Runtime Models Based on Dynamic Decision Networks: Enhancing the Decision-making in the Domain of Ambient Assisted Living Applications

Luis H. Garcia Paucar, Nelly Bencomo
ALICE, Aston University, UK
Email: garciapl@aston.ac.uk,nelly@acm.org

Kevin Kam Fung Yuen
Xi'an Jiaotong-Liverpool University, China
Email: kevin.yuen@xjtlu.edu.cn

***Abstract*—Dynamic decision-making for self-adaptive systems (SAS) requires the runtime trade-off of multiple non-functional requirements (NFRs) -aka quality properties- and the costs-benefits analysis of the alternative solutions. Usually, it requires the specification of utility preferences for NFRs and decision-making strategies. Traditionally, these preferences have been defined at design-time. In this paper we develop further our ideas on re-assessment of NFRs preferences given new evidence found at runtime and using dynamic decision networks (DDNs) as the runtime abstractions. Our approach use conditional probabilities provided by DDNs, the concepts of Bayesian surprise and Primitive Cognitive Network Process (P-CNP), for the determination of the initial preferences. Specifically, we present a case study in the domain problem of ambient assisted living (AAL). Based on the collection of runtime evidence, our approach allows the identification of unknown situations at the design stage.**

***Index Terms*—Self-adaptation; decision making; AHP; P-CNP; non-functional requirements trade-off; uncertainty**

I. Introduction

Dynamic decision-making is the core function of self-adaptation. Dynamic decision-making requires the runtime quantification and trade-off of multiple non-functional requirements (NFRs) and the cost-benefit analysis of alternative solution strategies. An important research issue has been the specification of the utility function to be used in the decision making process. This utility function includes the utility preferences (aka weights) associated with the NFRs and solution strategies. These preferences may vary from stakeholder to stakeholder and from one envisaged situation to another. Furthermore, different priorities may imply different decisions to be performed by the system. Additionally, in self-adaptive systems (SAS), the assumptions made at design time probably change at runtime causing changes on the defined priorities and therefore on the values for the utility preferences. We argue that modelling and reasoning with prioritization and preferences are the research fields that require further research efforts [1]. Different authors have approached these issues [2], [3], [4], [5], [6]. However, critical challenges are needed to be further explored. One of the issues is that current approaches focus on the design time activities and even if effective they are unlikely to be generalizable [7], [8], [1], [9]. Further, the needs for uncovering relationships between NFRs and updating utility preferences during runtime have been neglected [10], [6]. The steps of monitoring the environment, detecting the need of (self-) adaptation and deciding how to react are the challenges identified in the research area of SAS [2]. We argue that these challenges should involve the role of preferences and the re-prioritization of NFRs due to new evidence found at runtime. The role of runtime models to meet these challenges is crucial we believe [11].

The main contribution of this paper is the combination of conditional probabilities (using Bayesian inference) based on models of DDNs with Bayesian surprises, and Primitive Cognitive Network Process (P-CNP), an improved version of the Analytic Hierarchy Process (AHP) [12], for the determination of the initial preferences, to therefore allow the reassessment of NFRs preferences during runtime. The paper is organized as follows: Section II presents the background on P-CNP, DDNs and Bayesian Surprise where a back-review of related work is provided and the research gap is identified. In Section III, preliminary results that fills the identified research gap are shown and discussed. In Section IV we explaine the background of the domain problem and case study. In Section V we show and explain the experiments performed. Finally, in Section VI, we conclude with respect to our findings, and identify and discuss future research work.

II. Background

This section briefly overviews different Multi Criteria Decision Analysis Methods (MCDA), DDNs models and Bayesian Surprises. We briefly explain how they are relevant to runtime decision-making in SASs.

A. MCDA in SAS

When we make decisions, a natural approach is to evaluate our different alternatives and choose the best one(s) with respect to some given criteria. In SAS we must build intelligent systems being able to apply this way of reasoning to deal with environmental uncertain conditions. How to ensure a reliable decision trading off multiple factors being constantly affected by external changing conditions is the field of action of a well known set of methods, including Multi Criteria Decision Analysis Methods (MCDA) [14]. MCDA methods are currently applied in different fields and especially in self-adaptation. Different MCDA techniques are used for both, decision-making and preferences specification in SAS. Some MCDA approaches such as Primitive Cognitive Network Process (P-CNP) [15], [16] are used for the specification of quality attribute preferences (i.e. NFRs) and some others such as Analytic Hierarchical Process (AHP) [12] are used for specifiying quality attribute preferences and reasoning at runtime based on the prioritization of a set of alternatives decisions. For example in [17], Pimentel et.al. have implemented a routing protocol by using AHP at runtime for video dissemination over Flying Ad-Hoc Network. The approach takes into account multiple types of NFRs such as link quality, residual energy, buffer state, as well as geographic information and node mobility in a 3D space. It uses Bayesian networks and AHP to adjust the NFRs priorities based on instantaneous values obtained during system operation.

As an ideal alternative of AHP, P-CNP replaces the AHP paired ratio scale and performs paired comparison by using a paired differential scale [18]: $b_{ij} = v_i - v_j$. b_{ij} represents the result of paired differential comparisons between alternatives values v_i and v_j. For example, in Table I, row 1, the comparison between alternatives values v_1 (i.e., v_i) and v_2 (i.e.,v_j) will be represented as 3 = v_1 - v_2.

Paired differential scales and the use of pairwise opposite matrices (POM) [15], [19], [16], [20] are the foundations of P-CNP allowing a more precise and natural representation of stakeholders' perception of paired comparison [20]. P-CNP is our selected approach for the determination of the initial preferences of the case study, which involves the following steps:

- Problem cognition process: the idea is to formulate a decision problem as a measurable Structural Assessment Network (SAN) model. Fig. 1 shows a SAN with its main elements: the goal (aka functional requirement), a criteria structure (i.e., NFRs) and a set of alternatives A_n.
- Weight assessment and quality assessment with respect to criteria: The Weight assessment is performed by using differential pairwise comparisons for the criteria Minimize Energy Cost (MEC) and Maximize Reliability (MR) (see Fig. 1). The quality assessment is performed by using differential pairwise comparison between alternatives A_n and each criterion. In Table I it is shown an assessment form for comparison between MEC criterion and alternatives $A_1 ... A_8$.
- Cognitive prioritization process: The idea is to compute the priority, v_i, of each alternative A_i. The Row Average plus the Normal Utility (RAU) priorization method is used to derive the priority values from POM [15]. As a common practice the values are re-escaled to [0,1]. In Table II, it is shown the vector of the normalized values:
  0.1633,0.1394,0.1051,0.0919,
  0.1622,0.1304,0.1215,0.0662
  These values will be used as a input for the Utility Node U of a runtime model based on DDNs explained in section II-B. (See Fig. 2).

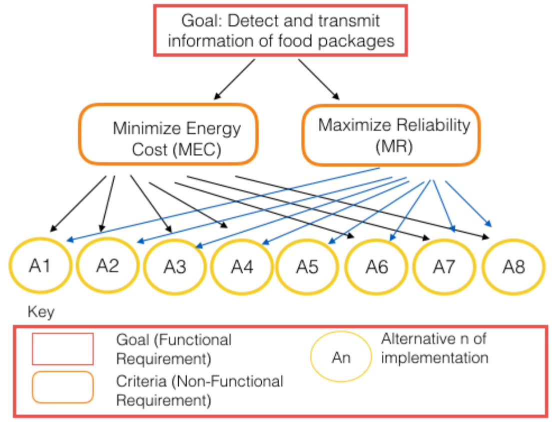

Fig. 1. Structural Assessment Network (SAN) [20]

Q2. The objective of this question is to evaluate the methods for detection and collection of food packages information with respect to Minimize energy consumption (MEC). Eight alternatives are proposed and showed as follow.																		
Criteria	Absolutely (8)	Outstandingly (7)	Signifficantly (6)	Strongly (5)	Highly (4)	Fairly (3)	Moderately (2)	Slightly (1)	Equally (0)	Slightly (-1)	Moderately (-2)	Fairly (-3)	Highly (-4)	Strongly (-5)	Signifficantly (-6)	Outstandingly (-7)	Absolutely (-8)	Criteria
A1						3												A2
A1					4													A3
A1		7																A4
A1												-3						A5
A1												-3						A6
A1				5														A7
A1	8																	A8
A2					4													A3
A2				5														A4
A2								1										A5
A2												-3						A6
A2				5														A7
A2		7																A8
A3							2											A4
A3										-1								A5
A3										-1								A6
A3												-3						A7
A3									0									A8
A4														-5				A5
A4										-1								A6
A4								1										A7
A4								1										A8
A5						3												A6
A5					4													A7
A5				5														A8
A6					4													A7
A6				5														A8
A7							2											A8

TABLE I
P-CNP ALTERNATIVE IMPORTANCE ASSESSMENT FORM

B. DDNs Model for Decision-Making in SAS

We have shown in [21], [22] how dynamic-decision networks (DDNs) offers abstractions that serve the purpose of modelling beliefs about the world, linking preferences and observation models (to obtain evidence from the operational environment at runtime) 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 [21], [22]. They are abstractions for reasoning about the world over time [23]. DDNs provide a set of random variables that represent the NFRs. Fig. 2 shows a DDN during several time slices where Xi denotes a set of state variables, which are unobservable, and E denotes the observable evidence variables. A DDN links decision maker preferences U (i.e. utility nodes), state and evidence variables to make informed decisions D (i.e. decision nodes).

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

Alternative	Detection Strategy	MEC	MR	Weight
A1	FD	T	T	0.1633
A2	FD	T	F	0.1394
A3	FD	F	T	0.1051
A4	FD	F	F	0.0919
A5	SD	T	T	0.1622
A6	SD	T	F	0.1304
A7	SD	F	T	0.1215
A8	SD	F	F	0.0862

Key

FD = Flexible Detection MEC = Minimize Energy Cost
SD = Strict Detection MR = Maximize Reliability

TABLE II
ALTERNATIVE PREFERENCES TABLE

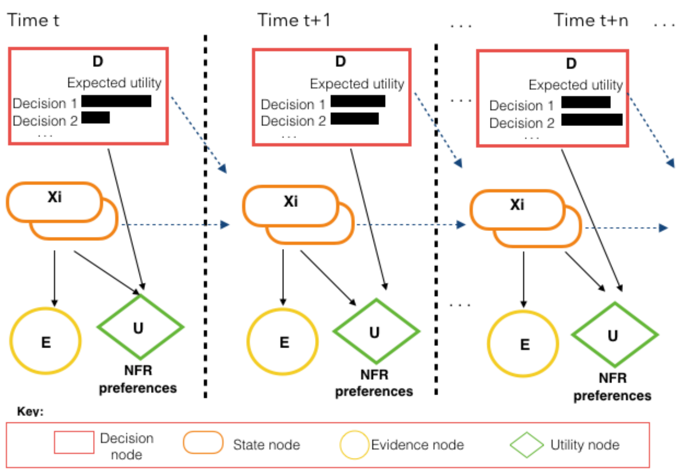

Fig. 2. Example of DDN Structure

as follows:

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

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

C. *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 affect the models or assumptions of the world during runtime [24]. The surprise S represents the divergence between the prior and posterior distributions of a NFR and is calculated by using the Kullback-Leibler divergence (KL) [25]. Lets 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 satisficed and P($NFR_i|E$) is the posterior probability of the NFR_i being partially satisficed given the evidence E.

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

D. *Research Gap*

In [26] we show that, even if scarce, there have been important research efforts towards decision-making for SAS taking into account NFRs. However, relevant results about dynamic reassessment and update of utility preferences are still challenges. The approaches studied show that different MCDA techniques stand out as common techniques used for reasoning optimization [8], [27]. Some approaches use ad-hoc methods for collecting users' preferences, while others use techniques such as MCDA [8], [7], [27]. In [7], [9], [28] the support for preferences update exists but requires user intervention. Some approaches offer potential to support autonomic preference updating. For example, [29] proposed an approach for mining users' behaviour while [27] used an autonomic preference tuning algorithm. [28] and [21] highlighted the relevance of using models that are needed to be learned and refined at runtime during the operation of the system. By using an MCDA technique (i.e., P-CNP) and a runtime model which involves DDNs and Bayesian Surprises, we are contributing to fill the identified research gap with a method for the reassessment of NFRs given new evidence found at runtime.

III. PROPOSAL

A. *Towards Reassessment of Utility Preferences*

Bayesian surprises have been exploited during runtime to improve better informed decision-making at runtime [30]. The approach supports the quantification of uncertainty over different time slices at runtime and helps the system to improve its behaviour based on the basis of learning during the operation of the system. This learning process has shown to be memory-intensive and therefore has presented scalability and memory issues in the past [22]. 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 executed environmental properties which have highest and possible unknown effects at design time on the satisfaction of the NFRs.

The method involves the following steps:

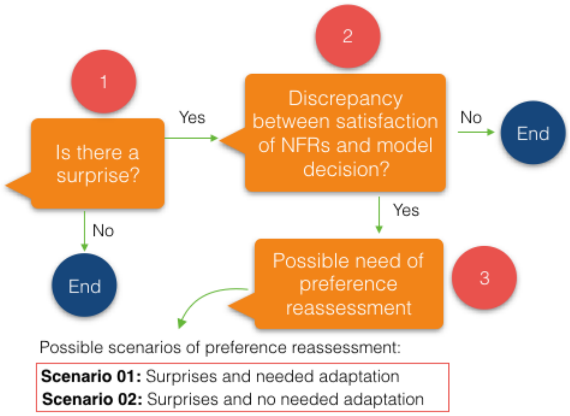

Fig. 3. Approach for preference reassessment at runtime

- 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 be adapted 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. 3 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. Ambient Assisted Living (AAL)

We conducted a case study originally provided by Fraunhofer IESE [1]. It was partially developed further during the execution of the RELAX research work shown in [31].

The case study is related to Mary, an elderly person who can benefit from an Ambient Assisted Living (AAL). Mary is a widow who is 65 years old, overweight and has high blood pressure and cholesterol levels. Mary will be provided with a new AAL system that offers an intelligent fridge. The fridge comes with 4 temperature and 2 humidity sensors and is able to read, store, and communicate RFID information on food packages. The fridge communicates with the AAL system in the house and embed itself in the system. Specifically, the intelligent fridge can detect the presence of spoiled food and discover and receive a diet plan to be monitored on the basis of what food items Mary consumes. The intelligent fridge also contributes to an important part of Mary's diet which is to ensure a minimum liquid intake. A complete description of the case study is shown in [31].

[1]http://www.iese.fraunhofer.de/en//press/press_archive/press_2012 /PM_2012_16_200912_optimaal.html

A specification of the requirements of the AAL at different levels has been extracted from the initial description in the document referenced above [31]. At the highest level, there is an implicit goal of keeping Mary healthy. The goal of the AAL is therefore: "The system SHALL monitor Mary's health and SHALL notify emergency services in case of emergency." Different subgoals (i.e., functional requirements) have also been identified.

- R1.1: The fridge SHALL detect and communicate with food packages.
- R1.2: The fridge SHALL monitor and adjust the diet plan.
- R1.3: The system SHALL ensure a minimum of liquid intake.

Further, softgoals (i.e. NFRs) have also been identified. For example:

- R1.4: The system SHALL minimize energy consumption during normal operation.
- R1.5: The system SHALL maximize reliability during normal operation.

Let us focus on R.1.1. For this functional requirement we have identified two realization strategies:

- Strict Detection (SD): it implies using all the available sensors and the computational resources available to process and fuse the collected sensor data. The fridge will be able to maximise detection of the number of food packages and collation of information about those food packages.
- Flexible Detection (FD): it implies that the system should be able to tolerate incomplete information about food packages. It will require techniques to deal with uncertainty and the identification of a range of suitable sensor types to monitor the food in the fridge.

This case study is implemented in a runtime model taking into account the requirements R1.1, R1.4 and R1.5, specially identifying at runtime the need of preference reassessment of the NFRs R1.4 and R1.5. It will be part of our future work the inclusion of the following NFR: R1.6 The system SHALL minimize latency when an alarm has been raised.

V. Experiments

The experiments are based on the application of our approach to the case study of an Ambient Assisted Living (AAL) application. The AAL system is an smart home for assisted living of elderly people and rely on adaptivity to work properly [31]. AAL can be configured in different ways, for example in terms of detecting and transmiting information of food packages, flexible detection (FD) vs. strict detection (SD), in terms of monitoring and adjusting diet plans or in terms of ensuring a minimum of liquid intake.

This research focuses on the detecting and transmitting information of food packages. Different strategies can be used to implement this requirement and offer different costs and benefits that would need to be traded-off. A SD strategy offers a higher level of reliability than an FD strategy. However, the energy consumption of sensors and computational techniques related to this strategy may be prohibitive. An assessment of

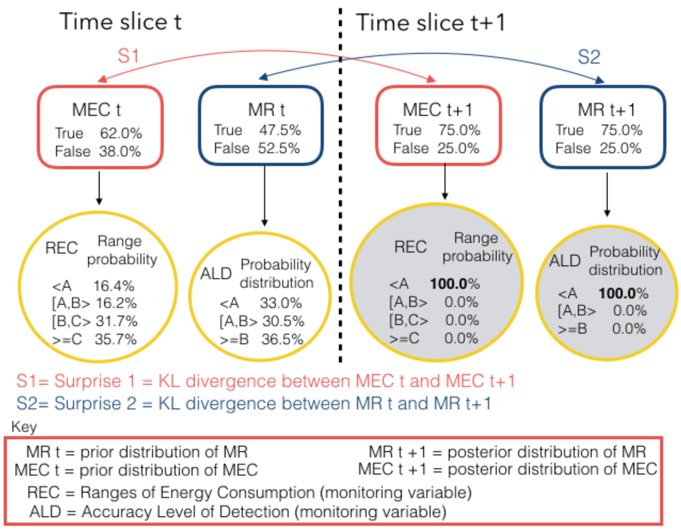

Fig. 4. Example of Computing Surprises - Exp.01 and Exp. 02

the trade-off between these two choices and the satisfaction levels of related NFRs need to be made at design-time and revisited at runtime under the light of new evidence found (See Table II).

A. Initial Setup of Experiments

For the experiments of this paper, a DDN for the application of AAL has been designed according to two alternatives for food packages detection: SD and FD as described above. Each configuration provides different levels of reliability and energy costs which are the NFRs Maximize Reliability (MR) and Minimize Energy Consumption (MEC).
Fig. 5 shows as an example, a DDN for the NFR Minimize Energy Consumption.

The scenario that has been used to perform the experiments, based on information provided by the system's experts, is described as follows: the states of two monitored variables REC="Ranges of Energy Consumption" and ALD="Accuracy Level of Detection" are monitored during runtime. The value of ALD can be three different ranges represented by $ALD < A$, ALD in [A,B>, and ALD>=B. The values for REC are different possible ranges represented by the following expressions: $REC < A$, REC in [A,B>, REC in [B,C>, and REC>=C. At design time, ALD have been considered >=B and REC >=C.

In order to evaluate the DDN shown in Fig. 5, we have considered the following initial conditional probabilities provided by the System's stakeholders:

- P($MEC = true|FD$)=0.55,
- P($MEC = false|FD$)=0.45,
- P($MEC = true|SD$)=0.48,
- P($MEC = false|SD$)=0.52,
- P($MR = true|FD$)=0.49,
- P($MR = false|FD$)=0.51,
- P($MR = true|SD$)=0.55,
- P($MR = false|SD$)=0.45,
- P($ALD < A|MR = true$)=0.15,
- P($ALDin[A, B > |MR = true$)=0.35,
- P($ALD >= B|MR = true$)=0.50
- P($REC < A|MEC = true$)=0.48,
- P($RECin[A, B > |MEC = true$)=0.38,
- P($RECin[B, C > |MEC = true$)=0.08,
- P($REC >= C|MEC = true$)=0.06

The weights associated with the possible combination of nodes are given in Table II. These weights express the preferences that represent the relative importance of each combination of effects of the detection strategy used on the NFRs. For this case study there is a preference for the detection strategy SD. For example, the 3rd row in Table II has a weight value (0.1051) and the 7th row has a weight value (0.1215). Both alternatives have equivalent effect on the two NFRs Minimize Energy Cost and Maximize Reliability (see the values T and F for the two NFRs), however the alternative related to the strategy SD is the most preferred.

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(MEC_t) and P(MR_t) and the posterior models when an evidence has been observed over the time are P($MEC_{t+1}|REC$) and P($MR_{t+1}|ALD$) (see Fig. 4). We have computed surprises based on the KL-divergence between the prior and the posterior probabilities during 13 time slices.

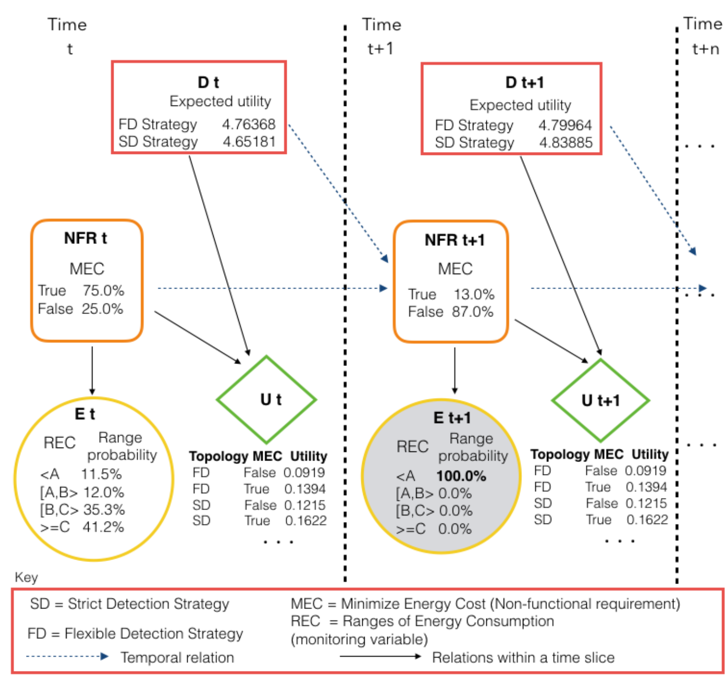

Fig. 5. Example of DDN for AAL System

B. Experiment 1

Surprises take place in several time slices where different specific situations have been identified. Fig. 8 shows the observed values for REC and ALD variables and the surprises S1 and S2. S1 and S2 are the divergence between the prior and posterior distributions for the non-functional requirements MEC and MR respectively. Both, S1 and S2, are computed

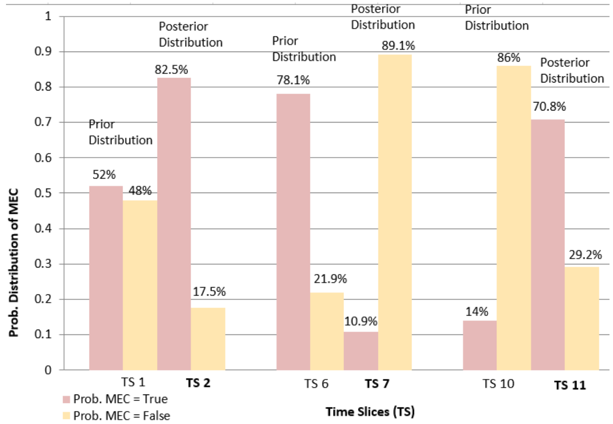

Fig. 6. Prob. distribution of NFR Minimize Energy Cost - Exp. 1

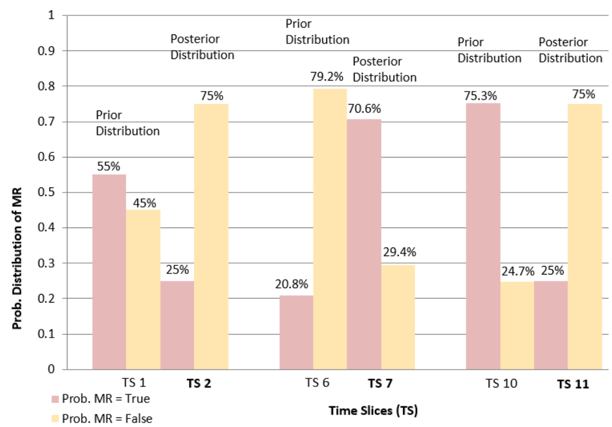

Fig. 7. Prob. distribution of NFR Maximize Reliability - Exp. 1

for each time slice during the experiment.

1) Surprises and adaptation: In time slice 2 we can observe two surprises and an adaptation that is suggested by the DDN (see Fig. 8, column adaptation). Studying the conditional probabilities provided by the DDN under the current conditions:

- P($MEC = true|REC < A, ALD < A$) = 82.5 % (see Fig. 6, time slice 2) and
- P($MR = true|REC < A, ALD < A$)=25% (see Fig. 7, time slice 2)

We can observe that while the probability for Minimize Energy Cost is high the probability for Maximize Reliability is low. The selected choice, i.e. to adapt from SD to FD, certainly sounds like a good selection given the current situation: high probability for Minimize Energy Cost and low probability for Maximize Reliability. Using FD would avoid unnecessary energy costs as the complementary information provided by the conditional probabilities suggest to use the less costly strategy FD. 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. 8). Studying the conditional probabilities provided by the DDN under the current conditions:

Time slice (TS)	Adaptation	S1	S2	REC Monitored values	ALD Monitored values	Current Strategy
1	-	0	0	REC >= C	ALD>B	SD
2	Yes	1.2365166898782605	1.0191044689241149	REC<A	ALD<A	FD
3		0.0	0.0	REC<A	ALD<A	FD
4		0.0	0.0	REC<A	ALD<A	FD
5		0.0	0.0	REC<A	ALD<A	FD
6		0.0	0.0	REC<A	ALD<A	FD
7	Yes	3.1609842539689894	5.537904333006592	REC>=C	ALD>=B	SD
8		0.0	0.0	REC>=C	ALD>=B	SD
9		0.0	0.0	REC>=C	ALD>=B	SD
10		0.0	0.0	REC>=C	ALD>=B	SD
11	-	2.357325286843151	9.44129286534432	REC in [A,B>	ALD<A	SD
12		0.0	0.0	REC in [A,B>	ALD<A	SD
13		0.0	0.0	REC in [A,B>	ALD<A	SD

Fig. 8. Surprises and monitored values - Exp. 1

- P($MEC = true|REC >= C, ALD >= B$) = 10.9% (see Fig. 6, time slice 7) and
- P($MR = true|REC >= C, ALD >= B$)=70.6%, (see Fig. 7, time slice 7)

We can observe that the probability for Minimize Enery Cost is low, however on the other hand, the probability for Maximize Reliability is high. The selected choice, i.e. to adapt from FD to SD, certainly may be a good selection for the current situation: low probability for Minimize Energy Cost and high probability for Maximize Reliability. The complementary information provided by the conditional probabilities suggest to use the stragegy FD. The surprises and the conditional probabilities help us in flaggingup 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. 8). Studying the conditional probabilities provided by the DDN under the current conditions:

- P($MEC = true|RECin[A, B >, ALD < A$) = 70.8% (see Fig. 6, time slice 11) and
- P($MR = true|RECin[A, B >, ALD < A$)=25.0% (see Fig. 7, time slice 11)

We can observe that the probability for Minimize Energy Cost is high. However, on the other hand, the probability for Maximize Reliability is low. The selected choice, i.e. not to adapt, certainly may not be the best choice given the current situation: high probability for Minimize Energy Cost and low probability for Maximize Reliability. Continuing using SD as the configuration would create unnecessary energy costs as the complementary information provided by the conditional probabilities suggest the use of the less costly strategy FD. 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

Time slice (TS)	Adaptation	S1	S2	REC Monitored values	ALD Monitored values	Current Strategy
1	-	0	0	REC >= C	ALD>B	SD
2	Yes	0.2946091496255637	0.26834831468717085	REC<A	ALD<A	FD
3	-	0.13049000186790083	0.007067319243155296	REC in [A,B>	ALD<A	SD
4		0.0	0.0	REC in [A,B>	ALD<A	SD
5		0.0	0.0	REC in [A,B>	ALD<A	SD
6	-	0.7566142417762437	0.8020257430291649	REC in [B,C>	ALD>=B	SD
7		0.0	0.0	REC in [B,C>	ALD>=B	SD
8		0.0	0.0	REC in [B,C>	ALD>=B	SD
9		0.0	0.0	REC in [B,C>	ALD>=B	SD
10	-	0.8092149004836475	0.8041042228464641	REC in [A,B>	ALD<A	SD
11		0.0	0.0	REC in [A,B>	ALD<A	SD
12		0.0	0.0	REC in [A,B>	ALD<A	SD
13		0.0	0.0	REC in [A,B>	ALD<A	SD

Fig. 9. Surprises and monitored values - Exp. 2

of how surprises and the conditional probabilities of the DDN can flag up the need of adaptation. Crucially, the situation detected implies the need to revisit the preferences defined by the stakeholders previously providing the opportunity to improve the behaviour of the system.

C. Experiment 2

The observed values for REC and ALD variables and the surprises S1 and S2 are shown in Fig. 9.

1) Surprises and adaptation: In time slice 2 we can observe surprises and that an adaptation is suggested by the DDN (see Fig. 9). Studying the conditional probabilities provided by the DDN under the current conditions:

- P($MEC = true|REC < A, ALD < A$) = 82.5% (see Fig. 10, time slice 2) and
- P($MR = true|REC < A, ALD < A$)=25.0% (see Fig. 11, time slice 2)

We can observe that the probability for Minimize Energy Cost is high. On the other hand, the probability for Maximize Reliability is low. The selected choice, i.e. to adapt from SD to FD, certainly looks to be a good selection given the current situation: high probability for Minimize Energy Cost and low probability for Maximize Reliability. Crucially, the complementary information provided by the conditional probabilities suggest to use the strategy FD. The surprises and the conditional probabilities help us in identifying this situation. The situation is therefore an example of agreement 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. 9). Studying the conditional probabilities provided by the DDN under the current conditions:

- P($MEC = true|RECin[A, B >, ALD < A$) = 64.7% (see Fig. 10, time slice 3) and
- P($MR = true|RECin[A, B >, ALD < A$)=20.8% (see Fig. 11, time slice 3)

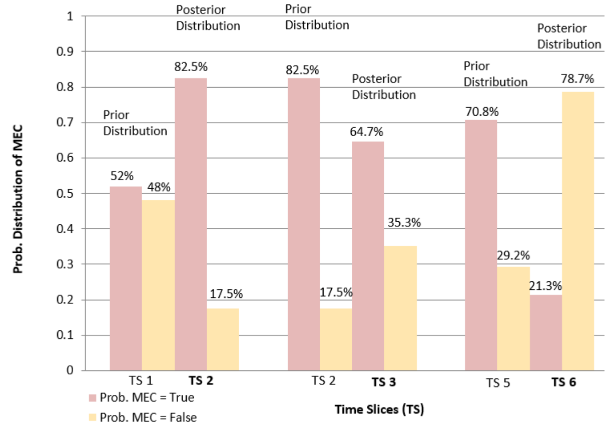

Fig. 10. Prob. distribution of NFR Minimize Energy Cost - Exp. 2

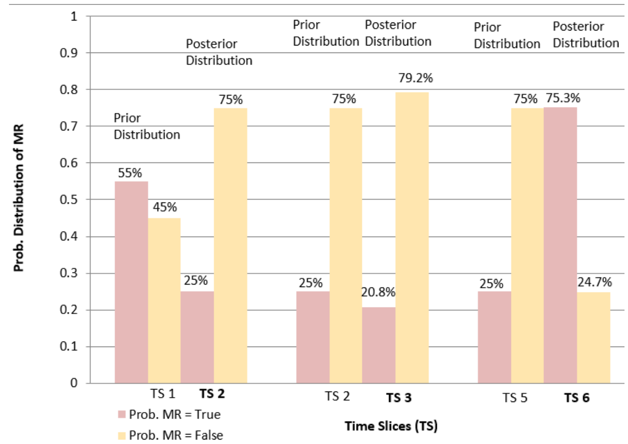

Fig. 11. Prob. distribution of NFR Maximize Reliability - Exp. 2

We can see that the probability for Minimize Energy Cost is high. On the other hand, the probability for Maximize Reliability is low. The selected choice, i.e. to adapt, certainly may not be a good selection for the current situation: high probability for Minimize Energy Cost and low probability for Maxmize Reliability. Using SD would create unnecessary energy costs as the complementary information provided by the conditional probabilities suggest to use the less costly strategy FD. The surprises and the conditional probabilities supported flagging up the situation. The situation is an example how surprises and conditional probabilities can highlight the needs of avoiding unnecessary adaptations. The previous findings imply the needs to reasses 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. 9). Studying the conditional probabilities provided by the DDN under the current conditions:

- P($MEC = true|RECin[B, C >, ALD >= B$) = 21.3% (see Fig. 10, time slice 6) and
- P($MR = true|RECin[B, C >, ALD >= B$)=75.3% (see Fig. 11, time slice 6)

We can see that the probability for Minimize Energy Cost is low. On the other hand, the probability for Maximize Reliability is high. The selected choice, i.e. not to adapt, certainly looks to be a good selection for the current situation: low probability for Minimize Energy Cost and high probability for Maximize Reliability. The complementary information provided by the conditional probabilities suggests that using SD is a better option than using FD. 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 10 there are surprises however, the DDN has not suggested any adaptation (see Fig. 9). 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.

D. Analysis of Results

Using our approach we have been able to identify four (4) scenarios that allows 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.

Scenarios 01 and 02 have been identified to flag up the need for revisiting the NFRs preferences defined by the stakeholders previously using an MCDM method (i.e. P-CNP) 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 [32].

It was possible to explore all these scenarios only by using surprises and Bayesian inference (conditional probabilities) at runtime. 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 [24], [22] [33], 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.

VI. Conclusions

In this paper we have used a better alternative of AHP, P-CNP, for the definition of preferences at design time and have shown its integration to our DDN-based approach. The approach can be used for preference updating at runtime. P-CNP method will provide a structured technique for runtime decision-making problems with multiple criteria (i.e., NFRs) by doing pairwise comparison during system operation between numerical values collected from sensors related to NFRs and their relative importance to adjust preferences at runtime.

The experiments performed required the setting of the utility preferences associated with NFRs. Those preferences were initially provided by the domain experts during the sensitivity analysis at design time. However, the experiments performed demonstrate how these utility preferences, even if meeting specific requirements identified at design time, may not be ideal for specific cases to be found at runtime. When preferences do not agree with specific situations identified at runtime and unknown at design time, the system may either suggest unnecessary adaptations or miss adaptations. These situations can potentially degrade the behaviour of the running system. The obtained results confirm the validity of our approach defined in our previous work [26]. Currently, to our knowledge, there is no related work to this specific issue in SASs.

Our approach takes advantage of Bayesian learning to collect evidence to improve the understanding of the environment and the decision making process by the running system. Furthermore, we have shown the power of runtime abstractions based on runtime DDN-based models to allow the better understanding of contexts that were not fully captured during the requirements elicitation. Challenges for future work still remain, specifically we are working on how to optimize and scale reasoning techniques to perform dynamic updating of NFR preferences when non-appropriate NFR preferences have been identified. The use of machine learning techniques and bayesian surprise for NFRs preferences learning and NFRs relaxation respectively may be a promissory path in our future work.

Acknowledgment

The research work reported in this paper is partially supported by Research Grants from National Natural Science Foundation of China (Project Number 61503306) and Natural Science Foundation of Jiangsu Province (Project Number BK20150377), China.

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] 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

[8] 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.

[9] H. Song, S. Barrett, A. Clarke, and S. Clarke, "Self-adaptation with end-user preference: Using run-time models and constraint solving," in *the Intrl. Conference MODELS*, USA, 09/2013 2013.

[10] N. Bencomo, "Quantun: Quantification of uncertainty for the reassessment of requirements," in *23rd IEEE International Requirements Engineering Conference, RE 2015, Ottawa, ON, Canada, August 24-28, 2015*, 2015, pp. 236–240.

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

[12] T. Saaty, "Decision making with the analytic hierarchy process," *Inter. Journal of Services Sciences,*, 2008.

[13] 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

[14] J. Figueira, S. Greco, and M. Ehrogott, *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer, 2005.

[15] K. K. F. Yuen and W. Wang, "Towards a ranking approach for sensor services using primitive cognitive network process," *4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2014*, pp. 344–348, 2014.

[16] K. Yuen, "Cognitive network process with fuzzy soft computing technique in collective decision aiding," *The Hong Kong Polytechnic University, PhD thesis*, 2009.

[17] L. Pimentel, D. Ros, and M. Seruffo, "Wired/Wireless Internet Communications," vol. 8458, pp. 122–135, 2014. [Online]. Available: http://link.springer.com/10.1007/978-3-319-13174-0

[18] K. K. F. Yuen, "Software-as-a-Service evaluation in cloud paradigm: Primitive cognitive network process approach," *2012 IEEE International Conference on Signal Processing, Communications and Computing, ICSPCC 2012*, pp. 119–124, 2012.

[19] Y. K.K.F., "The pairwise opposite matrix and its cognitive prioritization operators: the ideal alternatives of the pairwise reciprocal matrix and analytic prioritization operators," *Journal of the Operational Research Society*, no. 63, pp. 322–338, 2012.

[20] K. K. F. Yuen, "The Primitive Cognitive Network Process in healthcare and medical decision making: Comparisons with the Analytic Hierarchy Process," *Applied Soft Computing Journal*, vol. 14, no. PART A, pp. 109–119, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2013.06.028

[21] 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.

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

[23] 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.

[24] 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.

[25] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.

[26] L. H. G. Paucar and N. Bencomo, "The Reassessment of Preferences of Non-Functional Requirements for Better Informed Decision-making in Self-Adaptation," *AIRE - 3rd International Workshop on Artificial Intelligence for Requirements Engineering*, 2016.

[27] 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.

[28] 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.

[29] 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

[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] 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.

[32] 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.

[33] 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.