Understanding Uncertainty in Self-adaptive Systems

Radu Calinescu
University of York, UK
radu.calinescu@york.ac.uk

Raffaela Mirandola
Politecnico di Milano, Italy
raffaela.mirandola@polimi.it

Diego Perez-Palacin
Linnaeus University, Sweden
diego.perez@lnu.se

Danny Weyns
KU Leuven, Belgium
danny.weyns@kuleuven.be

***Abstract*—Ensuring that systems achieve their goals under uncertainty is a key driver for self-adaptation. Nevertheless, the concept of uncertainty in self-adaptive systems (SAS) is still insufficiently understood. Although several taxonomies of uncertainty have been proposed, taxonomies alone cannot convey the SAS research community's perception of uncertainty. To explore and to learn from this perception, we conducted a survey focused on the SAS ability to deal with unanticipated change and to model uncertainty, and on the major challenges that limit this ability. In this paper, we analyse the responses provided by the 51 participants in our survey. The insights gained from this analysis include the view—held by 71% of our participants—that SAS can be engineered to cope with unanticipated change, e.g., through evolving their actions, synthesising new actions, or using default actions to deal with such changes. To handle uncertainties that affect SAS models, the participants recommended the use of confidence intervals and probabilities for parametric uncertainty, and the use of multiple models with model averaging or selection for structural uncertainty. Notwithstanding this positive outlook, the provision of assurances for safety-critical SAS continues to pose major challenges according to our respondents. We detail these findings in the paper, in the hope that they will inspire valuable future research on self-adaptive systems.**

***Index Terms*—Self-adaptation, uncertainty, unanticipated change, models, modeling formalism, survey**

I. Introduction

Self-adaptation enhances a system with an external feedback loop that tracks the state of the system and, through adapting it to internal and environmental changes, ensures that a set of goals is consistently achieved [1]–[3]. A classic example is a service-based system whose feedback loop dynamically selects services that keep the failure rate below a required threshold, while also minimizing cost. Multiple terms have been used to refer to such systems, including *autonomic systems* [4], *dynamic adaptive systems* [5], and *self-adaptive systems* [6]. We will use the last term in our paper.

Self-adaptation was introduced about two decades ago as a means to manage the growing complexity of computing systems [4], [7]. While the initial focus was on automating the complex task of system operators, about a decade ago researchers and engineers started to realise that the presence of uncertainty is a central aspect of self-adaptation [8].

Self-adaptation introduces a blur between traditional offline activities performed by engineers and online activities performed by the system [9], [10]. In particular, a self-adaptive system (SAS) can be considered as a partially completed system with some degrees of freedom in terms of its configuration. This allows the SAS to adapt its configuration when the conditions change in order to deal with uncertainties that are difficult or impossible to anticipate before deployment. At runtime, the system collects additional information to resolve the uncertainty and adapt itself to preserve its goals.

Unfortunately, uncertainty is a complex concept that is difficult to understand, let alone to manage. Over the years a number of researchers have proposed initial taxonomies of uncertainty for self-adaptive systems [11]–[14]. While these taxonomies have been instrumental in putting the focus on uncertainty as a key driver for self-adaptation, the SAS research community's perception of what constitutes uncertainty remains unclear. As an illustration, a common topic of debate among members of the community is the extent to which self-adaptive systems deal with *unanticipated change*. Some people argue that no human-made system can handle unanticipated phenomena, while others argue that dealing with unanticipated change is exactly the key challenge of self-adaptation. Clarifying such differences in opinion is crucial for the community.

Our paper aims at shedding light on the perception of the community on the notion of uncertainty in self-adaptive systems. To this end, we conducted a survey about (1) the ability of systems to deal with unanticipated change, (2) the representation of uncertainty in SAS models, and (3) the challenges of handling uncertainty for systems with strict requirements. The 51 survey participants are actively involved in research on self-adaptation in the broader community. This paper presents the results of our survey, and is organized as follows.

We briefly summarize existing taxonomies for uncertainty in self-adaptive systems (Section II) and the scientific method we used in this research (Section III). We then report the analysis of the data collected (Section IV) and discuss the insights obtained (Section V). Finally we discuss threats to validity (Section VI) and wrap up with an outlook for future research in this area (Section VII).

II. Related studies

The notion of uncertainty has been studied in a wide variety of fields, usually in connection to decision-making; a recent example is [15]. Most of these studies assume that decision-making processes are "executed" by humans. However, in self-adaptive systems, the decisions are primarily made by software. This requires a fresh and innovative approach to the problem of decision-making under uncertainty.

Over the past years, the number of studies that take into account uncertainty in self-adaptive systems has gradually been increasing. A typical example is the use of probabilistic runtime models, such as Markov decision processes [16], [17] and parametric stochastic models [18], [19] to reason about change when making adaptation decisions. As progress is taking place, a more systematic understanding of SAS uncertainty is required.

Hereafter, we summarize a representative set of studies on the notion of uncertainty in self-adaptive systems and conclude with positioning the work presented in this paper.

Taxonomies: Ramirez et al. [11] provide a definition and taxonomy for uncertainty in dynamically adaptive systems. The taxonomy classifies sources of uncertainty for the requirements, design, and runtime phases of dynamically adaptive systems. The uncertainties are described using a template inspired by the established template for representing design patterns (name, classification, context, impact, mitigation strategies, sample illustration, related sources). Perez-Palacin et al. [13] present a taxonomy for uncertainty in SAS modeling that comprises three key dimensions: location, level, and nature. The location of uncertainty refers to the model aspects affected by the uncertainty. The level of uncertainty indicates where the uncertainty is placed on the spectrum between deterministic knowledge and total ignorance. Finally, the nature of uncertainty shows whether the uncertainty is due to the imperfection of the acquired knowledge or to the inherent variability of the modelled phenomena.

Literature Review: Mahdavi et al. [14] performed a systematic literature review on uncertainty in self-adaptive systems with multiple requirements. From the data collected from 51 primary studies the authors derive a systematic overview of uncertainty dimensions (location, nature, level/spectrum, emerging time, sources) with their respective options. The sources of uncertainty are further elaborated and are grouped into several classes (i.e., uncertainty of models, adaptation functions, goals, environment, resources, and managed system).

Others: Garlan [8] discusses several sources of uncertainty affecting modern software systems (humans in the loop, learning, mobility, cyber-physical systems, rapid evolution), and argues that uncertainty in software systems should be considered as a first-class concern throughout the whole system life cycle. Esfahani and Malek [12] study uncertainty in self-adaptive systems with an emphasis on sources of uncertainty that include: simplifying assumptions, model drift, noise, parameters in future operation, human in the loop, objectives, decentralization, context, and cyber-physical systems. Their study also investigate uncertainty characteristics (reducibility versus irreducibility, variability versus lack of knowledge, and spectrum of uncertainty).

Conclusion: Previous research has studied the notion of SAS uncertainty based on existing research literature and on individual projects and experiences. Our paper complements these important efforts by presenting insights on the SAS research community's *perception* of the notion of uncertainty in self-adaptive systems.

III. Research Method

To shed light on the research community's *perception* of uncertainty in self-adaptive systems, we carried out a survey, which is an empirical method where data are collected from a population using a questionnaire [20]. The population for the survey comprises those who perform research on uncertainty in self-adaptive systems and validate the results of this research in concrete systems. To obtain a representative sample, we gave the questionnaire to the researchers attending the main SAS conferences, and complemented this with direct email invitations sent to additional SAS experts. All survey participants were researchers with experience in dealing with uncertainty in self-adaptive systems.

We performed the survey following the guidelines by Kitchenham et al. [21]. In this section, we explain the research questions, and then summarize the methodological steps of the survey: planning, conducting, analyzing, and documenting.

A. Research Objective

The overall goal of this research has been to obtain insights into the perception of the community on the notion of uncertainty in self-adaptive systems. This goal translates to three research questions to be answered by the survey:

- RQ1 (Dealing with uncertainty): *What is the perception of the community on the ability of self-adaptive systems to deal with changes that were not anticipated when the systems were engineered?*
- RQ2 (Representation of uncertainty): *What is the perception of the community on the representation of uncertainty in runtime models and the paradigms used for modeling?*
- RQ3 (Challenges of uncertainty): *What are the challenges for uncertainty in self-adaptive systems perceived by the community, in particular for systems with strict goals?*

With RQ1, we wanted to gain insight into the scope of uncertainty, i.e., into the extent to which a self-adaptive system can handle changes not anticipated before the system deployment. With RQ2, we wanted to understand how uncertainty can be represented in runtime models and what the restrictions are. Finally, with RQ3 we wanted to gain insight into the main challenges researchers see with respect to dealing with uncertainty, in particular for systems with strict requirements.

B. Planning the Survey

We used a cross-sectional survey [21] with a questionnaire that we delivered to the participants personally or by email.

After defining the research questions, we designed the questionnaire to cover the different aspects of uncertainty targeted by these questions. To that end, we included: (i) closed questions with one or more choices complemented with a text box where respondents could elaborate on their choice using free text; and (ii) open questions that respondents could answer with free text. All questions were optional.

The questionnaire comprised in total seven questions. To answer RQ1, we formulated two questions; one aimed at understanding the perception of the respondents on the ability of self-adaptive systems to deal with unanticipated changes;

and the other one to gain insight into how the system may be able to gain awareness of change that it was not engineered for. To answer RQ2, we formulated four questions. With the first of these questions, we aimed at understanding the perception of the respondents on the aspects of SAS runtime models that can be associated with uncertainties. The following three questions then zoomed in on uncertainties in model parameters, the model structure, and the modeling formalism. Finally, to answer RQ3, we formulated a last question that aimed at gaining insight into the perception of the respondents on open challenges in handling uncertainty in self-adaptive systems with safety-critical requirements.

The questionnaire has been designed over several iterations. An initial set of questions was defined by three of the authors in a face-to-face meeting. The fourth author then checked the questionnaire, and proposed a number of refinements plus an additional question. The revised questions were then discussed among the four authors. After several adjustments, the questionnaire was finalized for release.

C. Conducting the Survey

We collected data using a combination of direct and indirect methods [20]. In particular, we distributed the questionnaire to the attendees at 2019 editions of the main venues for the SAS research community: SEAMS (International Symposium on Software Engineering for Adaptive and Self-Managing Systems), ICAC (International Conference on Autonomic Computing) and SASO (International Conference on Self-Adaptive and Self-Organizing Systems).[1] Additionally, we distributed the questionnaire to the participants at the Shonan seminar on "Controlled Adaptation of Self-adaptive Systems" (CASaS) in January 2020. Each of these events was attended by at least one of the authors. To enhance validity, we have sent personally invitations via email to several additional experts of the community, inviting them to complete the questionnaire.

All respondents were researchers with experience in dealing with uncertainty in self-adaptive systems. The sample included PhD students, postdoctoral researchers, and academics ranging from assistant professor to full professor. The respondents completed printed copies of the two-page questionnaire by hand. One of the survey authors then copied all the answers into a spreadsheet for analysis.

D. Data Analysis

To analyze the data collected from the answers with options, we used simple descriptive statistics. In particular, for each question, we determined the percentages of the different response options. We then complemented these results by analysing the comments provided by the respondents. To that end, we applied simple qualitative data analysis using coding. This type of analysis enables identifying patterns and relationships between the data [20], [22]. The coding was performed using the following steps:

1) *Extracting data*: we read and examined the data from the questions that allowed comments, and the answers to the open questions.
2) *Coding data*: we did not define any coding upfront; instead we analyzed the data and incrementally added codes to small coherent fragments of the text provided in different answers (as suggested in [23]).
3) *Translating codes into categories*: starting from the codes we then derived categories through an abstraction step where the different codes were thematically grouped.

To avoid bias in the identification of codes and the synthesis in categories, we performed both steps for each question in a team of two authors. Both authors worked independently and then exchanged their results. Differences where then discussed until consensus was reached. Finally, the other authors crosschecked the results to finalize the coding.

E. Documenting the Survey Results

The results of the survey are documented in this paper that was jointly produced by the four authors. All material of the survey is available online.[2]

IV. Results

We collected 51 completed questionnaires distributed as follows: 11 from SEAMS, 15 from ICAC/SASO, 11 from CASaS, and 14 from additional experts. In this section, we report the results of the analysis of the raw data for each research question and conclude with key findings. In the next section, we further discuss and interpret the results.

A. RQ1: Dealing with uncertainty

To address this research question, we analysed the participants' answers to the two survey questions shown in Table I.

RQ1a: Handling unanticipated changes. The first question asked for the participants' view on the possibility that SAS may only be able to deal with anticipated changes. Only 29% of those surveyed held this view, with a majority of 71% of our respondents deeming that SAS would be able to deal with (at least some level) of unanticipated changes.

Asked to explain their position, those who considered SAS unable to handle unanticipated changed suggested two main reasons for this (Table II):

- Unless a system is *built* to deal with a specific type of change from the outset, it will not be able to handle it;
- Unless a change is anticipated, a system will not be able to *monitor* its occurrence.

In contrast, the respondents who disagreed that SAS could only deal with anticipated changes held the views that (Table II):

- As a matter of *principle*, SAS ought to be able to handle unanticipated changes too;
- SAS can handle unanticipated changes *conditional* on their extent, frequency, etc. staying within certain limits;

[1] After their 2019 editions, ICAC and SASO merged into ACSOS.

[2] https://people.cs.kuleuven.be/danny.weyns/surveys/uncertainty/index.htm

TABLE I

SURVEY QUESTIONS ANALYSED TO ANSWER RQ1

ID	Question	Responses (out of 51 participants)
RQ1a	Self-adapting systems can deal only with anticipated changes. Self-adapting systems cannot deal with unanticipated changes.	Agree 15 Disagree 36
RQ1b	If you selected “Disagree” as answer for Question 1, please explain how the system may be able to gain awareness of the occurrence of a change that it was not engineered to anticipate.	

- SAS can handle unanticipated changes as long as learning about them, e.g., with human support or by employing evolutionary techniques, is feasible.

A broad spectrum of approaches that have the potential to allow SAS to deal with unanticipated changes have been proposed by these respondents. As shown in the last part of Table II, these approaches ranged from software “evolution”, (machine) learning and genetic techniques to the online synthesis (of “coping” strategies), runtime modelling, generalisation, and optimization-driven decision making.

RQ1b: Gaining awareness of unanticipated changes. This question was posed to the survey participants who indicated that SAS should be able to deal with unanticipated changes, asking them to suggest methods that these systems can use to *gain awareness* that such a change occurred. Their answers, summarised in Table III, identified three important categories of concerns associated with SAS gaining awareness of unanticipated changes:

- Unanticipated change awareness – different criteria can be used to decide whether a SAS is actually aware of an unexpected change. Five participants suggested that simply noting the presence of unexpected symptoms should be regarded as awareness that an unexpected change has taken place. In contrast, six participants insisted that awareness could only be claimed once the cause of the unexpected change was identified by the SAS. Finally, six additional respondents recognised the importance of deciding what it means for a SAS to be aware of an unanticipated change without specifying how this decision could be taken.
- Unanticipated change identification – a range of methods for identifying unexpected changes were suggested by the 36 participants whose answers to question RQ1a indicated that dealing with such changes should be feasible for SAS. The most frequently suggested methods (each mentioned by five participants) are: monitoring deviations in the system parameters; observing the consequences of changes; noticing a mismatch between the SAS internal models and runtime observations; and analyzing historical data collected through monitoring the SAS. A few other methods were also suggested: noticing the lack of knowledge/understanding of the SAS status (mentioned by four participants); identifying broad change classes (proposed by two participants); and receiving information from a human (indicated by one participant).
- Reacting to unanticipated changes – three main classes of methods have been suggested. First, nine respondents indicated that existing SAS actions for dealing with expected changes should be *adapted* to deal with the unexpected change, e.g., by applying evolutionary approaches to the available set of actions. Second, four respondents proposed the *synthesis* of (completely) new such actions, although no clear approach to achieve this was suggested. Finally, three respondents indicated that using a *default*, fail-safe action could allow SAS to deal with unanticipated changes, albeit in an over-conservative way.

Key findings from RQ1:

- Over two thirds of the survey participants hold the view that self-adaptive systems can be engineered to cope with some level of unanticipated changes.
- The research community has mixed views on whether a SAS can be deemed aware of unanticipated changes when their symptoms are observed or only when the cause for these symptoms is identified.
- Three types of SAS reactions to unanticipated changes were proposed: adapt existing actions, synthesise new ones, or just use a default fail-safe action.

B. RQ2: Representation of uncertainty

To answer this research question, we analysed the participant answers to the survey questions shown in Table IV.

RQ2a: Uncertainties in model elements and formalism. The first question that we analysed explored the respondents’ view on the characteristics of runtime models that uncertainties can be associated with. High percentages of 96% and 94% of the respondents held the views that these characteristics included the model parameters and structure, respectively. Additionally, 82% of the respondents deemed the modelling formalism potentially unable to capture relevant aspects of the SAS and its environment. All these 42 respondents also selected model parameters and structure in their answers, leaving only nine participants who did not agree with all three concrete options suggested in question RQ2a. The question also allowed the participants to provide additional comments on uncertainty sources in a free-text box. The uncertainty sources mentioned in these comments (Table V) can be organised into four categories:

- Modelling limitations – Ten respondents ascribed sources of uncertainty to modeling limitations. Out of these, six participants identified modelling constraints and four modelling assumptions as main causes of problems.
- Monitoring limitations – A second group of five respondents identified as monitoring limitations the origin of uncertainty, where the limitation can be located in the scope of the monitoring (three answers) or in the monitoring process itself (two answers).

TABLE II
QUALITATIVE ANALYSIS OF EXPLANATIONS FROM RQ1A

Categories & codes	#	Example quote(s)
Reasons to agree†		
Not built for	6	"System (instances) cannot adapt to changes for which they have not been built (designed, prepared for)," "any capability to adapt to new situations where no explicit action is provided must have been built into the system from the beginning"
Non-monitorable	9	"If it is not anticipated, the system cannot monitor the issue", "The possibility of change should be in some way already present in the system."
Reasons to disagree†		
Matter of principle	7	"Self-adapting system SHOULD adapt to unanticipated changes in some manner," "I disagree in principle, but I don't think we have yet reached this goal as fully as possible."
Conditional	6	"it will depend on the kind of unanticipated changes, their extent, their frequency, etc. No system will adapt to anything anytime," "It depends on what changes and reactions one wants to consider. If the reaction is always the same, then any change can be considered"
Propositive	23	"Learning approaches could allow the systems to learn new information about unanticipated changes especially if this happens with a 'human in the loop' approach", "If you can detect the consequences of the changes, you might be able to cope with that using genetic techniques"
Unanticipated change support		
Evolution	4	"System (instances) cannot adapt to changes for which they have not been built (designed, prepared for). They require software evolution."
Learning	9	"a self-adaptive system should learn during runtime, so it should be able to deal with unexpected changes (to some extent-it depends on the knowledge base"
Genetic techniques	4	"If you can detect the consequences of the changes, you might be able to cope with that using genetic techniques."
Online synthesis	2	"It depends on the capabilities of the system. The system may be able to recognize an unknown situation and synthetise a way to cope with it"
Modelling	5	"if unanticipated changes are reflected to model, SAS can deal with them"
Generalization	2	"Generalization capabilities of utilized algorithms, for example."
Decision making	2	"the adaptation should be seen as an optimization problem and not a selection between predefined plans, No rules - mathematical optimization"
Others	5	"I imagine a system that dynamically discovers a new sensor and uses the input to react to changes of the environment which it was not able to detect before."

†that SAS cannot deal with unanticipated changes

- Novelty – Three participants attributed to the novelty of phenomena the cause of uncertainty.
- External entities – The intervention of external malicious entities or humans, has been indicated by five respondents as a major source of the uncertainty.

TABLE III
QUALITATIVE ANALYSIS OF EXPLANATIONS FROM RQ1B

Categories & codes	#	Example quote(s)
Defining awareness		
Symptoms observed	5	"the change can sometimes be anticipated indirectly by affecting on other features/behaviors, i.e. [...] its partial consequence can be anticipated"
Cause identified	6	"a learning module could discover the correlation between a certain change in the environment and some bad behavior of the system and learn from this"
Unspecified	6	–
Unanticipated change identification		
Monitor parameter deviations	5	""drop in the system utility", "sensors are not necessary limited to detect the consequences of 'anticipated' changes"
Observe consequences	5	"multiple factors can [make] a robotic system lose its ability to make a right turn [and] it may be enough to understand the change rather than its root cause", "measuring the effect of an uncertain variable without measuring the variable directly"
Internal model mismatch	5	"having a model [...] and checking it; mismatch can indicate [unanticipated] change", "initial model can be partially wrong/incomplete"
Analyze history	5	"examining historical patterns among data/behaviors"
Unknown current status	4	"no matching rule in the knowledge base"
Identify change class	2	"predict 'classes' of likely changes carrying common characteristics and requirements for adaptation"
Human support	1	"a human in the loop could give the system awareness of the change"
Unanticipated change reaction		
Adapt existing actions	9	"genetic algorithms could search [for] plans similar to what the SAS knows, [and] apply [them] to new circumstances", "[use] cross-learning [...] i.e. learn from similar systems to improve handling [of] changes"
Synthesize new actions	4	"[in] a situation for which [the SAS] has no solution, it reaches an exception state and [...] synthesizes a completely new adaptation", "engineering of systems at design time that will have the ability to autonomously and independently modify themselves [...] to successfully cope with the [unanticipated] changes at runtime"
Use default action	3	"general reactions [that] can solve any issue", "driving itself off in case of unforeseen [change]"

Interestingly, only two respondents identified other model characteristics that uncertainties can be associated with:

- One of these states that '*One needs to distinguish between configuration (parameter, structure) and monitoring results (values and structures)*,' emphasizing the distinction between the system state and the properties monitored.
- The other respondent mentions that '*no model is correct at any time. the best you can do is good enough decisions, soon enough to matter*,' pointing out that models cannot always represent real-world phenomena correctly.

RQ2b: Handling uncertainties in the parameters used by the system. This question probed the respondents' view on

TABLE IV
SURVEY QUESTIONS ANALYSED TO ANSWER RQ2

ID	Question	Responses (out of 51 participants)
RQ2a	Assuming that the knowledge a self-adaptive system collects and generates (about the system it manages and its environment) is represented as a runtime model, then uncertainties in such a model can be associated with (*select all that apply*):	Model parameters 49 Model structure 48 Modeling formalism[1] 42 Other (please specify) 22
RQ2b	Uncertainties in the values of model parameters can be dealt with by expressing them using (*select all that apply*):	Concrete values[2] 26 Intervals of values 36 Probabilities & probability distributions 44 Combination of intervals & probabilities[3] 46 Other (please specify) 13
RQ2c	Uncertainties in the model structure (i.e., elements in the model or parts of the model) can be handled by (*select all that apply*):	Using multiple models and applying: 42 - model averaging[4] 25 - model selection[5] 33 Alternative structures within a model[6] 36 Other (please specify) 14
RQ2d	When the uncertainty comes from the modeling formalism that does not allow capturing all relevant aspects of the system, it is not possible to handle this uncertainty at runtime.	Agree 28 Disagree (why?) 23

[1]i.e., uncertainties can be associated with the inability of the modeling formalism to capture the relevant aspects of the system
[2]estimated at design time, or determined at runtime; [3]e.g., confidence intervals; [4]i.e., combining several models into one model;
[5]i.e., selecting one model among different models based on some criteria; [6]i.e., alternative structures with associated probabilities;

possible ways to express uncertainty in the parameters of models. The most selected method from those that we proposed was, with 90% of respondents, the combination of intervals and probabilities. When these methods appear separated, the probabilities and intervals options alone were selected by 86% and 71% of respondents, respectively. The method of concrete values was selected by 51%. One respondent pointed out that the addition of a concrete value that is used as tolerance transforms '*concrete values*' into '*intervals of values*'.

Thirteen of the respondents provided additional methods to handle uncertainty. Table VI reports them, organised into two categories: methods to reduce the uncertainty in parameter values and methods to express the lack of certainty.

- The methods to reduce uncertainty in parameter values included the use of domain knowledge (mentioned by three respondents), the continuous search and refinement at runtime to increase the level of certainty (suggested by three respondents), and the use of different sources of data for each parameter (proposed by two respondents).
- Those who proposed methods for expressing the lack of certainty suggested the use of: sensitivity and stability analysis; and relations between the values of parameters and relevance of the parameters.

The use of a free-text box in this question allowed us to observe the lack of community consensus on the uses of SAS models. A respondent questioned here the necessity of a model and a modelling formalism, and also stated the belief in systems that adapt without previously determining what they can adapt to.

RQ2c: Handling uncertainties in the model structure. The third question collected the opinions of the participants about methods for handling uncertainties in the model structure. The question proposed two families of methods for models with uncertain structure: using multiple models and using alternative structures in the model. The two types of methods were selected by 82% and 71% of the respondents, respectively.

In addition, we split the utilisation of multiple models into two concrete methods: model averaging and model selection. Among the group of respondents who selected the utilisation of multiple models (42 out of the total of 51), the model selection method was chosen more often than the model averaging, with 79% and 60% of the answers, respectively. Interestingly, 23 out of the 42 respondents selected both options. This means that only two people selected exclusively model averaging, but 10 of them pointed out exclusively a model selection method.

A free-text box allowed respondents to specify other methods for handling model structure. Table VII summarises the 14 responses we received, organised into four groups:

- Online model discovery or online model learning was referred to by five respondents, and involves the runtime generation and modification of the model elements.

TABLE V
QUALITATIVE ANALYSIS – EXPLANATIONS OF ANSWERS TO RQ2A

Categories & codes	#	Example quote
Modelling limitations		
Modelling constraints	6	"The structure may be adjusted based on noisy models and models typically are not able to capture all aspects of the systems," "Complexity of model: scale, multi-dimensionality"
Modelling assumptions	4	"The more of the model we assume, the less resilient the system is to uncertainty,"Incorrect assumptions on the behavior of the system"
Monitoring limitations		
Scope of monitoring	3	"What your monitor can provide," "Values and structure based monitoring results (monitorable properties)."
Monitoring process	2	"Parameters may be tuned based on noisy measurements, errors related to the sensing process (resolution, accuracy)."
Novelty		
Novel phenomena	3	"Novel Stimuli e.g. obstacles, novel interactions, new requirements and priorities."
External entities		
Malicious entities	3	"Adversarial actors in the environment, reality (because all models leave out stuff that we believe doesn't matter and sometimes it does)"
Human involvement	2	"human in the loop error - multiple ownerships."

TABLE VI
QUALITATIVE ANALYSIS – EXPLANATIONS OF ANSWERS TO RQ2B

Categories & codes	#	Example quote
Reducing uncertainty		
Domain knowledge	3	"Starting guesses", "Design time estimation is also a realistic starting point (e.g., backed up by expert knowledge and knowledge of the domain)"
Continuous refinement	3	"Refinement by experimentation and reflection", "Mechanisms like metaheuristics [...] to search for or construct *new certainty*"
Redundancy	2	"redundancy", "multi-sensor data fusion"
Expressing uncertainty		
Through Analysis	2	"Sensitivity and stability analysis can support the concrete values"
Through Relationships	3	"Implications and other relations between parameters", "Value constraints and relevance and importance measures"

- Flexible models were considered in five answers. This group of methods uses models that can fit more than one structure at the same time, for instance, incomplete models, models of boundaries, or approximate models.
- Other proposed methods were based on a combination or aggregation or multiple models, but more advanced compared to the provided options. These methods include, for instance, model interpolation.
- Methods that employ multiple or multi-view models to capture probabilities or to compare their results.

RQ2d: Uncertainties in model formalism prevent handling them at runtime. The fourth question of the questionnaire related to RQ2 investigated the community belief about the impossibility of SAS to deal with uncertainties related to

TABLE VII
QUALITATIVE ANALYSIS – EXPLANATIONS OF ANSWERS TO RQ2C

Categories & codes	#	Example quote
Reducing uncertainty		
Online learning	5	"New model elements can be discovered at runtime", "Online model generation"
Flexible models	5	"Incomplete models", "A flexible model that does not fit a structure, but rather defines boundaries"
Models combination	3	"Interpolation that may be seen as some form of averaging but it may be different" and "subjective logic [...] can be considered for aggregating the models"
Multiple models	3	"Multi-view models capturing probabilities", "Comparing results of multiple models"

modelling formalisms. Analysing the answers (Table IV) we observed 45% of respondents agreeing with the statement and 55% of respondents claiming that uncertainties can be dealt with notwithstanding their origin in the modelling formalisms. The survey asked for explanations from participants.[3] Their explanations (Table VIII) can be organised into two categories:

- Uncertainty management – Eight respondents stated that in this case uncertainties can be only partially managed, while twelve respondents believed in the unlimited ability of SAS to deal with this type of uncertainty.
- Methods to handle uncertainties – Model evolution has been proposed by six participants, with four participants suggesting the possibility of online model adaptation, and two of the opinion that SAS can create new online models; Combination of techniques is the solution foreseen by seven respondents, through the combination of modelling formalisms (two answers) or the combination of different approaches (five answers). Eight participants proposed alternative techniques, including model-free approaches (two answers) and best-effort techniques (six answers).

Key findings from RQ2:

- Almost all members of the community agree that parameters and structure of models are key artifacts to capture uncertainties in self-adaptive systems.
- To handle uncertainty in models' parameters almost all the members of the community would adopt a combination of confidence intervals and probabilities.
- A large part of the community hold the opinion that uncertainty in models' structure can be dealt with using multiple models and then applying model averaging or model selection.
- Almost half of the community members hold the belief that when uncertainties are related to model formalisms, SAS are not able to handle them at runtime.

C. RQ3: Challenges of uncertainty

This question was answered by analysing the responses to the survey question shown in Table IX.

Challenges in handling uncertainties of safety-critical SAS. Among the proposed challenges, providing assurances that

[3]The questionnaire asked a clarification from participants that selected disagree, but besides 23 of those, also 6 participants that selected agree provided a clarification of their choice.

TABLE VIII
QUALITATIVE ANALYSIS – EXPLANATIONS OF ANSWERS TO RQ2D

Categories & codes	#	Example quote
Uncertainty management		
Partial	8	"I think it would still be possible to tackle uncertainties to a certain degree. I think it [...] depends on the case on hand" " it might not be possible to handle it perfectly, but it can affect some other aspects captured in the model. Can be handled partially",
Unlimited	12	"It is possible to handle the uncertainty, but the system has to have additional runtime mechanism to learn beyond the model initially provided' "feedback mechanism can deal with unforeseen events or with unmodelled dynamics"
Methods		
Models evolution		
Online model adaptation	4	"Parameter learning can be an answer here," "the controller of a SAS could in principle be able to recognize its limitations and resolve them by extending its own functionalities, e.g., via a genetic algorithm."
Online model acquisition	2	"model creation and adaptation," "the system can acquire the new formalism at runtime"
Techniques combination		
Modelling formalisms	2	"Alternatives modelling formalism can be used in a complementary manner"
Other techniques	5	"Take, for example human control in the loop to resolve situations that the model cannot capture yet", "The modelling formalism can be combined with another technique that could be triggered when the model that initially was conceived is not longer valid"
Alternative techniques		
No models	2	"Could be handled without a model using model-free [approach]", "preemptive mechanisms without explicit reasoning can be employed (e.g., moving target defense)"
Best effort	6	"Prepare to make non optimal decisions", "Systems are built with a (usually small) finite set of actions they can take.Often the best of this set can be picked with (very) incomplete information."

adaptation decisions are correct with respect to the goals was selected by 86% of participants, making self-adaptation proactive instead of reactive was selected by 55% of participants, 57% of participants selected integrating machine learning into the self-adaptation process, and ensuring the scalability of the self-adaptation was selected by 63% of participants.

This question received several comments from respondents in the free text box (49% of them filled the free text box). Table X reports the answers, organised into two categories: comments that refined or provided additional information to one of the challenges listed among the options in the questionnaire, and comments that pointed out other distinct challenges.

The comments that refined the suggested challenges in the questionnaire relate to system characteristics that self-adaptation should guarantee, and aspects that should be included in the research of systems with assurances and using machine learning. The main properties mentioned in comments on system assurances were the safety, timeliness, reliability, and trust. Across the answers, seven respondents urged for caution and potential risk when applying self-adaptation to safety-critical systems. For instance, respondents noted that deciding actions in novel contexts is risky for safety-critical systems both for reactive and proactive decisions, that the utilisation of machine learning hinders the correct behaviour of the system in first phases of its execution, and that machine learning complicates the computation of formal guarantees about the system behaviour.

The comments that pointed out other challenges for handling uncertainty in safety-critical self-adaptive systems are organised in three groups (the second half in Table X):

- Lack of understanding or incomplete knowledge of the environment – Respondents noted that knowledge is limited and that models are incomplete representations.
- Human in the loop – Deciding the most appropriate granularity of control operations for human operators is a challenge. Further, respondents emphasise the need for self-explainable systems, for instance for self-adaptive systems that work in cooperation with humans, and as part of the provision of assurances generated by the system.
- Ethical and moral aspects – Some decisions raise moral and ethical questions, which make the correct outcome undefined or, at least, not univocal. Therefore, these type of decisions must be taken by humans.

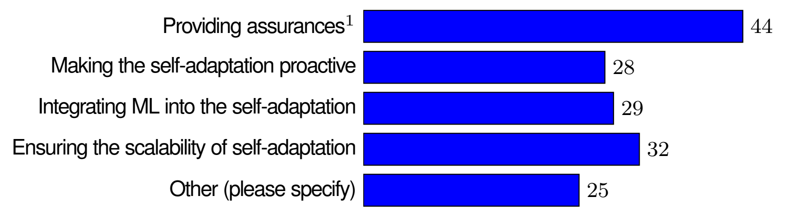
Key findings from RQ3:

- A large part of the community deems the assurances guarantee (for instance for safety, timeliness, reliability) as a key challenge for safety-critical self-adapting systems.
- Caution should be taken when applying self-adaptation in safety-critical systems, in particular regarding novel situations and in relation to the use of machine learning.
- Dealing with lack of knowledge, supporting humans in the loop, and dealing with ethical aspects are key challenges of safety-critical self-adapting systems.

V. DISCUSSION OF RESULTS

Unanticipated change. One of the key insights of our survey is the disagreement about the ability of self-adaptive systems to handle unanticipated change: 71% of the participants agree that self-adaptive systems can deal with unanticipated change, while the remaining 29% disagrees. Note that there is less contrast in the SASO community with 86% and 14%, compared to the rest with 60% and 40%. One of the participants that disagreed wrote: "I haven't seen proposal for creative thinking that enable systems to do things they were not engineered for [...] If they contact other components or services to find solutions to unexpected issues, then they were engineered to do that." This position conflicts with the statements from those agreeing that self-adaptive systems can handle unanticipated change. A major argument for agreeing lays in the use of machine learning and search-based techniques as expressed in "a self-adaptive system should learn during runtime, so it should be able to deal with unexpected changes" and "Our research on using GA to explore

TABLE IX
SURVEY QUESTION ANALYSED TO ANSWER RQ3

ID	Question	Responses (out of 51 participants)
RQ3	Handling uncertainty in safety-critical self-adaptive systems (e.g., self-driving cars) is difficult because of remaining open challenges associated with (*select all that apply*):	

[1] i.e., that self-adaptation decisions are correct with respect to the specified goals

TABLE X
QUALITATIVE ANALYSIS – EXPLANATIONS OF ANSWERS TO RQ3

Categories & codes	#	Example quote
Refinements of proposed challenges		
Providing assurances	11	"Ensuring the efficiency and reliability of the self-adaptation", "Providing assurances includes 3 aspects: timely assurances (in time for action), reachability/selection of solution in time (guarantee that a safety critical adaptation will converge), explanation of adaptation sufficient for the understanding+trust (collaborating agents, humans or machines)"
Utilization of machine learning	3	"The real challenge associated with ML here is to provide formal guarantees".
Caution and risk	7	"Both reactive and proactive decisions can be risky in novel contexts", "[with ML] the system will not behave correctly in these initial phases"
Other challenges		
Incomplete knowledge	6	"Lack of knowledge on how to handle it, i.e., incomplete model", "Lack of knowledge about environment", "Incompleteness of the knowledge and consequently of the models representing the knowledge"
Human in the loop	7	"Providing the right granularity, API, etc. for the human control is hard", "[...] self-explainablity of the system and the role of humans –that is even if the system does something super smart, it works in cooperation with humans which have no clue what the system does and why"
Ethical aspects	3	"Moral and ethical questions which are only to be answered by humans (e.g., the well-known dilemma about risking either the life of the driver of those of passer-by in a self-driving car", "Ethical reasons: what is the ground truth (correct outcome)?"

unknown unknowns shows you can adapt the adaptation to new circumstances." The current situation is probably best reflected in: "Most self-adaptive systems are partly designed using primitive adaptive mechanisms. They do have limitations [...] but there are much better methods that do not" and "maybe the current self-adapting systems deal with only anticipated change, but all self-adaptive systems need to have mechanisms to be able to deal with [unanticipated changes]." In conclusion, as the ability of self-adaptive systems to handle unanticipated change is subject of debate, the community would benefit from a principled discussion about this topic. This would improve our understanding of uncertainty and set the right expectations for what self-adaptive systems can handle and what is beyond their capabilities.

Support for unanticipated change. The survey participants provided a rich palette of potential methods to equip self-adaptive systems with support for handling unanticipated change. Four main groups of methods can be distinguished. The first group is software evolution, which is the traditional method to enhance a software system with new functionality. Integrating adaptation with evolution goes back to the pioneering work of Oreizy et al. [7]. The second group is modeling and abstraction. These methods highlight the need for modeling techniques that allow incorporating unanticipated change in some way in runtime models. This way, the feedback loop system will be able to reason about these changes and take them into account in the decision-making. The challenge here will be in equipping a modeling technique with the ability to incorporate change that was not anticipated. The third group is the use of online techniques to handle unanticipated change as exemplified by online synthesis. Synthesis techniques like [19], [24] can automatically produce a controller given a model of the target system, the set of controllable events, and the controller goal. Nevertheless, supporting online synthesis for unknown unknowns remains an open challenge. The fourth and final group is exploiting machine learning and genetic techniques. There is a strong belief that these approaches will push the abilities of self-adaptation beyond what we have been able to achieve so far. Several participants take even a stronger position as reflected in the statement "machine learning should be mandatory for appropriate adaptation." In contrast, other participants highlight that there is no free lunch, as expressed in "I believe the real challenge associated with ML here is how to provide formal guarantees of correctness, timeliness, safety, and other important qualities that these system should have." Concretely addressing the different challenges associated with these groups of methods will require substantial research.

Open challenges. The participants put assurances for self-adaptive systems that operate under uncertainty as the top challenge for the community, emphasizing the need to manage the dichotomy between uncertainty and guarantees. Other important challenges are proactive adaptation, integration with machine learning, and scalability. Emerging challenges are self-explainability and consideration of ethical aspects. Tackling these challenges will require an extensive joint effort across teams within the community, and collaboration with researchers from other disciplines.

VI. Threats to Validity

We assess threats to the validity of the study using the guidelines proposed in [25]. We focus on *construct validity* (extent to which we obtained the right measure and whether we defined the right scope in relation to the study goal), *external validity* (extent to which the findings can be generalized), and *reliability* (extent to which we can ensure that our results are the same if our study would be conducted again).

a) Construct validity: The survey required respondents with a basic knowledge of self-adaptive systems and uncertainty necessary to interpret the questions properly. We mitigated this threat by selecting experienced participants at the main venues of the community and invited additional experts ensuring that the required basic knowledge was present. Additionally, respondents could clarify issues in the free text provided with the questions. Some questions may have been formulated such that respondents were forced to provide an answer that may not have objectively expressed their opinion. We mitigated this threat by allowing the respondents to provide comments with their answers. Another possible threat is a bias in formulating the questions. To mitigate this risk, we used a refinement process when defining the questions, where the four involved researchers reviewed the questions, individually and as a group.

b) External validity: Generalization of the study results might be a potential threat to validity. The main issue here is the selection of the sample of the population that may not have been representative. This may lead to study results that may be imprecise. To that end, we selected participants at the main venues of the community and invited additional experts, increasing the confidence that the sample was representative.

c) Reliability: Data analysis and coding in particular are creative tasks that are to some extent subjective. To mitigate bias, two researchers performed the data analysis of each question in an iterative way and then the results were cross-checked by the two other researchers. Any differences where discussed until we reached consensus. In addition, we made all the material of the survey publicly available.

VII. Conclusion

We reported the results of a survey aimed at shedding light on the perception of the research community on uncertainty in self-adaptive systems. The survey generated multiple insights. The majority of the participants consider that self-adaptive systems can be engineered to cope with unanticipated change. Uncertainty can be represented using parameters and structure of runtime models, and the modeling formalism. Proposed techniques to handle uncertainty include software evolution, online modeling mechanisms, and learning techniques. The survey results suggest the need for a research agenda centered on assurances, proactive adaptation, integration with machine learning, and scalability. Emerging challenges include self-explainability and ethical aspects. We hope that these findings will inspire valuable future research on self-adaptive systems.

References

[1] B. H. Cheng *et al.*, "Software engineering for self-adaptive systems: A research roadmap," in *Software engineering for self-adaptive systems*. Springer, 2009.

[2] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 2, 2009.

[3] D. Weyns, "Software engineering of self-adaptive systems," *Handbook of Software Engineering*, 2017.

[4] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[5] J. Zhang and B. H. Cheng, "Model-based development of dynamically adaptive software," in *28th International Conference on Software Engineering*, 2006, pp. 371–380.

[6] D. Garlan, S. Cheng, A. Huang, B. Schmerl, and P. Steenkiste, "Rainbow: Architecture-based self-adaptation with reusable infrastructure," *Computer*, vol. 37, no. 10, pp. 46–54, 2004.

[7] P. Oreizy, N. Medvidovic, and R. Taylor, "Architecture-based Runtime Software Evolution," in *International Conference on Software Engineering*. IEEE, 1998.

[8] D. Garlan, "Software engineering in an uncertain world," in *Future of Software Engineering Research*. ACM, 2010.

[9] L. Baresi and C. Ghezzi, "The disappearing boundary between development-time and run-time," in *Future of Software Engineering Research*, 2010.

[10] J. Andersson *et al.*, "Software engineering processes for self-adaptive systems," in *Software Engineering for Self-Adaptive Systems II*, 2010.

[11] A. Ramirez, A. Jensen, and B. H. Cheng, "A taxonomy of uncertainty for dynamically adaptive systems," in *Software Engineering for Adaptive and Self-Managing Systems*, 2012.

[12] N. Esfahani and S. Malek, "Uncertainty in self-adaptive software systems," in *Software Engineering for Self-Adaptive Systems II*. Springer, 2013.

[13] D. Perez-Palacin and R. Mirandola, "Uncertainties in the modeling of self-adaptive systems: a taxonomy and an example of availability evaluation," in *International Conference on Performance Engineering*, 2014.

[14] S. Mahdavi-Hezavehi, P. Avgeriou, and D. Weyns, "A classification framework of uncertainty in architecture-based self-adaptive systems with multiple quality requirements," in *Managing Trade-Offs in Adaptable Software Architectures*. Morgan Kaufmann, 2017, pp. 45 – 77.

[15] A. Keith and D. Ahner, "A survey of decision making and optimization under uncertainty," *Annals of Operations Research*, 2019.

[16] R. Calinescu *et al.*, "Synthesis and verification of self-aware computing systems," in *Self-Aware Computing Systems*. Springer, 2017.

[17] G. A. Moreno, J. Camara, D. Garlan, and B. Schmerl, "Proactive self-adaptation under uncertainty: A probabilistic model checking approach," in *Foundations of Software Engineering*. ACM, 2015.

[18] R. Calinescu, M. Ceska, S. Gerasimou, M. Kwiatkowska, and N. Paoletti, "Efficient synthesis of robust models for stochastic systems," *Journal of Systems and Software*, vol. 143, pp. 140–158, 2018.

[19] S. Gerasimou, R. Calinescu, and G. Tamburrelli, "Synthesis of probabilistic models for quality-of-service software engineering," *Automated Software Engineering*, vol. 25, no. 4, 2018.

[20] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. Wiley, 2012.

[21] B. Kitchenham and S. Pfleeger, "Personal opinion surveys," in *Guide to Advanced Empirical Software Engineering*. Springer, 2008.

[22] K.-J. Stol, P. Ralph, and B. Fitzgerald, "Grounded theory in software engineering research," in *International Conference on Software Engineering*. ACM, 2016.

[23] L. Prechelt, D. Graziotin, and M. Fernández, "A community's perspective on the status and future of peer review in software engineering," *Information and Software Technology*, 10 2017.

[24] L. Nahabedian, V. Braberman, D. N., J. Kramer, and S. Uchitel, "Dynamic reconfiguration of business processes," in *Business Process Management*. Springer, 2019.

[25] C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in Software Engineering*. Springer, 2012.