

Towards Priority-Awareness in Autonomous Intelligent Systems

Huma Samin
SEA, Aston University
Birmingham, UK, B47ET
<https://cs.aston.ac.uk/sea/>
h.samin@aston.ac.uk

Nelly Bencomo
SEA, Aston University
Birmingham, UK, B47ET
<https://cs.aston.ac.uk/sea/>
nelly@acm.org

Luis H. Garcia Paucar
SEA, Aston University
Birmingham, UK, B47ET
<https://cs.aston.ac.uk/sea/>
garcial2@aston.ac.uk

Peter Sawyer
SEA, Aston University
Birmingham, UK, B47ET
<https://cs.aston.ac.uk/sea/>
p.sawyer@aston.ac.uk

ABSTRACT

In Autonomous and Intelligent systems (AIS), the decision-making process can be divided into two parts: (i) the priorities of the requirements are determined at design-time; (ii) design selection follows where alternatives are compared, and the preferred alternatives are chosen autonomously by the AIS. Runtime design selection is a trade-off analysis between non-functional requirements (NFRs) that uses optimisation methods, including decision-analysis and utility theory. The aim is to select the design option yielding the highest expected utility. A problem with these techniques is that they use a uni-scalar cumulative utility value to represent a combined priority for all the NFRs. However, this uni-scalar value doesn't give information about the varying impacts of actions under uncertain environmental contexts on the satisfaction priorities of individual NFRs. In this paper, we present a novel use of Multi-Reward Partially Observable Markov Decision Process (MR-POMDP) to support reasoning of separate NFR priorities. We discuss the use of rewards in MR-POMDPs as a way to support AIS with (a) priority-aware decision-making; and (b) maintain service-level agreement, by autonomously tuning NFRs' priorities to new contexts and based on data gathered at runtime. We evaluate our approach by applying it to a substantial Network case.

CCS CONCEPTS

• **Computing methodologies** → **Probabilistic reasoning**; • **Theory of computation** → **Sequential decision making**; • **Software and its engineering** → **Extra-functional properties**; **Designing software**;

KEYWORDS

Autonomous and Intelligent software systems, runtime models, Priorities, Decision-Making, Non-functional Requirements

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8104-8/21/03.

<https://doi.org/10.1145/3412841.3442007>

ACM Reference Format:

Huma Samin, Luis H. Garcia Paucar, Nelly Bencomo, and Peter Sawyer. 2021. Towards Priority-Awareness in Autonomous Intelligent Systems. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea*. ACM, New York, NY, USA, Article 4, 10 pages. <https://doi.org/10.1145/3412841.3442007>

1 INTRODUCTION

Autonomous and Intelligent systems (AIS) exhibit autonomous decision-making to deal with the environmental uncertainty they find during execution and therefore, to adapt accordingly [12]. Such systems make decisions based on the trade-offs between non-functional requirements (NFRs) [8] in terms of their priorities for satisfaction [3, 5, 6, 13, 21, 33].

As such, a number of runtime modelling techniques have been developed that take into account NFRs' priorities [1, 4, 15, 17]. These modelling techniques are based on optimisation methods, including decision analysis and utility theory [25, 26]. The decision-making process supported by these techniques can be roughly divided into two parts: (i) at first the priorities of NFRs are determined by the stakeholders at design time, (ii) design selection follows where alternative options in the form of components, algorithms, protocols etc. are compared. Based on that, the ultimate preferred alternatives are chosen autonomously at runtime by the AIS. The alternatives chosen should be those that best satisfy the requirements, focusing most particularly on those requirements that constitute any Service Level Agreement (SLA) to which the AIS is subject to. We refer to the selection and binding of these design choices at runtime as *adaptive actions*. Based on these optimisation techniques, decision-making is focused on choosing the adaptive actions, under varying environmental contexts, that are expected to yield the highest utility value.

[Problems] A feature of such AIS is that they typically employ single objective optimization techniques. These techniques use a uni-scalar cumulative utility value to represent a combined priority for all NFRs [1, 2, 4, 15, 17]. However, the adaptive actions taken by the AIS can have different impacts, either positive or negative, on the satisfaction levels of individual NFRs. For example, in the case of an IoT network, decreasing the transmission power on the networks links supports the minimization of energy consumption in exchange of a negative impact on the packet delivery performance. The problem with existing techniques is that the combined priority

for all NFRs hides any information about the different impacts of adaptive actions on the NFRs in terms of the extent to which their individual priorities are respected. Further, these impacts may change over time during the operation of AIS. For example, following with the example of the IoT network, varying traffic loads and the subsequent interference on the network links may lead the IoT system to evolve its adaptation strategies. This, in turn, may affect the NFRs' satisfaction, leading to violation of the SLA. Hence, the priorities assigned at design time may no longer be valid at runtime due to unforeseen environmental contexts. The element of unpredictability arising from the environment, in which an AIS operates and must respond to, means that the AIS has to make decisions about adaptation actions under uncertainty. This is a key challenge for the AIS and an important reason why techniques that oversimplify adaptation actions or obfuscate decision rationale by, for example, collapsing all NFR priorities into a single utility function, are inappropriate.

[Principal Ideas and Contributions] We argue that the priorities of NFRs should not be modelled as a single combined value. Nor should they be considered static. In this paper, we propose the use of reinforcement learning, based on Multi-Reward Partially Observable Markov Decision Process (MR-POMDP) [24]. MR-POMDP is a multi-objective sequential decision-making technique that we use to support the runtime modelling and reasoning of *distinct priority* values of NFRs using a *vector-valued reward function*. We discuss the means of using the concept of rewards in MR-POMDPs as a way :

- a) to support better-informed decision-making in terms of the distinct priorities of NFRs and study the explicit impacts of the adaptation actions on the individual NFRs hence making an AIS *Priority-Aware* [22, 27, 29].
- b) to provide an AIS with a principled way to maintain the required NFR satisfaction levels by *autonomously tuning* NFRs' priorities at runtime when an unanticipated context is encountered for which the prescribed priorities, initially provided by stakeholders, would result in violation of the *SLA*.

We also provide a proof of concept by applying the proposed approach to the case of a Remote Data Mirroring (RDM) network [10] and comparing it to the existing state-of-the-art techniques. On the basis of the experiments performed, we show that the *priority-aware* decisions taken using MR-POMDP better satisfy the NFRs, thereby conforming to the *SLA*.

[Organization of the paper] The paper is organized as follows: Section 2 gives an overview of the background with the state of the art techniques. Section 3 explains concepts related to autonomous decision making. Section 4 presents the MR-POMDP++, a runtime model to support priority-aware autonomous decision making. In Section 5, the experiments and evaluations are presented. Finally, the conclusions and future work are presented in Section 6.

2 BACKGROUND

In this section, we discuss different state-of-the-art techniques that deal with prioritization of NFRs in AIS. We have classified these techniques in two categories as follows:

2.1 Techniques used at Design Time and Off-line

The Analytic Hierarchy Process (AHP) and Primitive Cognitive Network Process (P-CNP) are techniques based on Multi-Criteria Decision Making (MCDM) that support the ranking and explicit modelling of NFRs. These approaches have been used in [13] and [33] respectively as design time techniques. However, the authors of [18] have shown how they can support a POMDP model with P-CNP approach to assist with the prioritization of NFRs at runtime. However, even if P-CNP has been used as a runtime technique, it is not autonomous and does not work within the POMDP model.

Other approaches support optimization of NFRs using search-based techniques [3, 20]. These approaches have tackled the optimization of priorities either at design time or off-line. In contrast to these techniques described above, our approach supports *runtime reasoning of distinct priorities* associated with NFRs to therefore offer *priority-aware decision-making*.

2.2 Techniques used at Runtime

In order to support decision-making under uncertainty exhibited by an AIS, a number of runtime modelling techniques exist that use prioritisation of NFRs. The techniques in [1, 4, 15, 17] make use of Markov-based approaches such as Markov Decision Process (MDPs), Partially Observable Markov Decision Processes (POMDPs) and Discrete Time Markov Chains (DTMCs) along with probabilistic model checking. As these techniques are *Markov-based* they also support the *quantification of uncertainty* by the use of probabilities over the variables of the states of the environment. They lack treatment of the distinct NFRs' priorities. Furthermore, the approaches based on MDPs and POMDPs model the priorities of the NFRs as a scalar reward value to denote a cumulative priority of all the NFRs. This can lead to undesirable effects. A change in this single reward value can have a significant impact on the adaptation decision to be selected and will lead to ignorance of some NFRs during decision-making. The approach of RE-STORM [17] uses POMDPs and therefore it lacks the modelling of distinct NFRs' priorities. However, it uses ARROW [18] (as an external support) to reason about the priorities based on runtime evidence which is external to POMDP. Control theory based approaches such as [14, 19] have also been used to support explicit runtime configuration and tuning of NFRs. However, they are subject to more severe limitations w.r.t. the treatment of priorities than the approaches described above. The technique in [14] lacks the autonomous prioritization of NFRs, while the approach in [19] lacks the capability of dealing with the NFRs having the same priority rank. These techniques also lack the consideration of uncertainty as a quantifiable measure.

In summary, different approaches have been shown to support runtime decision-making in AIS. However, existing Markov based solutions fail to adequately take account of NFRs' priorities, potentially leading to compromised adaptation decisions. Our work seeks to support autonomous decisions whilst explicitly respecting the NFRs' priorities.

3 KEY CONCEPTS

In this section, we briefly present key concepts related to autonomous decision-making in AIS driven by NFRs [8, 9], Partially Observable

Markov Decision Process [31] and Multi-Reward Partially Observable Markov Decision Process (MR-POMDP)[24].

3.1 Autonomous Decision Making

AIS are continuously exposed to uncertain environmental contexts that may cause changes on the the satisfaction of NFRs. During the decision-making process of an AIS, the system performs different adaptation actions that have different effects on the satisfaction levels of NFRs. The key entities involved in the decision-making process of an AIS are as follows [2]:

3.1.1 NFR. The main objective of autonomous decision-making is to choose the best adaptation strategy, in order to satisfy its quality goals (i.e. the NFRs), while achieving the functional goals. The NFRs are associated with two important concepts [9]:

- **Satisfaction Level:** The satisfaction level of a particular NFR refers to the extent to which that NFR has been satisfied as a result of the action a (related to task a) performed by the AIS. The satisfaction level can be represented by the conditional probability distribution $P(\text{NFR}_i \text{ is satisfied} \mid \text{action } a)$ [2].

- **Priority:** The priority value for a NFR is a scalar value that represents its significance or relevance for satisfaction. Due to the change in environmental contexts at runtime, the priority for satisfaction of NFR may change as well.

3.1.2 Monitorables. The AIS continuously monitors environmental changes over time in order to support the decision-making. Monitoring is done by using monitorables that collect information from the environment.

3.1.3 Tasks. The task is a concept taken from goal models community. Tasks are defined as the adaptation strategies or actions that correspond with a discrete set of software configurations or adaptations that are selected by an AIS during decision-making. In order to achieve the target functional goal along with satisfying the NFRs, the AIS selects a task [2, 17] on the basis of the monitorable values, the satisfaction levels and priorities for the NFRs at a given time step.

3.1.4 Task Contribution. Task contribution refers to the effect (good or bad) of performing a task on the NFRs' satisfaction levels at a particular time step.

As tasks are defined as the adaptation strategy or actions, we use the term of *actions* in this paper to refer to the term *tasks*.

3.2 POMDPs

POMDPs [31] consider the decision-making agents working in a partially observable environment in order to deal with and quantify the effects of uncertainty about the state of the environment. A POMDP is specified as a tuple $\langle S, A, Z, T, O, R, \gamma \rangle$ where S represents the set of states; A is the set of Actions; Z is the set of Observations related to the set of states; T is the transition function $T(s, a, s') = P(s' \mid s, a)$ specifying the probability of moving to the next state s' given an action a and current state s ; O is the observation function $O(s, a, z) = P(z \mid s, a)$ specifying the probability of observing the observation z given an action a and resulting state s ; R is the reward function $R(s, a)$ specifying a scalar real value generated by the environment as a feedback of the action a performed by the

agent given the state s of the environment; γ is the discount factor. The agent tries to find the policy π , a mapping from the state of the environment to action, that maximizes the value function i.e. the expected utility value of the sum of discounted rewards as follows:

$$V_\pi = E_\pi [R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} \dots \mid s_t] \quad (1)$$

On the basis of the current state of the system, the value function is used to compute the cumulative reward value that we would expect to get in the future if a particular action is performed in that state. Thus, the reward value $R(s, a)$ generated by the environment is used to evaluate the effect of performing an action a during the state s with the help of a value function at runtime. Therefore, it assigns a cardinal scale to each decision made during a specific state as a result that indicates its *priority*.

As the states in a POMDP are not fully observable, a belief b over the states of the system is maintained. In reference to point-based planning methods [28, 31] for solving POMDPs, the value function V_b , in terms of the belief, is represented by a set of α -vectors A . Each α -vector, associated with an action a , has a length of $|S|$ to provide a value for each state s . The α -vector is computed as follows:

$$\alpha_a = [V(s_i), V(s_{i+1}), \dots, V(s(n))] \quad (2)$$

Here $V(s_i)$ represents the value of the value function for state s_i given a total number of n states.

Thus, the value of the belief given A is computed as:

$$V_b = \max_{\alpha \in A} b \cdot \alpha \quad (3)$$

Therefore, for each belief b , a set of α -vectors A provides a policy π_A for the action that maximizes the value.

3.3 MR-POMDPs

Multi-reward POMDPs (MR-POMDP for short) [23, 24], based on the constructs of Partially Observable Markov Decision Process (POMDP)[31], are used to solve multi-objective decision problems. In contrast to POMDP, which is a single objective approach. MR-POMDP [23, 24] has more than one reward value represented in the form of a vector-valued reward function \mathbf{R} . In MR-POMDPs, each objective (i.e. each NFR in our case) has its own separate reward value. Hence, the size of the reward vector equals the number of objectives. These reward vector values represent explicitly the cardinal impacts of the actions on individual objectives under uncertain contexts. Furthermore, the value function, given an initial belief \mathbf{V}_b for the policy π of the MR-POMDP, is also a vector. Thus, each single element in the value function vector represents the expected utility value related to one objective. We use this inherent capability of MR-POMDPs to compute expected utility value for each individual objective as a base for the *autonomous tuning* process of the priorities at runtime.

Given \mathbf{R} as a vector, each element in the α -vector is also a vector thereby, creating an α -matrix, \mathbf{A} . Each row in the α -matrix represents the values for the objectives in a particular state. The multi-objective value of taking an action a associated with alpha matrix \mathbf{A} under a belief b is computed as follows:

$$\mathbf{V}_b = b \mathbf{A} \quad (4)$$

As the value function is represented as a vector in the MR-POMDP, there may be multiple policies. In order to select the best optimal policy from these multiple policies, a scalarization function $f(\mathbf{V}_b, W)$ is used to scalarize the value vectors \mathbf{V}_b with respect to the weights W [23] that is computed as follows:

$$f(\mathbf{V}_b, W) = W \cdot V_b = w_1 V_{b_1} + w_{i+1} V_{b_{i+1}} \dots + w_n V_{b_n} \quad (5)$$

where w_i and V_{b_i} refer to the weight and value for the i th objective given n number of total objectives. The size of the weights vector W is also equal to the number of the objectives. These weights vector values are computed by the decision-making agent using the Optimistic Linear Support (OLS) algorithm [24] at runtime. Hence, for a given belief b , α -matrix for each action and weight w , we can compute the policy π_A that takes the maximal value using equations 4 and 5 as:

$$V_b^*(w) = \max_{A \in \mathcal{A}} bAW \quad (6)$$

4 PRIORITY-AWARE MR-POMDP++

This section presents MR-POMDP++, a priority-aware model, for autonomous decision-making of the AIS. Next, we present the case of a RDM network and explain the concept using this case as an example.

4.1 Remote Data Mirroring

In order to illustrate our proposed approach, we consider the case of a RDM network [10] based on the operational model presented in [11]. The RDM system is a disaster recovery system that tolerates failures by maintaining multiple copies (i.e. replicas) of data at remotely located servers (i.e. Mirrors). Therefore, it maintains data availability by preventing data loss. Each network link has an associated operational cost¹ and a measurable throughput, latency and loss rate used to determine the reliability, performance and cost of the RDM system. The goal here is to satisfy the NFRs of Minimization of Costs (MC), Maximization of Performance (MP)² and Maximization of Reliability (MR) under environmental uncertainty of link failures and varying ranges of bandwidth consumption [10]. For this purpose, the network is required to continuously take adaptive actions of switching between the topological configurations of Minimum Spanning Tree (MST) and Redundant Topology (RT) to maintain better levels of satisfaction of NFRs. Both the configurations offer a different impact on the NFRs' satisfaction. The topological configuration of RT provides a higher level reliability than MST topology but it has a negative impact on the satisfaction of the MC and MP as the cost of maintaining non-stop RT topology will be high and due to data redundancy, the performance can be reduced. On the other hand, MST topology supports the satisfaction of MC and MP by maintaining a minimum spanning tree for the network.

The state of the RDM corresponds with the state of the NFRs of the RDM (i.e. MC,MP,MR), which is hidden and just partially observable by the monitoring infrastructure. Based on the partial knowledge collected as evidence from the monitoring infrastructure, the RDM infers the level of satisfaction of the NFRs to then take the decision of selecting the best topology. The decision-making takes

¹In RDM system, Operational Cost is measured in terms of intersite network traffic.

²In the case of RDM network, we are measuring performance in terms of total time to write the data i.e. the sum of the time to write each copy of data on each remote site.

into account the SLAs (stated from the requirements specifications), and priorities of the NFRs. The use of MR-POMDP to support such a decision making is presented next, which is called MR-POMDP++.

4.2 MR-POMDP++

In this section, we present MR-POMDP++, an extension of MR-POMDPs to represent the satisfaction levels of NFRs and deal with priorities to underpin autonomous decision-making in AIS, and support priority-awareness (see Fig 1). Next, we present the rules to map NFRs, their priorities and their related decision making to a MR-POMDP to support priority-aware runtime decision-making in AIS.

4.2.1 NFR satisfaction and MR-POMDP states. An AIS takes different adaptive actions to achieve satisfaction of NFRs at runtime. These actions have different effects (either positive or negative) on the satisfaction of NFRs. As the NFRs can neither be labelled as fully satisfied nor fully violated, their satisfaction levels cannot be represented as absolute values True or False. This is known as the lack of crispness in the nature of satisfaction of NFRs [8]. Therefore, the satisfaction levels of NFRs are modeled in this paper using probability distributions. A NFR is considered as satisfied if it meets an acceptability threshold defined by the experts, which is expressed using probably values between [0,1]. For example, in the RDM case, the satisfaction level of MC can be represented as $P(\text{MC}=\text{True})=0.75$ and the NFR of MC can be considered as highly satisfied if it meets the acceptability threshold of 0.7 i.e. $P(\text{MC}=\text{True}) \geq 0.7$ in order to conform to the SLAs.

In MR-POMDP++, we consider each state as the representation of the set of combinations of NFRs as shown in Fig 1. As the case of the states in a MR-POMDP, the properties associated with the NFRs are not directly observable, instead a belief (i.e. a probability) over the states is maintained. Hence, the satisfaction levels of NFRs are described in the form of probability distributions $P(NFR_i)$ where NFR_i belongs to the set of NFRs [17].

On the basis of this description, we derive a mapping rule as follows:

Rule: 1 *The state $s \in S$ in MR-POMDP represents the set of combinations of the non-functional requirements ($NFR_1 \dots NFR_n$). As the states in the MR-POMDP are partially observable, the satisfaction levels of the NFRs can be represented in the form of probability distributions $P(NFR_i)$.*

These probabilities can be used to conclude if the satisfaction levels meet acceptable thresholds.

Using Rule 1, the size of set S in MR-POMDP representing the total number of states in terms of the satisfaction levels of NFRs can be computed as $|S| = 2^{|NFR|}$. Where $|S|$ corresponds to the size of set S , NFR corresponds to the NFR and 2 corresponds to True and False. For example, in RDM network, we consider the three NFRs of MC, MR and MP. As the number of NFRs is 3, so the number of states for MR-POMDP will become $|S| = 2^3 = 8$ as shown in Table 1.

4.2.2 NFR priorities and Rewards Vectors. NFRs have priorities associated with them that signify their importance for satisfaction. The higher the priority, the more important it is to satisfy the NFR. The MR-POMDPs use a vector-valued reward function

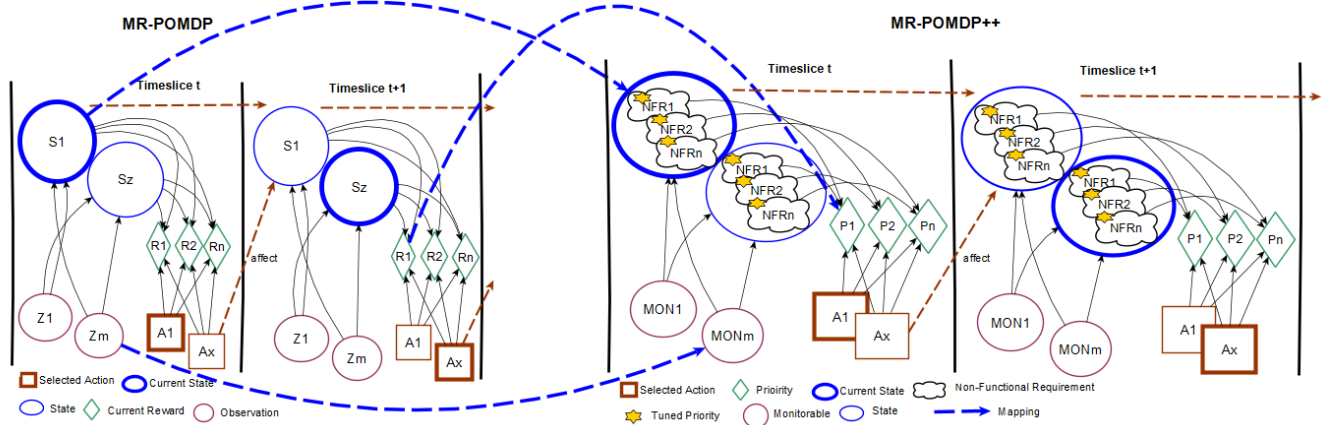


Figure 1: Mapping of MR-POMDP to Priority-Aware MR-POMDP++

Table 1: States of Priority-Aware MR-POMDP for RDM Network

S	NFR ₁ =MC	NFR ₂ =MR	NFR ₃ =MP
s ₁	True	True	True
s ₂	True	True	False
s ₃	True	False	True
s ₄	True	False	False
s ₅	False	True	True
s ₆	False	True	False
s ₇	False	False	True
s ₈	False	False	False

to associate a separate reward value with each of the objectives (NFRs in our case). These reward values are generated as a feedback signal according to the decision prompted by MR-POMDPs. The reward value associated with a particular objective indicates the effect, either positive or negative, of performing an action on the satisfaction of that objective. Consequently, the reward vector values specify a relative ranking of the objectives in terms of the cardinal effect that an action will have on the fulfillment of that objective under an uncertain environmental context.

For example, in the case of the RDM network, if network link failures are observed at a particular point of time, it may result in an increase in data packet loss. As a consequence, a negative effect on the reliability of the system would be observed. In order to support the satisfaction of MR, the decision-making agent might select the adaptation action of switching to RT. However, RT might have its own negative effects on the NFRs MC and MP. Consequently, given the current environmental context of link failures, on the basis of the selected action of RT, the system will generate a higher immediate reward for MR (e.g. 75) than for MC and MP (that could be e.g. -50 and -25). In this case, according to MR-POMDP, the reward for MR is greater than that of MC and MP because the satisfaction of MR is more important at this point of time given the current conditions. Hence, the reward vector $\mathbf{R}(s,a) = [R_{MR}, R_{MC}, R_{MP}] = [75, -50, -25]$ presents a relative ranking of the NFRs MR, MC and MP in terms of the effect that the action RT will have on their satisfaction. Thus, the *reward vector* in MR-POMDP can be used to model the *NFRs' priorities* by indicating their desirability in terms of their satisfaction given a particular state of the environment as shown in Fig 1. These reward values are initially assigned by the

Table 2: Reward Values for the NFRs in the RDM Network

S	Action(A)	Reward Vector Values		
		$R_{NFR1} = R_{MC}$	$R_{NFR2} = R_{MR}$	$R_{NFR3} = R_{MP}$
s ₁	MST	39.17	39.0	40.0
s ₂	MST	41.0	40.0	39.0
s ₃	MST	39.0	38.0	38.5
s ₄	MST	17.0	16.0	15.0
s ₅	MST	44.0	43.0	43.5
s ₆	MST	29.0	28.0	27.0
s ₇	MST	14.0	13.0	13.5
s ₈	MST	2.0	1.0	1.0
s ₁	RT	41.0	43.0	41.0
s ₂	RT	32.0	33.0	31.0
s ₃	RT	28.0	29.0	27.0
s ₄	RT	26.0	27.0	25.0
s ₅	RT	28.0	29.0	27.0
s ₆	RT	16.0	17.0	15.0
s ₇	RT	23.0	24.0	22.0
s ₈	RT	11.0	12.0	10.0

requirements engineers and domain experts. Using this description, we derive our next rule as follows:

Rule: 2 The values in the reward vector $\mathbf{R}(s,a)$ over the execution of an action $a \in A$ given state $s \in S$ in MR-POMDP represent the priorities of non-functional requirements (NFRs).

$$\mathbf{R}(s,a)=[R_{NFR1},R_{NFR2},\dots,R_{NFRm}]$$

where R_{NFR1} represents the reward for NFR1 indicating the priority value of NFR_1 and so on.

In the RDM system, the number of NFRs under consideration is 3, therefore the reward vector is represented as $\mathbf{R}(s,a)=[R_{MC},R_{MR},R_{MP}]$. R_{MC} represents the priority value for MC, R_{MR} represents the priority values for MR and R_{MP} represents the priority values for MP at runtime. The reward vector values (provided by the experts) for NFRs in the RDM network are shown in Table 2.

4.2.3 Expected Utility Values and Autonomous Tuning of Priorities:

In AIS, the priorities of NFRs may vary according to the changes in the context. Using the reward vector, MR-POMDP++ offers the opportunity for *autonomously tuning* the NFRs' priorities, by computing the separate *expected utility value* for each NFR, during the operation of MR-POMDPs using equation 1 as follows:

$$V_{NFRi} = E_{\pi}[Ri_t + \gamma Ri_{t+1} + \gamma^2 Ri_{t+2} \dots | s_t] \quad (7)$$

where V_{NFRi} and Ri represents the *expected utility value* and reward values for NFRi. Here, the rewards that represent the initial

priorities, are taken into account for the computation of the distinct *expected utility values* for each NFR. Hence, these *expected utility values* represent the new values for the *tuned priorities* of the different NFRs. The *expected utility values* ponder the individual effect of performing an action on the satisfaction of a NFR given a particular context, and are considered while making the decisions at runtime.

4.2.4 Modelling of Actions, Transitions and Observations:

The actions, transition and observation functions of a MR-POMDP in terms of the NFRs, adaptation strategies and the monitoring variables of the RDM system are explained as follows:

Actions: represent the adaptation strategies to maintain the satisfaction of the NFRs of MC, MR and MP. The actions for the case of RDM network are the two topological configurations of Minimum Spanning Tree (MST) and Redundant Topology (RT).

Transition Function: According to Rule 1, the states are represented as a combination of NFRs in MR-POMDP++. The transition probabilities $T(s,a,s')$ are factored as marginal conditional probabilities of NFRs $P(MC' | MC, a)$, $P(MR' | RPL, a)$ and $P(MP' | MP, a)$ using the property of conditional independence and Bayes rule [17] as follows:

$$T(s, a, s') = P(s' | s, a) = P(MC' | MC, a)P(MR' | MR, a)P(MP' | MP, a) \quad (8)$$

The conditional transition probabilities for the NFR MC going from one state to another as a result of action for the RDM case are shown in Table 3³. These transition probabilities are also defined based on expert knowledge.

Observations: As the states of NFRs are not directly observable, we use monitorables to obtain observations required for monitoring the satisfaction levels of NFRs on the basis of the set of possible values or information obtained from the environment. For this purpose, three monitorable variables Ranges of Bandwidth Consumption (RBC), Active Network Links (ANL) and Total Time for Writing (TTW) related to the NFRs MC, MR and MP respectively are specified in the RDM system. Higher the value for the ANL, higher will be satisfaction of MR. On the other hand, lower the values of RBC and TTW, higher will be the satisfaction of MC and MP. In RDM System, all of these monitorable variables have range boundaries that are specified in the requirements specifications by the experts and are provided in [32].

In MR-POMDP++, the observations of each monitorable are described by the probability distributions over its possible values. Hence, Observation probability represented by $O(s',a,z)$ (i.e. $P(z | a, s')$) specifies the probability of observing the observation z after executing action a and transition to state s' .

Like the transition model, we also factor the observation model into the product of conditional probabilities such that: $P(z | s', a) = P(Mon_1, \dots, Mon_n | s', a)$ [17]. Hence for RDM we have:

$$\begin{aligned} P(z | s', a) &= P(RBC, ANL, TTW | s', a) \\ &= P(REC | s', a)P(ANL | s', a)P(TTW | s', a) \end{aligned} \quad (9)$$

³The conditional probabilities tables for MR and MP are provided on the following link: https://gitlab.com/re_research/rdmexperiments/

Table 3: Transition Probabilities for NFR MC

NFR ₁ =Minimization of Cost (MC)					
Action(A)	MC _t	MR _t	MP _t	P(MC _{t+1} = T)	P(MC _{t+1} = F)
MST	True	True	True	0.9	0.1
MST	True	True	False	0.88	0.12
MST	True	False	True	0.92	0.08
MST	True	False	False	0.9	0.1
MST	False	True	True	0.85	0.15
MST	False	True	False	0.83	0.17
MST	False	False	True	0.87	0.13
MST	False	False	False	0.85	0.15
RT	True	True	True	0.86	0.14
RT	True	True	False	0.84	0.16
RT	True	False	True	0.88	0.12
RT	True	False	False	0.86	0.14
RT	False	True	True	0.73	0.27
RT	False	True	False	0.71	0.29
RT	False	False	True	0.75	0.25
RT	False	False	False	0.73	0.27

Table 4: Observation Probabilities- CPT RBC: $P(RBC | MC, a)$

Mon ₁ =Ranges of Bandwidth Consumption(RBC)				
Action(A)	MC _t	P(RBC _{t+1} < x)	P(MC _{t+1} in [x, y])	P(MC _{t+1} >= y)
MST	True	0.8	0.15	0.05
MST	False	0.72	0.18	0.1
RT	True	0.78	0.16	0.06
RT	False	0.68	0.2	0.12

The conditional observation probabilities for the monitorable RBC, defined by the experts, for the RDM case are presented in Table 4⁴.

5 EXPERIMENTAL EVALUATION

This section describes the experiments performed by applying the decision-making process of MR-POMDP++ to the case of RDM network [10]. For validation purposes, comparisons are presented with the existing techniques of RE-STORM [17] and ARROW [18]. As the approach of RE-STORM is implemented using DESPOT (a POMDP solver) and ARROW using DESPOT with P-CNP [33], we refer to these techniques as DESPOT and DESPOT-ARROW respectively from now onwards in the paper. Due to the limitation of space, a more detailed report of the experiments performed is documented in [32].

Next, we discuss the MR-POMDP solver that we have used as well as the initial setup for RDM network describing the SLA and the experimental scenario.

5.1 OLSAR: a MR-POMDP Solver

Solving MR-POMDP is often a computationally intractable problem [30]. However, techniques exist that solve MR-POMDP as an approximation. In general, MR-POMDP solving methods can be divided into two main categories: optimal solving methods, and near-optimal and approximate methods. The technique used in the experiments of this paper is the Optimistic Linear Support with Alpha Re-use (OLSAR) algorithm [24], a point based MR-POMDP solver that generates approximate solutions by performing approximate backups during the computation of α -vectors over a set of sampled belief values. The algorithm has proven to scale during the experiments performed. The reuse of the alpha matrix in the algorithm also makes it efficient. The experiments were performed on a Lenovo Thinkpad with intel Core i7, 8th Gen processor and 16 GB RAM.

⁴The observation probabilities for ANL and TTW are provided on https://gitlab.com/re_research/rdmexperiments/

5.2 Initial Setup

The RDM network under consideration consists of 25 RDM Mirrors (i.e. the servers), which are used to hold multiple copies of data, with 300 physical links in total that can be used to transfer data between the mirrors [7]. In such a setup, the maximum possible number of concurrent active network links that does not affect the assigned budget is considered as 120 [7]. For the current set of experiments, our focus is on the NFRs concerned with the quality and performance attributes [8] of the RDM network such as Minimization of Costs (MC), Maximization of Reliability (MR) and Maximization of Performance (MP). For the initialization of the MR-POMDP model, the initial conditional probabilities used for the Transition model are presented in Table 3⁵, for the Observation Model in Table 4⁶ and the Reward vector values are presented in Table 2.

5.2.1 Service Level Agreements. The initial set up of the experiments also considers the Service Level Agreements (SLA) for the RDM network that reflect the required satisfaction levels of NFRs. These SLA are about the satisfaction thresholds and are defined by the system experts. The knowledge and domain expertise applied is based on [7, 10], The SLA, which indicate the requirements of the suitable zone of NFRs' satisfaction expressed in terms of the MR-POMDP++, are as follows:

R1: The probability of satisfaction of Minimization of Cost shall be greater than or equal to 0.70. i.e. $P(\text{MC}=\text{True}) \geq 0.70$

R2: The probability of satisfaction of Maximization of Reliability shall be greater than or equal to 0.85 i.e. $P(\text{MR}=\text{True}) \geq 0.85$

R3: The probability of satisfaction of Maximization of Performance shall be greater than or equal to 0.75. i.e. $P(\text{MP}=\text{True}) \geq 0.75$

The NFRs that have satisfaction levels below these threshold values are considered to be presenting poor levels of satisfaction.

5.3 Experimental Scenario

The experiments have been designed to study the unforeseen impacts of actions on the levels of satisfaction of NFRs. The unmatching NFRs' priorities in such an uncertain context may lead to the need of tuning these priorities by the MR-POMDP++. The idea is to evaluate how MR-POMDP++ offers priority-aware decision-making and informed choices related to individual NFRs.

In order to study the impact of actions on the satisfaction levels of the NFRs MC, MR and MP, we simulate dynamic situations by introducing random changes in the environment of the RDM network. The random changes are introduced to simulate failures in the network links during execution of AIS. Such failures may be due to problems in devices such as switches or routers. We refer to these random changes as the disturbance levels that may occur at runtime. In order to introduce these disturbance levels, deviations from the initially defined transition probabilities (i.e. $P(\text{NFR}_{t+1} = \text{True} | \text{NFR}_t, A_t)$) for the topologies (MST and RT) are introduced randomly at runtime.

⁵The conditional probabilities for NFRs MR and MP are provided in https://gitlab.com/re_research/rdmexperiments/

⁶The observation probabilities for ANL and TTW are provided in https://gitlab.com/re_research/rdmexperiments/

Table 5: Experiment Results for timesteps 45 - 51

Time	Action	Val _{MC}	Val _{MR}	Val _{MP}	Sat _{MC}	Sat _{MR}	Sat _{MP}
45	MST	392.131	388.213	395.327	0.90299	0.92395	0.91017
46	MST	391.980	388.045	395.155	0.90445	0.89257	0.91361
47	MST	391.978	388.043	395.154	0.9052	0.89170	0.91431
48	MST	391.715	387.744	394.835	0.90532	0.83722	0.91445
49	RT	386.767	392.139	386.258	0.90618	0.92089	0.91524
50	MST	392.302	388.331	395.313	0.86462	0.95651	0.84732
51	MST	392.139	388.219	395.331	0.90089	0.92493	0.90972

^aVal_{NFR} represents the expected utility value of NFRi

^bSat_{NFR} represents satisfaction level of NFRi

We have considered the following context scenario for the purpose of evaluation of our results:

Detrimental Context: The disturbance levels, that are introduced, simulate an unexpected data packet loss during the execution of the AIS. An increase in the packet loss during the execution of the RT topology would result in an unusual rate of data forwarding that would lead to higher level of bandwidth consumption and reduction of the performance. As a result, the satisfaction levels of MC i.e. $P(\text{MC}=\text{True})$ and MP i.e. $P(\text{MP}=\text{True})$ would be expected to be reduced. For the current set of experiments, to simulate small realistic changes, a maximum deviation of 12 percent from the current transition probabilities is introduced for a randomly selected duration (between 5 to 15 time steps) for the selected disturbance level.

Case:1– Priority-Aware Decisions and Autonomous Tuning of NFRs' Priorities: Here we focus on demonstrating priority-awareness by MR-POMDP++ and how it supports compliance with the SLA. In order to perform priority-aware decision-making, our proposed approach makes use of MR-POMDPs, to represent distinct priorities of NFRs. We study how the priorities of NFRs, represented in the form of rewards in MR-POMDP++ model, have an impact on the action selection for the purpose of satisfaction of NFRs as shown in Table 5. For example, using the initial setup, at *timestep 45*, the selection of the MST topology over RT provides the best possible trade-off. This is due to the fact that the expected utility values for MP and MC are 395.327 and 392.131 respectively, which are higher than those for MR, which has an *expected utility value* of 388.213 as presented in Table 5.

The above shows that the usage of MST has more positive impact on MP and MC than on MR when compared to the alternative RT topology (for which the *expected utility values* were 386.718 for MC, 392.086 for MR and 386.203 for MP at this point of time). Therefore, the system selects MST as the preferred topology, in order to support the reduction of inter-site network links cost and therefore, improve the performance of the network.

In contrast, at *timestep 49*, the system takes the adaptive action of switching to the topology RT. This is due to the fact that the expected utility value of MR being 392.139 is higher than the *expected utility values* of MC and MP i.e. 386.767 and 386.258 respectively, as shown in Table 5. As a result, the adaptive action of RT topology at this timestep improves the satisfaction level of MR from $P(\text{MR}=\text{True})=0.83722$ to $P(\text{MR}=\text{True})=0.92089$ and therefore, satisfying the SLA of $P(\text{MR}=\text{True}) \geq 0.85$. Hence, these *expected utility values* representing the new values for the tuned priorities of the

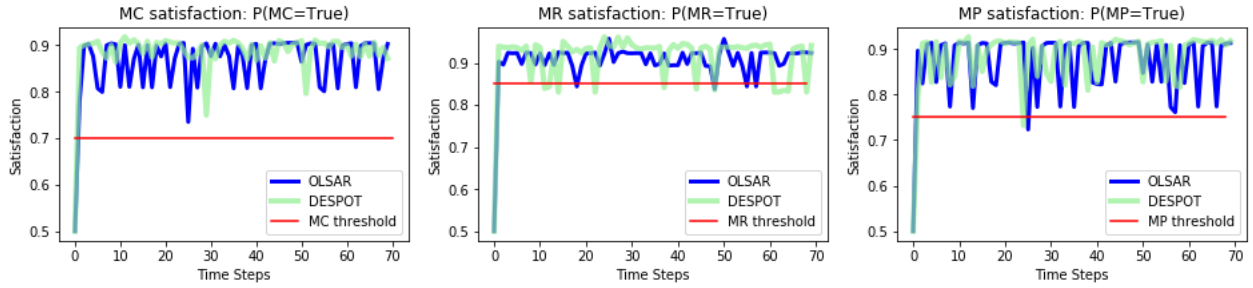


Figure 2: Satisfaction of NFRs over Time under Stable Conditions

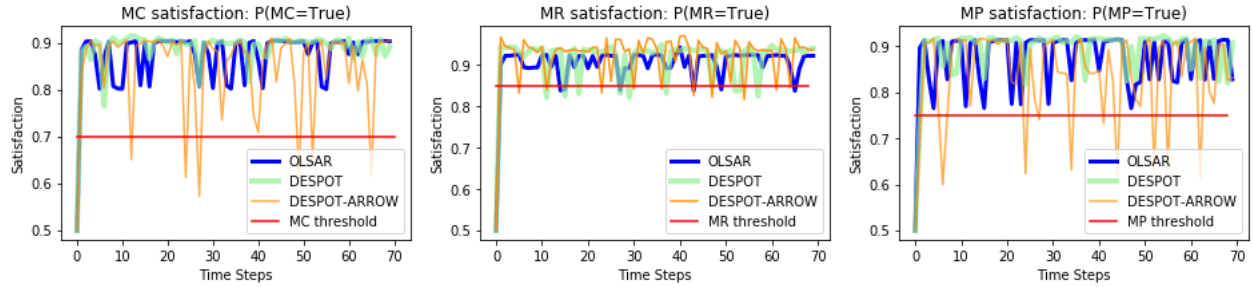


Figure 3: Satisfaction of NFRs over Time under Detrimental Conditions

NFRs are taken into account by MR-POMDP++ during decision-making. This is one of the *contributions* of the MR-POMDP++ approach, different from other approaches, to offer the AIS with a priority-aware decision-making process.

During the setup of experiments, the initial set of priorities for NFRs of the RDM network were defined in advance considering the different foreseeable runtime contexts. Following the mapping rules of Section 4, the priorities are inserted as the initial rewards for the initialization of the runtime MR-POMDP++ model (as shown in Table 2). During the autonomous decision-making, in situations where it was considered appropriate by the MR-POMDP++, the predefined priorities were *tuned autonomously* with the computation of *expected utility values* for individual NFRs (using equation 7). Such considerations obey to keeping levels of satisfaction according to the *SLA*, by the support of the new priorities under the new contexts found. These new priorities correspond with the *expected utility values* shown in Table 5. The goal of this *autonomous tuning* with help of *expected utility values* is to meet the required *SLA* for the NFRs.

Case: 2-Impacts of Priority-Aware Autonomous Decisions on satisfaction levels of NFRs. We have also studied the impact of priority-aware decisions by MR-POMDP on satisfaction levels of NFRs under the detrimental conditions and, have compared its results with existing techniques that use DESPOT [17] and DESPOT-ARROW [18].

Let us observe Figures 2 and 3, which show the results of the solvers OLSAR and DESPOT under i) the initial set of pre-defined rewards, transition and observation probabilities (normal or stable conditions), and ii) the detrimental scenario where the disturbance levels are introduced. Under these scenarios, both OLSAR

and DESPOT show comparable results by maintaining the NFRs of MC, MR and MP in the suitable zone of satisfaction, i.e. above the threshold values of $P(MC=True) \geq 0.70$, $P(MR=True) \geq 0.85$ and $P(MP=True) \geq 0.75$. Both techniques show preference for the MST topology, with an increase in the use of MST topology by OLSAR under detrimental context to support the satisfaction of both MC and MP as shown in Fig. 4 and 5.

Furthermore, we have also compared our results with the technique using DESPOT and ARROW (based on P-CNP) [18], which offers the support of updating initially defined rewards for DESPOT under the detrimental contexts. However, even if the technique DESPOT-ARROW supports the satisfaction of MR under the detrimental context, it leads the levels of satisfaction of MC and MP to poor zones of satisfaction during several time steps as shown in Fig. 3. Hence, from the results observed, we can deduce that the priority-aware decision-making process of MR-POMDP++ offers higher levels of satisfaction even in the detrimental scenarios observed when compared to the existing single-objective techniques. Further, another downside of the technique DESPOT-ARROW is that the tuning offered is not autonomously executed by the POMDP as is the case of MR-POMDP++. The former creates problems of efficiency in the case of DESPOT-ARROW.

Discussion: From the results, it is evident that the approach of MR-POMDP++, using the priority-aware decisions, maintains the *SLA* for the RDM network both under the operation of the system under initially defined stable conditions and the detrimental conditions introduced at runtime. The average satisfaction levels for all the NFRs of MC, MR and MP, generated by OLSAR, under initial stable conditions are 0.8732, 0.9069 and 0.8747 respectively. For the detrimental scenario, the average satisfaction levels are $P(MC=True) = 0.8756$,

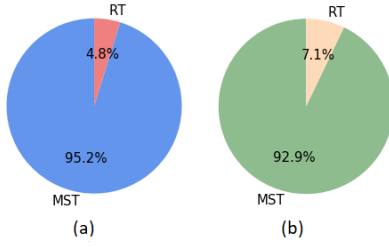


Figure 4: Topologies Selection under Stable Context a) OLSAR b) DESPOT

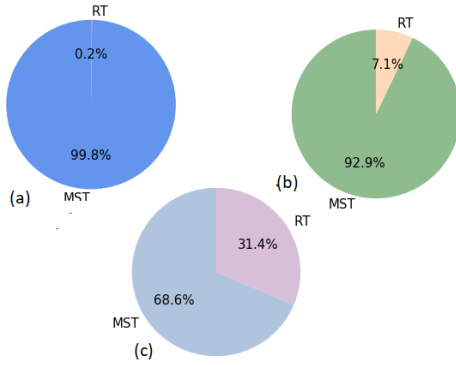


Figure 5: Topologies Selection under Detrimental Context a) OLSAR b) DESPOT c) DESPOT-ARROW

$P(MR=True)=0.9042$ and $P(MP=True)=0.8811$. Hence, the system is conforming to the required SLAs of $P(MC = True) \geq 0.70$, $P(MR = True) \geq 0.85$ and $P(MP = True) \geq 0.75$ as shown in Fig. 6. In order to further evaluate our approach, we have also executed other experiments considering other detrimental scenarios, which are reported in [32].

Under all the scenarios provided in [32], our approach shows comparable and sometimes even better satisfaction levels for NFRs than DESPOT and DESPOT-ARROW which are representatives of a single objective approaches. Hence, on the basis of the results, we can deduce that our approach using multiple rewards provides statistically better results by providing more *awareness* to the decisions in terms of NFRs' priorities as compared to the single objective approaches.

6 THREATS TO VALIDITY

A threat to validity posed by the use of MR-POMDPs is the computational cost. It is known that solving a MR-POMDP is often a computationally intractable problem at worst [23]. The OLSAR algorithm that we use, overcomes computational scalability issues related to the "curse of history" [24]. However, in a MR-POMDP++, the states are defined in terms of combinations of satisfaction levels of NFRs. If the number of NFRs is 2, we would have 4 states for MR-POMDP, for 3 NFRs, it would be 8, and so on. Therefore, the number of NFRs that can be handled by the approach of the MR-POMDP++, is limited. It hints to the fact that it faces a limitation in terms of a

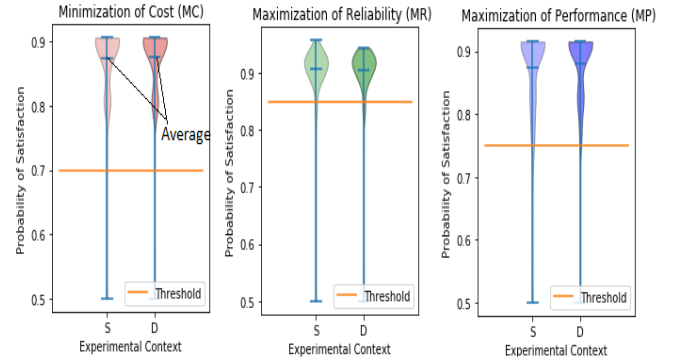


Figure 6: Average Satisfaction of NFRs under different Environmental Contexts—Stable (S) and Detrimental (D)

maximum number of NFRs. While this limit exists, MR-POMDP++ is scoped to domains that need multi-objective sequential decision-making mechanisms with a focus both theoretically and practically on few objectives [23]. Nevertheless, further research is needed to understand the extent to which this will prove to be a limitation to generalizability.

7 CONCLUSION AND FUTURE WORK

We have presented MR-POMDP++, a priority-aware model, to perform decision-making under uncertainty for AISs [22, 27, 29]. The use of MR-POMDP++ underpins awareness and reasoning about the distinct priorities of NFRs and allow tuning of priorities of the NFRs to match newly found contexts. For the purpose of evaluation of the proposed approach, we have applied it to a robust case study called RDM network, and have also compared the results with the techniques of DESPOT and DESPOT-ARROW, which are techniques that treat the NFR priorities using a single utility value. Our experiments have demonstrated that, based on empirical data gathered at runtime, the use of MR-POMDP++ offers the following contributions:

- provides support for priority-aware decision-making for AIS, which takes into account the individual priorities of NFRs during the decision-making process.

- provides an AIS with a principled approach to maintain the required SLA by autonomously tuning the priorities of NFRs at runtime in order to match the changing environmental contexts.

In future, we plan to use MR-POMDP++, as a tool for the *a priori*-elicitation of priorities for NFRs. Specifically, MR-POMDP++ can be used during simulations to further learn about the environment of the AIS, and to uncover unforeseen contexts that may be difficult otherwise, to envisage in advance. Further, as the approach helps to discover further knowledge about a system's behaviour and its environment based on runtime evidence, it can also be used to provide post-hoc explanations for AISs' behaviour [16].

ACKNOWLEDGMENTS

This work has been partially supported by The Lerverhulme Trust Fellowship "QuantUn: quantification of uncertainty using Bayesian

surprises" (Grant No. RF-2019-548/9) and the EPSRC Research Project Twenty20Insight (Grant No. EP/T017627/1).

REFERENCES

- [1] Marco Abundo, Valeria Cardellini, and Francesco Lo Presti. 2011. An MDP-based admission control for a QoS-aware service-oriented system. In *2011 IEEE Nineteenth International Workshop on Quality of Service*. IEEE, 1–3.
- [2] Nelly Bencomo and Amel Belaggoun. 2013. Supporting decision-making for self-adaptive systems: from goal models to dynamic decision networks. In *REFSQ*. Springer, 221–236.
- [3] Kate M Bowers, Erik M Fredericks, and Betty HC Cheng. 2018. Automated Optimization of Weighted Non-functional Objectives in Self-adaptive Systems. In *International Symposium on Search Based Software Engineering*. Springer, 182–197.
- [4] Javier Cámara, Antónia Lopes, David Garlan, and Bradley Schmerl. 2016. Adaptation impact and environment models for architecture-based self-adaptive systems. *Science of Computer Programming* 127 (2016), 50–75.
- [5] Bihuan Chen, Xin Peng, Yijun Yu, and Wenyun Zhao. 2014. Requirements-driven self-optimization of composite services using feedback control. *IEEE Transactions on Services Computing* 8, 1 (2014), 107–120.
- [6] Rogério De Lemos, David Garlan, Carlo Ghezzi, Holger Giese, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Danny Weyns, Luciano Baresi, Nelly Bencomo, et al. 2017. Software engineering for self-adaptive systems: Research challenges in the provision of assurances. In *Software Engineering for Self-Adaptive Systems III. Assurances*. Springer, 3–30.
- [7] Erik M. Fredericks. 2015. Mitigating uncertainty at design time and run time to address assurance for dynamically adaptive systems.
- [8] Martin Glinz. 2007. On non-functional requirements. In *RE (2007)*. IEEE, 21–26.
- [9] Heather J Goldsby, Pete Sawyer, Nelly Bencomo, Betty HC Cheng, and Danny Hughes. 2008. Goal-based modeling of dynamically adaptive system requirements. In *15th annual IEEE international conference and workshop on the engineering of computer based systems (ecbs 2008)*. IEEE, 36–45.
- [10] Minwen Ji, Alistair C. Veitch, and John Wilkes. 2003. Seneca: remote mirroring done write. In *USENIX Annual Technical Conference, General Track*.
- [11] Kimberly Keeton, Cipriano Santos, Dirk Beyer, Jeffrey Chase, and John Wilkes. 2004. Designing for Disasters. (March 2004).
- [12] Rogério de Lemos (Ed.). 2013. *Software engineering for self-adaptive systems II: international seminar, Dagstuhl Castle, Germany, October 24–29, 2010: revised selected and invited papers*. Number 7475 in LNCS. Springer, Berlin ; New York. OCLC: ocn839358754.
- [13] Sotirios Liaskos, Sheila A. McIlraith, Shirin Sohrabi, and John Mylopoulos. 2011. Representing and reasoning about preferences in requirements engineering. *REJ '11* 16, 3 (Sept. 2011), 227–249.
- [14] Martina Maggio, Alessandro Vittorio Papadopoulos, Antonio Filieri, and Henry Hoffmann. 2017. Automated control of multiple software goals using multiple actuators. In *Proceedings of the 2017 11th joint meeting on foundations of software engineering*. 373–384.
- [15] Gabriel A Moreno, Javier Cámara, David Garlan, and Bradley Schmerl. 2018. Flexible and efficient decision-making for proactive latency-aware self-adaptation. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 13, 1 (2018), 1–36.
- [16] Juan Marcelo Parra-Ullauri, Antonio García-Domínguez, Luis Garcia, and Nelly Bencomo. 2020. Temporal Models for History-Aware Explainability. In *Proceedings of SAM 2020*.
- [17] Luis H. Garcia Paucar and Nelly Bencomo. 2018. RE-STORM: mapping the decision-making problem and non-functional requirements trade-off to partially observable markov decision processes. In *SEAMS '18*. 19–25.
- [18] Luis H. Garcia Paucar, Nelly Bencomo, and Kevin Kam Fung Yuen. 2019. ARRoW: automatic runtime reappraisal of weights for self-adaptation. In *SAC '19*. ACM Press, Limassol, Cyprus, 1584–1591.
- [19] Xin Peng, Bihuan Chen, Yijun Yu, and Wenyun Zhao. 2010. Self-Tuning of Software Systems Through Goal-based Feedback Loop Control. In *Conf. RE '10*. IEEE, Sydney, Australia, 104–107.
- [20] Andres J. Ramirez and Betty H. C. Cheng. 2011. Automatic Derivation of Utility Functions for Monitoring Software Requirements. In *MODELS (LNCS)*. Springer Berlin Heidelberg, 501–516.
- [21] Andres J Ramirez, Adam C Jensen, Betty HC Cheng, and David B Knoester. 2011. Automatically exploring how uncertainty impacts goal satisfaction. *Department of Computer Science, Michigan State University, East Lansing, Michigan, Tech. Rep. MSU-CSE-11-14* (2011).
- [22] Berrevoets Robbe and Weyns Danny. 2018. A qos-aware adaptive mobility handling approach for lora-based iot systems. In *SASO (2018)*. 130–139.
- [23] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. 2013. A Survey of Multi-Objective Sequential Decision-Making. *Journal of AI Research* 48 (2013).
- [24] Diederik Marijn Roijers, Shimon Whiteson, and Frans A. Oliehoek. 2015. Point-Based Planning for Multi-Objective POMDPs. In *IJCAI('15)*.
- [25] F. W. Roush. 1993. Multicriteria decision-aid. : Philippe Vincke, Chichester: John Wiley, 1992, 154 pages, [UK pound]29.95. *Mathematical Social Sciences* 25, 2 (February 1993), 204–204.
- [26] B. Roy. 1996. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht.
- [27] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. Finkelstein. 2010. Requirements-Aware Systems: A Research Agenda for RE for Self-adaptive Systems. In *IEEE Conf. RE (2010)*. 95–103.
- [28] Guy Shani, Joelle Pineau, and Robert Kaplow. 2013. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems* 27, 1 (2013), 1–51.
- [29] Vítor E. Silva Souza, Alexei Lapouchnian, William N. Robinson, and John Mylopoulos. 2011. Awareness requirements for adaptive systems. In *SEAMS(2011)*. ACM Press, 60.
- [30] Harold Soh and Yiannis Demiris. 2011. *Multi-Reward Policies for Medical Applications: Anthrax Attacks and Smart Wheelchairs*.
- [31] M. T. J. Spaan and N. Vlassis. 2005. Perseus: Randomized Point-based Value Iteration for POMDPs. *Journal of Artificial Intelligence Research* 24 (Aug. 2005), 195–220. arXiv: 1109.2145.
- [32] xyz. 2020. *Remote Data Mirroring-Experiments*. Technical Report v0.9. https://gitlab.com/re_research/rdmexperiments/
- [33] Kevin Kam Fung Yuen. 2014. The Primitive Cognitive Network Process in health-care and medical decision making: Comparisons with the Analytic Hierarchy Process. *Applied Soft Computing* 14 (Jan. 2014), 109–119.