DeSiRE: Further Understanding Nuances of Degrees of Satisfaction of Non-functional Requirements Trade-off

Ross Edwards
SARI, ALICE, Aston Univeristy
Birmingham, UK
edwarrj2@aston.ac.uk

Nelly Bencomo
SARI, ALICE, Aston Univeristy
Birmingham, UK
nelly@acm.org

ABSTRACT

[Context/Motivation] Self-adaptive systems (SAS) are being deployed in environments of increasing uncertainty, in which they must adapt reconfiguring themselves in such a way as to continuously fulfil multiple objectives according to changes in the environment. The trade-offs between a system's non-functional requirements (NFRs) need to be done to maximise a system's utility (or equity) with regards to the NFRs, and are key drivers of the adaptation process. Decision-making for multiple objective scenarios frequently uses utility functions as measures of satisfaction of both individual and sets of NFRs, usually resulting in a weighted sum of the different objectives. [Questions/Problems] However, while adaptations are performed autonomously, the methods for choosing an adaptation are based on the criteria of human expert(s), who are susceptible to bias, subjectivity and/or lack of quantitativeness in their judgements. Thus, there is a need for a non-subjective and quantitative approach to reason about NFR satisfaction in multi-objective self-adaptation without relying on human expertise. Furthermore, human biases can also apply to the relationships between two or more NFRs (e.g. how much the satisfaction of one NFR affects the satisfaction of another), resulting in emergent inaccuracies affecting the decision(s) chosen. [Principal ideas/ results] This paper presents DeSiRE (Degrees of Satisfaction of NFRs), a purely automated objective statistical approach to quantifying the extent that a requirement is violated or satisfied, and its application to further explore the trade-offs between NFRs in decision making. Experiments using case studies have positive results showing the identification of a Pareto optimal set of candidate solutions, in addition to a ranking of these configurations by their satisfaction of each NFR.

KEYWORDS

non-functional requirements, self-adaptative systems, multi-criteria decison making

ACM Reference Format:
Ross Edwards and Nelly Bencomo. 2018. DeSiRE: Further Understanding Nuances of Degrees of Satisfaction of Non-functional Requirements Trade-off. In *SEAMS '18: SEAMS '18: 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems , May 28–29, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3194133.3194142

SEAMS '18, May 28–29, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5715-9/18/05...$15.00
https://doi.org/10.1145/3194133.3194142

1 INTRODUCTION

Self-adaptation underpins software capabilities that allow the system to adapt according to a volatile and stochastic environment [8]. With self-adaptation, systems can change their behaviour at runtime by reconfiguring; while keeping NFRs associated with systems properties satisfied. Optimising trade-offs between NFRs, Quality of Service (QoS) attributes, etc. is a key factor in the adaptation process [25].

Some initiatives exist for automating the decision making based on the trade-off of the levels of satisfaction of NFRs. However, the process of defining the exact degree to which an NFR is satisfied or violated is currently mainly driven by human experts and the software specifications based on their expertise knowledge [5, 14], as well as the degree to which individual objectives are in tradeoff with each other [10]. Using human experts brings with it the problem of their judgements being prone to subjectivity, uncertainty, vagueness and variation from person to person, for example the amount of utility gained (or lost) from an adaptation. These problems can lead to requirements that are unrealistic in terms of satisfaction threshold, and an inaccurate depiction of the degree to which two or more NFRs are in tradeoff with each other, especially if the NFRs refer to different system properties with different units.

We argue that quantitatively identifying the extent to which a given NFR (or multiple thereof) is satisfied and its effects to the extent of satisfaction of other NFRs (i.e requirements in trade-off) needs to be an explicit part of the decision-making process. This analysis can also provide designers with valuable input for refining the adaptation decisions to balance the satisfaction of individual NFRs and the trade-offs between them. One objective is to provide designers of self-adaptive systems with a basis for further analysis, to revise and improve the understanding of the environment and its effect on NFRs and any decision-making thereafter.

In this paper, we present DeSiRE (Degrees of Satisfaction in Requirements Engineering), a statistical approach for measuring the extent to which a requirement is satisfied (or violated), its application in exploring the possible trade-offs between NFRs when choosing among configurations of the system to better fulfil a set of NFRs. We also present two case studies: one to show an application of DeSiRE in multicriteria decision making (MCDM) and one to show an application of DeSiRE in the live monitoring of NFR satisfaction.

The organization of this paper is as follows: Section 2 gives the background to our work; Section 3 provides the process of applying DeSiRE to NFRs and system properties and the purposes of the outputted value *Extent of Satisfaction (ExS)*; Section 4 describes experiments with exemplar systems applying our proposed approach.

Section 5 discusses related work and finally, Section 6 draws conclusions and gives possible directions for future research.

2 BACKGROUND

Methods of requirements specifications for decision-making have focused on identifying if a given requirement has been violated or satisfied as a boolean-style measure, as opposed to a continuous measure giving the extent of satisfaction (or violation). Many publications [9, 12, 17, 25, 26, 32] have shown different uses of the NFR Framework to explore alternatives for adaptable software applications. When using the NFR Framework to model NFRs as soft goals, an analyst explores alternatives to meet the soft goal and chooses the valid alternatives by tagging them as satisfied. The label of each selected soft goal is evaluated to determine how it impacts the achievement of its parent goal. Authors have used discrete, qualitative codes and notations for describing impact upon a goal (e.g. makes, helps, hurts, breaks, etc.).

Attempts to provide flexibility to NFR assessment include the RELAX requirements language [35] and its application to Claims [22]. RELAX was designed to aid program design by reasoning about how well a system is doing with respect to target behaviours making emphasis on how to tolerate the fact that a given NFR could be slightly violated but just temporarily, in order to avoid costly adaptations. RELAX uses fuzzy sets [36] with boundaries predefined by experts to quantify deviations from expected behaviour as specified by an NFR. In this case, the satisfaction of an NFR ranges from zero (fully violated) to one (fully satisfied) with intermediate values representing partial satisfaction/violation.

In the domain of trade-off between system properties using NFR specifications, Talon et al [27] proposed a solution in the domain of service-based SAS. Torres et al [28] is a similar approach, but also allows for individual users to tailor decision making to their priorities. [21] and [11] both cover requirements trade-off, through using evolutionary computation to generate "adaptation paths" that balance NFR trade-offs while minimising reconfiguration costs, and devising a heuristic algorithm to produce alternative design solution in the absence of numerical data respectively. However, these approaches focus on optimisation of one measurable system property as opposed to multiple properties and models their decision-making system on expert-defined trapezoidal fuzzy sets connected to linguistic (i.e. discrete) variables, as opposed to representing the extent of NFR satisfaction/violation as a numerical, continuous measure.

In [7], we studied how self-adaptive systems can assess deviations from its specified behaviour (i.e. negative satisfaction) and use these deviations to trigger adaptation. This deviation is the extent (or quantification) of the violation of an NFR. Different from the methods described above and from DeSiRE, this approach uses probabilities and Bayesian theory to measure the level of satisfaction.

The approaches described above describe NFR satisfaction compared to a pre-defined value (e.g. the probabilities used to calculate surprise in [7]), requiring expert knowledge that may not necessarily be available. Furthermore, each measure has hard limits for "full" satisfaction and violation (i.e. the measure is *bounded* or *clamped*).

We argue that the trade-offs in the decision-making should be able to explicitly and boundlessly measure both the extent of the positive satisfaction of a NFR but also the effect on the negative satisfaction (i.e. violation) carried out on other NFRs. Such assessments can support more accurate analysis about the relationships and interconnections between the levels of satisfaction and violation between NFRs, focusing on questions such as:

- *What is the extra extent of service time or reliability that the system is providing in comparison to the extra extent of costs committed? Is it worth the investment?*
- *What are the potential configurations that allow a (quantified) small sacrifice in the level of service time while providing quantified lower costs?*

We believe that these questions can be approached with Multi-Criteria Decision Analysis [13] methods and techniques via modelling NFRs in trade-off as *cooperative non-zero sum games* with NFR satisfaction modelled as utility (similar to [4]). Belton [6] claims that the these particular games are the *closest branch of game theory to MCDA*. Modelling the satisfaction of an NFR as the utility of a criterion effectively allows us to apply MCDA techniques such as Pareto dominance [18] to eliminate candidate solutions that are *worse in every way* than one or more other solutions.

We consider that there is a clear research gap for reasoning about NFRs using non-bounded quantitative measures of NFR satisfaction and violation, which our proposed approach aims to provide.

3 DESIRE: DEGREES OF SATISFACTION/VIOLATION

In Section 1, we illustrated examples of requirement satisfaction being a measure between two values representing "fully violated" and "fully satisfied" (usually zero and one respectively). By contrast, DeSiRE treats zero as "the boundary between satisfied and violated", with positive values representing a degree (or extent) of satisfaction and negative values a degree (or extent) of violation. Thus, the larger the magnitude of the value, the larger the degree of satisfaction/violation is said to be.

A requirement R_n pertaining to property p has an extent of satisfaction $\mathrm{ExS}(R_n)$ calculated by the following equation:

$$\mathrm{ExS}(R_n) = \frac{\Delta p}{\sigma p}$$

where σp is the standard deviation of the measured property, and Δp is the equation in Table 1 corresponding to the comparator in the NFR, where p_m is the measured value and p_r is the reference value specified by the NFR. The equations in Table 1 are engineered so that an increase in requirement satisfaction results in a increase in ExS, e.g. with a minimisation objective, a **decrease** in the measured value would result in an **increase** in ExS.

DeSiRE operator	Equation
LESS / EARLIER THAN	$\Delta p = p_r - p_m$
MORE / LATER THAN	$\Delta p = p_m - p_r$
AS MANY AS POSSIBLE	$\Delta p = p_r - p_m$
AS FEW AS POSSIBLE	$\Delta p = p_m - p_r$
AS CLOSE AS POSSIBLE TO	$\Delta p = \lvert p_m - p_r \rvert$

Table 1: ExS calculation for each DeSiRE operator

Let's study the need to treat satisfaction of NFRs as a continuous measure with the following example non-functional requirement:

Component C shall perform Action A between once every 23 and once every 24 hours.

Supposing that the time point t is the time after Action A was last performed, then if Component C were to perform Action A at any time before $t+23$ hours or any time after $t+24$ hours, the requirement would be considered "violated", while any time $23hrs \leq t \leq 24hrs$ would consider the requirement "satisfied", without regard to the proximity of t to each boundary.

We believe that the current methods do not take into account the actual value in comparison to the requirement. Given that many properties have a proportional, non-boolean effect on the overall utility of a system, we believe that current methods disregard information that could be potentially useful in the adaptation selection process.

As a result, requirements being satisfied would give a measured value of zero or greater (+), and requirements being violated would give a measured value of less than zero (-). The extent of a requirement being satisfied or violated is proportional to the magnitude of its measure.

As the process of calculating the ExS of an NFR divides a measured property by its standard deviation, and given that both of these are of the same units, ExS provides a dimensionless quantity (i.e. a number without units), and is comparable with any other values of ExS, even if the properties they pertain to are in different units.

It is therefore possible to compare the ExS of different system properties, for example the weighted sum [29] of a set of NFRs by:

$$\text{ExS}(R_{1\ldots n}) = \sum_{i=1}^{n} weight(R_i) \times \text{ExS}(R_i)$$

The limitations of this quantification of goal satisfaction are that in order to properly calculate ExS, the standard deviation of the property must be known or calculable. DeSiRE attempts to alleviate this limitation by maintaining a store of monitored readings for the system property from which its standard deviation can be calculated, and updating the store over time (e.g. by deleting readings from before a certain point in time).

3.1 Comments on min/maximisation objectives

One of the aims of DeSiRE is to allow the comparison of requirement satisfaction/violation among requirements referring to system properties with different mean values and/or distributions. As a result, it is possible for system properties to be equally "minimised" despite their values having different averages, or belonging to different scales and/or units. Any reference point that is constant across all NFRs does not take the variation of mean values into account, and this is reflected in the output ExS values. In cases of a reference point not being given, the mean value of the measured property is used as a default.

3.2 The structure of DeSiRE-compatible requirements

The specification of requirements with DeSiRE is syntactically inspired by the RELAX specification language for requirements [35], however it is not necessarily related to or compatible with the initial aim of RELAX. RELAX has been designed to explicitly address the uncertainty found in self-adaptive systems, giving formal linguistic operators that are mapped to Fuzzy Branching Temporal Logic (FBTL) [16] expressions. These operators serve as the inspiration for comparators in a DESiRE-compatible (or DeSiREable) NFR.

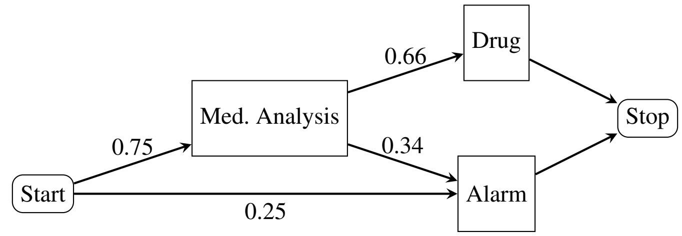

Figure 1: Work flow of a single invocation of TAS, with probability of each service being invoked.

The components of a DeSiREable software requirement consist of a system property to be measured (e.g. mean time between failures, service time, etc.), a comparative word or phrase (e.g. earlier than, more than, as close as possible to) and optionally a reference value to compare measurements to. In lieu of a reference value, DeSiRE will default to the mean value of the system property being measured (for rationale, consult Section 3.1). The DeSiREable requirement may specify alternative states or actions that also lead to its fulfilment. Some examples of comparative phrases and their application to the calculation of DeSiRE are given in Table 2, along with the FBTL expression in [35] that the operator is drawn from.

As can be seen, as the components of a DeSiREable requirement do not insist on specific keywords and do not necessarily have to be in a specific order, it is possible for the requirement text alone to be misleading or ambiguous. To alleviate this, we propose that either each component word/phrase be clearly marked differentiating between type (e.g. by differently-coloured text) or a table similar to the uncertainty factors in [35] specifying what each component is.

The output measure Extent of satisfaction (ExS, pronounced "excess") is defined as the amount of standard deviations that the measured value is "better" than the reference value. In this case, "better" refers to the comparator given in the NFR (see Table 1).

4 CASE STUDIES

4.1 Tele Assistance System (TAS)

Tele Assistance System (TAS) [33] is an exemplar service-based SAS in which a scenario of healthcare assistance is modeled. Consumers choose from a selection of providers for 3 different services: a medical analysis service, a drug service, and an alarm service. A consumer may invoke TAS in two ways: a regular check on the consumer's vitals (medical analysis service) leading to either new medication being delivered (drug service) or an ambulance being called (alarm service), or immediately invoking the alarm service as a panic button. Figure 1 shows the full invocation process along with the probability each service is called.

This case study is the analysis of Danny Weyns' case studies in [34], namely whether judging by the sum of the degree of NFR satisfaction gave a better-informed decision than the elimination-based method used in [34]. Initial findings were promising, as not only were all adaptation options ranked taking into account all NFRs (as opposed to a portion of the adaptation options ranked according

Comparators	Description	Example in DeSiRE	FBTL Exp.
BEFORE / AFTER	Action should be performed a given amount of time before or after a given event	*The heaters shall activate 5 minutes BEFORE the user is due to arrive home.*	$A\chi_{[</>]e_d}\phi$
GREATER / LESS THAN	Monitored value should be higher or lower than given target	*There may be no greater than three failures per month, or an uptime greater than 90%.*	$AF\chi_{[</>]e_q}\phi$
AS CLOSE AS POSSIBLE TO	Monitored value should equal target, or action should be performed at a time closest to target as possible	*The internal temperature shall be as close as possible to 18 degrees C.*	$A((\Delta(\phi) - q) \in S_n)$
AS MANY/FEW AS POSSIBLE	Monitored value should be as greater/less than its mean as possible	*The operation should take as little memory as possible.*	$A((\Delta(\phi) - q) \in S_{[0/\infty]})$

Table 2: Temporal and ordinal operators in the context of DeSiRE

to one NFR), but additional possible configurations were found that were previously overlooked.

Figure 2 shows the 125 configurations in terms of cost against failure rate on the left hand side, with cost against service time on the right. Configurations that are Pareto non-dominated across both objectives depicted by the graph's axes are mark as filled circles, configurations that are Pareto non-dominated across all objectives are shown as solid circles, with configurations Pareto dominated across all objectives being faded out. Note that:

- a configuration being Pareto optimal across any two objectives implies Pareto optimality across all three objectives.
- it is possible for a configuration that is Pareto dominated as depicted in two dimensions to be Pareto optimal when the third objective is included in the reasoning.
- **configurations exist that are Pareto dominated across all possible two-objective combinations, but are Pareto optimal across all three objectives**.

This suggests the possibility that in multi-objective scenarios, reasoning about only a subset of the objectives measured leads to a poorly-informed decision being made, as shown by an incomplete or inaccurate Pareto frontier.

In DesiRE, the statistical process of normalisation is applied to the measured value of each system property, in order to account for the variation in mean, standard deviation and units of the various properties. In all current instances, normalisation results in the ExS measure as discussed in Section 3.

NFR no.	NFR description
R1	$failurerate <= 0.15$
R2	$cost <= 8$
R3	$servicetime <= 80$

Table 3: NFRs used for the case study given in Section 4.1.

4.2 The DeltaIoT Exemplar

Iftikhar et al [15] proposed an exemplar self-adaptive system in the domain of Internet of Things (IoT). The exemplar consists of multiple embedded computers (referred to as "motes") located around the KU Leuven campus communicating with each other with the functional goal of relaying information to a manned control centre ("gateway"), with the locations of the various motes and gateways are based on the KU Leuven campus. Adaptation engines may change the connections between motes dynamically, provided that every mote can reach a gateway through the network.

No.	Objective	Objective as DeSiREable NFR
R1	*The packet loss experienced across the network is to be minimised.*	*Packet loss **shall** be **as low as possible**.*
R2	*The energy consumption for each mote in the network is to be minimised.*	*Energy consumption **shall** be **as low as possible**.*

Table 4: Non-functional objectives for Scenario S2 of DeltaIoT.

A scenario simulating wireless interference and a fluctuating traffic load with the objectives to minimise both packet loss and energy consumption was implemented, and can be said to contain the two non-functional objectives as given in Table 4. Both of the system properties mentioned in the objectives (packet loss and energy consumption) are measured in arbitrary (but not identical) units, and it is assumed that the objectives have equal preference.

4.2.1 Analysis of a Simple Adaptation Engine. An initial case study to apply DeSiRE to DeltaIoT was to analyse the *simple self-adaptation solution* in [15] where compared to a control run without adaptation, power consumption was reduced from a value of 33 to 22 arbitrary units, however packet loss was increased from 0.05 to 0.17 arbitrary units. We set out to measure the utility gained or lost from this adaptation using DeSiRE and concluded that with the assumption that both objectives are given equal preference, the utility gained by decreasing power consumption outweighs the utility lost from a resultant increase in packet loss.

Figures 3 and 4 show the change in satisfaction over time for the two objectives in Scenario S2, and figure 5 shows the difference of satisfaction (i.e. the NFR satisfaction when using adaptation minus the satisfaction when not using adaptation) for both objectives over time. The graph shows that while the second objective (minimise power consumption) is continuously more satisfied when

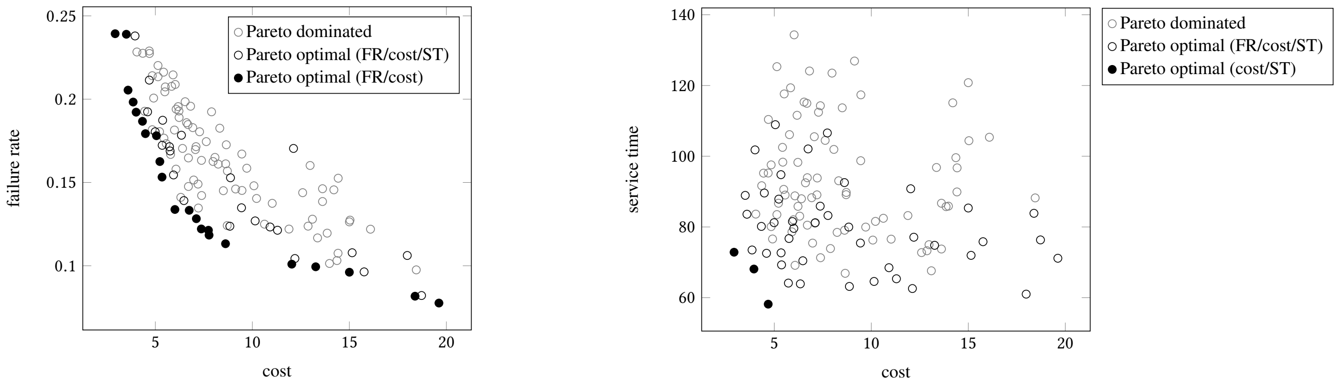

Figure 2: Configurations of TAS, and their non-functional metric performances.

using adaptation by a mean of 1.82 ExS, the first objective (minimise packet loss) is less satisfied by a mean of 1.14 ExS. Thus, according to DeSiRE, we can say that using Iftikhar's adaptation engine has provided a net gain of 0.68 ExS across the two objectives.

Using the process of DeSiRE has allowed us to normalise both the units and deviation of each system property into a measure of how much a given objective is satisfied. By normalising units and spreads, we allow system properties to be comparable with each other, specifically to compare the loss of energy consumption with the gain in packet loss using adaptation, leading us to determine that it is a trade-off that results in a net gain of utility.

4.2.2 Accuracy Loss from a Limited Data Store. After using DeSiRE to give a quantitative measure of the utility gained from adaptation, we then proceeded to test whether DeSiRE could give a meaningful measure of NFR satisfaction in real time. We tested this claim with a scenario where for any given point in the experiment, the DeSiRE monitor could only access the current environment state plus states for the previous t hours (essentially, DeSiRE derives a property's mean and standard deviation from the last t hours as opposed to the entire 24-hour period at once). ExS readings for t values of 6, 12, 18 and 24 hours are given in Figure 6 as dashed lines. The requirement being measured is objective 2 in Table 4.

As shown in Figure 6, the ExS ratings using a limited reading store tends pre-empt the ratings using a full reading store (in that the peaks and troughs in the graph happen at an earlier time), and that the peaks and troughs are of a higher magnitude. However, the general shape of the graph is similar, leading us to believe that runtime monitoring does not result in a loss of information to the extent that runtime monitoring does not give an accurate view of the performance of an SaS with regards to a single objective. We hypothesise that the difference in magnitude between the two graphs is because the full reading store includes both values before and after adaptation and as a result the readings have a higher standard deviation, and therefore a lower magnitude of ExS.

5 RELATED WORK

DeSiRE opens further possibilities of interactions with other domains. In the domain of economics, Weiser [30] formalised this concept as *opportunity cost*, defined today as *the added cost of using resources (as for production or speculative investment) that is the difference between the actual value resulting from such use and that of an alternative (such as another use of the same resources or an investment of equal risk but greater return)*. Applied to DeSiRE, the *opportunity costs* would be the difference in ExS between the taken decision and a given alternative option available.

Authors of [19] have used Partially Observable Markov Decision Process (POMDP) to serve as a framework model to support decision making. We are working on the use of the rewards functions (with positive and negative values) to support an alternative implementation of ExS. The negative values are of special interest to represent the equivalent of opportunity costs.

We also envision DeSiRE being used for or inspiring an implementation of the RELAX requirements language [35], as the ExS value can be used to measure how much an NFR is to be violated (or *RELAXed*) as a fuzzy measure (with some conversion from a ExS value $\in \mathbb{R}$ to a RELAXation value $\in [0, 1]$). DeSiRE could also be linked to self-explanation [31] based on the rationale available about the trade-off implemented (for example, determining a correlation between two NFRs from the candidate solutions measured).

Approaches for automated reasoning about NFR trade-off include [23], focusing on interdependencies among NFRs and proposing a design-time approach to optimising NFR satisfaction using fuzzy logic and decision support systems [20]. Another similar approach is [2, 3], which instead uses a search-based method to elicit design trade-offs and optimise NFR satisfaction. The author previously proposed a model-based approach [1] to determine search spaces in design for feedback controls, which was used to find Pareto optimal configurations within that search space. Sandionigi et al [24] have also proposed an approach to optimise QoS attributes within a service selection problem related to wireless devices.

6 CONCLUDING REMARKS

We argued the case for taking into account the extent or degree of non-functional requirement satisfaction in the balancing of requirements trade-off in self-adaptive systems. We introduced DeSiRE for measuring the degree of satisfaction or violation for NFRs, and guidelines to writing NFRs that are compatible with DeSiRE. We also provided the quantitative measure "Extent of Satisfaction" (ExS) used in this process.

The ExS of a single NFR was defined as the difference between a measured value of a system property and the reference value specified by the NFR specification, divided by the standard deviation of the property. A first proposal for a global measure of the ExS of a

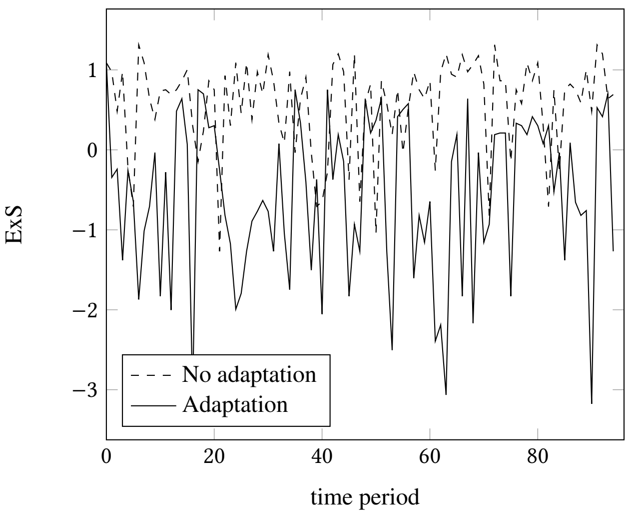

Figure 3: Satisfaction of objective 1 (minimise packet loss) over 15-minute time periods.

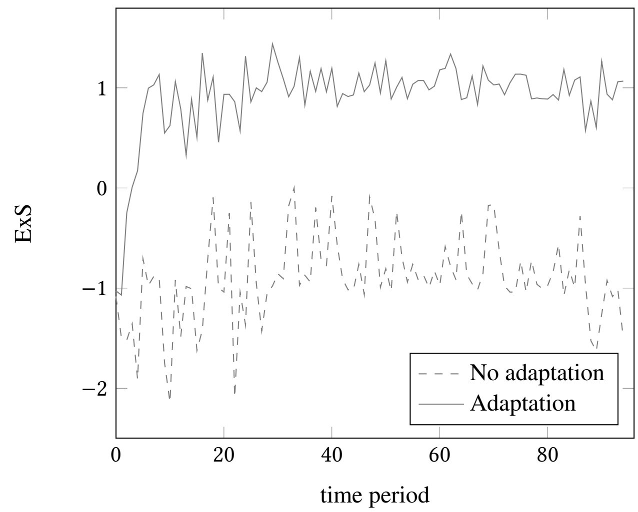

Figure 4: Satisfaction of objective 2 (minimise energy consumption) over 15-minute time periods.

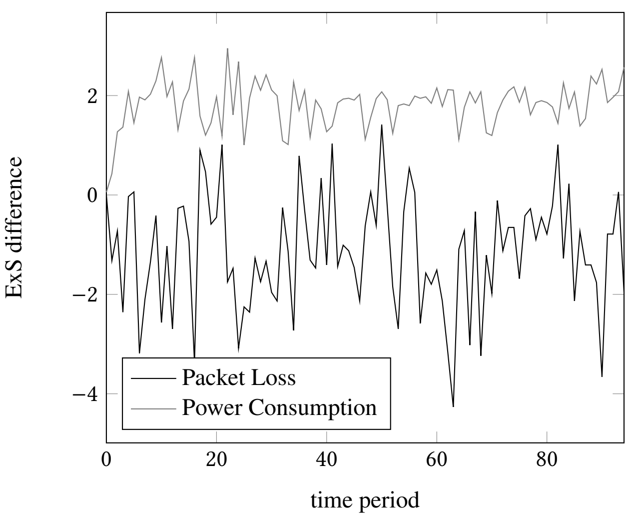

Figure 5: Difference in objective satisfaction when using adaptation, compared to when not using adaptation.

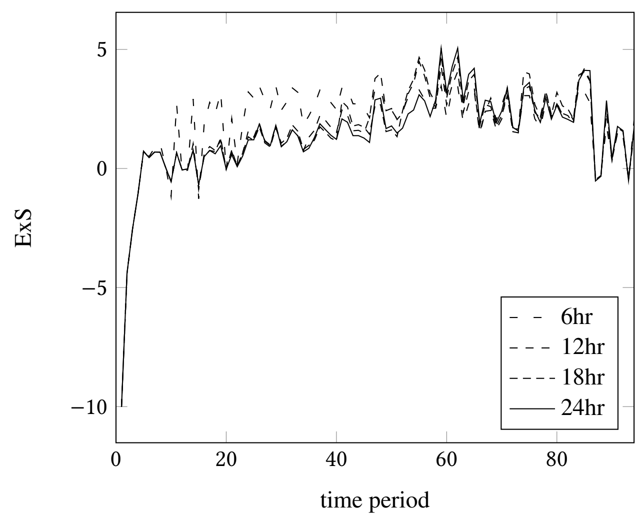

Figure 6: ExS ratings of objective 2 for each 15-minute period for the duration of the experiment (target value: 20) with store periods of the previous x hours.

system was defined as the sum of the ExS of all its NFRs. Applications of the ExS measure were discussed, and the use of DeSiRE in a case study detailing was demonstrated. The current results support the notion that DeSiRE leads to a better-informed requirements trade-off compared to more traditional selection methods.

We plan to further investigate how to determine reference values when the NFR specification does not specify a given one (e.g. in the case of *"as little as possible"*). The idea is to improve the current method of using the mean value of a property as a reference as it is currently assumed that all measured properties are normally distributed. This will require further discussion about quantitatively measuring how "minimised" or "maximised" a system property is.

DeSiRE can also be further refined by seeing if determining the Pareto Frontier of the available configurations before any calcuations of ExS values are performed in order to reduce the total number of calculations for each time a decision is to be made using ExS values, and whether this process saves on computational cost or not. DeSiRE can also be tested against a problem domain that contains a concave Pareto frontier(s), to verify if ExS values are still closer to the equally balanced configurations in these scenarios or if the ExS values tend to increase towards the edges of the concave front. Another research avenue is the application of the method at runtime, where the search space may not to be known in its entirety. We envision ExS values also be able to determine whether an SAS is to make an adaptation in the first place, with a view to minimising unnecessary adaptations.

REFERENCES

[1] S. S. Andrade and R. J. d. A. Macêdo. 2013. A Search-Based Approach for Architectural Design of Feedback Control Concerns in Self-Adaptive Systems. In *IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2013)*. 61–70. https://doi.org/10.1109/SASO.2013.42

[2] Sandro S Andrade and J Raimundo. 2014. Do search-based approaches improve the design of self-adaptive systems? A controlled experiment. In *Brazilian Symposium on Software Engineering (SBES 2014)*. IEEE, 101–110.

[3] Sandro S Andrade and J Raimundo. 2015. Assessing the benefits of search-based approaches when designing self-adaptive systems: a controlled experiment. *Journal of Software Engineering Research and Development* 3, 1 (2015), 2.

[4] K. Angelopoulos, A.V. Papadopoulos, V.E. Silva Souza, and J. Mylopoulos. 2016. Model predictive control for software systems with CobRA. *11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2016)* (2016), 35–46. https://doi.org/10.1145/2897053.2897054

[5] Luciano Baresi, Liliana Pasquale, and Paola Spoletini. 2010. Fuzzy goals for requirements-driven adaptation. In *18th IEEE International Requirements Engineering Conference (RE 2010)*. IEEE, 125–134.

[6] Valerie Belton and Theodor Stewart. 2002. *Multiple criteria decision analysis: an integrated approach*. Springer Science & Business Media.

[7] N. Bencomo. 2015. QuantUn: Quantification of uncertainty for the reassessment of requirements. In *Proceedings - IEEE 23rd International Requirements Engineering Conference (RE 2015)*. 236–240.

[8] Betty H. C. Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, Jeff Magee, Jesper Andersson, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Giovanna Di Marzo Serugendo, Schahram Dustdar, Anthony Finkelstein, Cristina Gacek, Kurt Geihs, Vincenzo Grassi, Gabor Karsai, Holger M. Kienle, Jeff Kramer, Marin Litoiu, Sam Malek, Raffaela Mirandola, Hausi A. Müller, Sooyong Park, Mary Shaw, Matthias Tichy, Massimo Tivoli, Danny Weyns, and Jon" Whittle. 2009. *Software Engineering for Self-Adaptive Systems: A Research Roadmap*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–26. https://doi.org/10.1007/978-3-642-02161-9_1

[9] L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos. 1999. *Non-Functional Requirements in Software Engineering*. Springer US.

[10] Lamiae Dounas, Raul Mazo, Camille Salinesi, and Omar El Beqqali. 2015. Continuous monitoring of adaptive e-learning systems requirements. In *IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA 2015)*. IEEE, 1–8.

[11] Golnaz Elahi and Eric Yu. 2011. Requirements Trade-offs Analysis in the Absence of Quantitative Measures: A Heuristic Method. In *Proceedings - the 2011 ACM Symposium on Applied Computing (SAC '11)*. ACM, 651–658.

[12] Heather J. Goldsby, Pete Sawyer, Nelly Bencomo, Danny Hughes, and Betty H.C. Cheng. 2008. Goal-Based Modeling of Dynamically Adaptive System Requirements. In *Proceedings - 15th Annual IEEE International Conference on the Engineering of Computer Based Systems (ECBS 2008)*.

[13] Salvatore Greco, J Figueira, and M Ehrgott. 2016. *Multiple criteria decision analysis*. Springer.

[14] Deshuai Han, Jianchun Xing, Qiliang Yang, Juelong Li, and Hongda Wang. 2016. Handling Uncertainty in Self-adaptive Software Using Self-learning Fuzzy Neural Network. In *IEEE 40th Annual Computer Software and Applications Conference (COMPSAC 2016)*, Vol. 2. IEEE, 540–545.

[15] M.U. Iftikhar, G.S. Ramachandran, P. Bollansée, D. Weyns, and D. Hughes. 2017. DeltaIoT: A Self-Adaptive Internet of Things Exemplar. In *12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2017)*. IEEE, 76–82.

[16] Seong-ick Moon, Kwang Hyung Lee, and Doheon Lee. 2004. Fuzzy branching temporal logic. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34, 2 (2004), 1045–1055.

[17] John Mylopoulos, Lawrence Chung, Stephen Liao, Huaiqing Wang, and Eric Yu. 2001. Exploring Alternatives During Requirements Analysis. *IEEE Softwware* 18, 1 (2001), 92–96. https://doi.org/10.1109/52.903174

[18] V. Pareto. 1896. *Cours D'Economie Politique*. F. Rouge, Lausanne, Switzerland.

[19] Luis Hernán García Paucar, Nelly Bencomo, and Kevin Kam Fung Yuen. 2017. Juggling Preferences in a World of Uncertainty. In *25th IEEE International Requirements Engineering Conference (RE 2017)*. 430–435. https://doi.org/10.1109/RE.2017.12

[20] Daniel J Power, Ramesh Sharda, and Frada Burstein. 2015. *Decision support systems*. Wiley Online Library.

[21] Andres J. Ramirez, Betty H.C. Cheng, Philip K. McKinley, and Benjamin E. Beckmann. 2010. Automatically Generating Adaptive Logic to Balance Non-functional Tradeoffs During Reconfiguration. In *Proceedings - the 7th International Conference on Autonomic Computing*. ACM, 225–234.

[22] A. J. Ramirez, B. H. C. Cheng, N. Bencomo, and P. Sawyer. 2012. *Relaxing claims: Coping with uncertainty while evaluating assumptions at run time*. Lecture Notes in Computer Science, Vol. 7590 LNCS. Springer. 53–69 pages.

[23] Mehrdad Saadatmand and Sahar Tahvili. 2015. A Fuzzy Decision Support Approach for Model-Based Tradeoff Analysis of Non-Functional Requirements. In *12th International Conference on Information Technology - New Generations (ITNG 2015)*. IEEE, 112–121.

[24] Chiara Sandionigi, Danilo Ardagna, Gianpaolo Cugola, and Carlo Ghezzi. 2013. Optimizing service selection and allocation in situational computing applications. *IEEE Transactions on Services Computing* 6, 3 (2013), 414–428.

[25] Nary Subramanian and Lawrence Chung. 2001. Software Architecture Adaptability: An NFR Approach. In *Proceedings - 4th International Workshop on Principles of Software Evolution (IWPSE 2001)*. ACM, New York, NY, USA, 52–61. https://doi.org/10.1145/602461.602470

[26] Sam Supakkul, Tom Hill, Lawrence Chung, Thein Than Tun, and Julio Cesar Sampaio do Prado Leite. 2010. An NFR Pattern Approach to Dealing with NFRs.. In *Requirements Engineering*. IEEE Computer Society, 179–188.

[27] A. F. Talon, E. R. M. Madeira, and M. B. F. Toledo. 2014. Self-adaptive fuzzy architecture to predict and decrease e-contract violations. In *Proceedings - 2014 Brazilian Conference on Intelligent Systems (BRACIS 2014)*. 294–299.

[28] R. Torres, H. Astudillo, and R. Salas. 2011. Self-adaptive fuzzy QoS-driven web service discovery. In *Proceedings - 2011 IEEE International Conference on Services Computing (SCC 2011)*. 64–71.

[29] E. Triantaphyllou. 2000. Multi-criteria decision making methods. In *Multi-criteria decision making methods: A comparative study*. Springer, 5–21.

[30] Friedrich von Wieser. 1927. *Social Economics*. Adelphi.

[31] Kristopher Welsh, Nelly Bencomo, Pete Sawyer, and Jon Whittle. 2014. Self-Explanation in Adaptive Systems Based on Runtime Goal-Based Models. *Trans. Computational Collective Intelligence* 16 (2014), 122–145. https://doi.org/10.1007/978-3-662-44871-7_5

[32] Kristopher Welsh and Peter Sawyer. 2009. Requirements Tracing to Support Change in Dynamically Adaptive Systems. In *Proceedings - 15th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2009)*.

[33] D. Weyns and R. Calinescu. 2015. Tele Assistance: A Self-Adaptive Service-Based System Exemplar. In *Proceedings - 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015)*. 88–92.

[34] D. Weyns and M. U. Iftikhar. 2016. Model-based simulation at runtime for self-adaptive systems. In *Proceedings - 2016 IEEE International Conference on Autonomic Computing (ICAC 2016)*. 364–373.

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

[36] Lotfi A Zadeh. 1996. Fuzzy sets. In *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh*. World Scientific, 394–432.