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2184075.2184076>
- [11] T. Saaty, "Decision making with the analytic hierarchy process," *Inter. Journal of Services Sciences.*, 2008.
- [12] L. Garcia-Paucar and N. Bencomo, "A survey on preferences of quality attributes in the decision-making for self-adaptive and self-managed systems: the bad, the good and the ugly," Aston University, Tech. Rep., 2016.
- [13] H. Song, S. Barrett, A. Clarke, and S. Clarke, "Self-adaptation with end-user preference: Using run-time models and constraint solving," in *the Intl. Conference MODELS*, USA, 09/2013 2013.
- [14] 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.
- [15] 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.
- [16] G. Elahi and E. Yu, "Requirements trade-offs analysis in the absence of quantitative measures: A heuristic method," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 651–658. [Online]. Available: <http://doi.acm.org/10.1145/1982185.1982331>
- [17] E. Letier and A. van Lamsweerde, "Reasoning about partial goal satisfaction for requirements and design engineering," in *Symposium on Foundations of software engineering*. NY, USA: ACM, 2004, pp. 53–62.
- [18] J. García-Galán, L. Pasquale, P. Trinidad, and A. Ruiz-Cortés, "User-centric adaptation of multi-tenant services: Preference-based analysis for service reconfiguration," in *SEAMS*, ser. SEAMS 2014. USA: ACM, 2014, pp. 65–74. [Online]. Available: <http://doi.acm.org/10.1145/2593929.2593930>
- [19] A. J. Ramirez and B. H. C. Cheng, "Automatic derivation of utility functions for monitoring software requirements," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6981 LNCS, pp. 501–516, 2011.
- [20] J. P. Sousa, R. K. Balan, V. Poladian, D. Garlan, and M. Satyanarayanan, "User guidance of resource-adaptive systems," in *In Proc. of International Conference on Software and Data Technologies*, 2008.
- [21] X. Peng, B. Chen, Y. Yu, and W. Zhao, "Self-tuning of software systems through goal-based feedback loop control," in *Requirements Engineering Conference (RE)*, Sept 2010, pp. 104–107.
- [22] 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.
- [23] N. Bencomo, A. Belaggoun, and V. Issarny, "Dynamic decision networks to support decision-making for self-adaptive systems," in *(SEAMS)*, 2013.
- [24] 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.
- [25] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.
- [26] 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.
- [27] A. Ramirez, B. Cheng, N. Bencomo, and P. Sawyer, "Relaxing claims: Coping with uncertainty while evaluating assumptions at run time," *MODELS*, 2012.
- [28] N. Bencomo, K. Welsh, P. Sawyer, and J. Whittle, "Self-explanation in adaptive systems," in *Engineering of Complex Computer Systems (ICECCS), 2012 17th International Conference on*, july 2012.
- [29] G. Blair, N. Bencomo, and R. B. France, "Models@ run. time," *Computer*, vol. 42, no. 10, pp. 22–27, 2009.