History-Aware Explanations: Towards Enabling Human-In-The-Loop in Self-Adaptive Systems

Juan Parra-Ullauri
j.parra-ullauri@aston.ac.uk
Aston University
Birmingham, UK

Antonio García-Domínguez
a.garcia-dominguez@aston.ac.uk
Aston University
Birmingham, UK

Nelly Bencomo
nelly.bencomo@durham.ac.uk
Durham University
Durham, UK

Luis Garcia-Paucar
garciapl@aston.ac.uk
Aston University
Birmingham, UK

ABSTRACT

The complexity of real-world problems requires modern software systems to autonomously adapt and modify their behaviour at run time to deal with internal and external challenges and contexts. Consequently, these self-adaptive systems (SAS) can show unexpected and surprising behaviours to users, who may not understand or agree with them. This is exacerbated due to the ubiquity and complexity of AI-based systems which are often considered as "black-boxes". Users may feel that the decision-making process of SAS is oblivious to the user's own decision-making criteria and priorities. Inevitably, users may mistrust or even avoid using the system. Furthermore, SAS could benefit from the human involvement in satisfying stakeholders' requirements. Accordingly, it is argued that a system should be able to explain its behaviour and how it has reached its current state. A *history-aware, human-in-the-loop* approach to address these issues is presented in this paper. For this approach, the system should i) offer access and retrieval of historic data about the past behaviour of the system, ii) track over time the reasons for its decisions to show and explain them to the users, and iii) provide capabilities, called *effectors*, to empower users by allowing them to steer the decision-making based on the information provided by i) and ii). This paper looks into enabling a *human-in-the-loop* approach into the decision-making of SAS based on the MAPE-K architecture. We present a feedback layer based on temporal graph databases (TGDB) that has been added to the MAPE-K architecture to provide a two-way communication between the human and the SAS. Collaboration, communication and trustworthiness between the human and SAS is promoted by the provision of history-based explanations extracted from the TGDB, and a set of effectors allow human users to influence the system based on the received information. The encouraging results of an application of the approach to a network management case study and a validation from a SAS expert are shown.

MODELS '22 Companion, October 23–28, 2022, Montreal, QC, Canada

ACM ISBN 978-1-4503-9467-3/22/10.
https://doi.org/10.1145/3550356.3561538

CCS CONCEPTS

• **Software and its engineering → Software reliability**; • **Information systems → Information extraction**; • **Human-centered computing → HCI theory, concepts and models**.

ACM Reference Format:
Juan Parra-Ullauri, Antonio García-Domínguez, Nelly Bencomo, and Luis Garcia-Paucar. 2022. History-Aware Explanations: Towards Enabling Human-In-The-Loop in Self-Adaptive Systems. In *ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS '22 Companion), October 23–28, 2022, Montreal, QC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3550356.3561538

1 INTRODUCTION

Autonomous, self-organised and self-adaptive systems (SAS) need to monitor their environmental context using sensors to dynamically adapt to changes in the highly volatile and heterogeneous environments that characterise them [13, 37]. The inherent complexity of SAS makes them difficult to be understood by their users and stakeholders in general [31]. The issue is of significant importance, since users who cannot understand or feel part of the decision-making process may mistrust the system or simply stop using it [48]. The complexity is exacerbated with the use of artificial intelligence (AI) and machine learning (ML) [44]. Weyns et al. in [54] stated that the absence of human understanding might be one of the causes for the lack of widespread adoption of SAS. It is essential to improve the trust and understanding between the user and the system [33], to enhance collaboration, and to promote confidence [31]. Explaining the decision-making processes becomes increasingly important to enhance collaboration, and to increment confidence [33]. This is ratified by the General Data Protection Regulation (GDPR) law, which enshrines the right to explanation [11].

On the other hand, one popular architecture for building SAS is the MAPE-K loop proposed by IBM [30], which consists of four key stages; Monitoring, Analysing, Planning, and Executing around a Knowledge base. Traditionally, MAPE-K considers humans as external entities to the decision-making process. Integrating humans in this enclosed loop is an ongoing research challenge as these systems are in principle foreseen to be autonomous [13]. However, fully autonomous systems are sometimes infeasible due to the complexity of real-world problems [1], the presence of potentially dangerous

outcomes, or discrepancies between the system and user preferences [46]. Further, the decision making could benefit from expert human guidance [1]. Lately, researchers have started to consider the ethical reasons associated with leaving the human out of the MAPE loop in SAS [16].

It is discussed that the human role in the decision-making loop of a SAS can be seen as either passive (i.e. observing and understanding the decision-making process) or active (i.e. steering the decision-making process). Ideally, the decision-making should take into account the execution history [17]. Moreover, the understanding of the decision making by stakeholders should also include the system's reasoning history [25]. Therefore, a SAS should (i) offer access and retrieval to historic data about its behaviour, (ii) track over the time the reasons for its decisions so they can be used to explain and further inform end users and other stakeholders, and (iii) if an active role by the user is required, the SAS should also provide *effectors* to therefore empower human stakeholders to steer the decision-making while being informed by (i) and (ii). Examples of these stakeholders are developers and operators.

Previous work on history-aware explanations has focused on the analysis of decision-making over time with no active role played by the human. For example [25] offered forensic explanations. In contrast, [17] provide what can be called on-line explanations of the decision making for monitoring purposes. This paper addresses the third stage proposed in [40], using history-aware explanations to enable a human-in-the-loop approach where users take an *active* role in the decision-making of the SAS. The approach extends the MAPE-K loop to support both explanations and user interaction. A middleware layer based on temporal graph databases (TGDB), *the Explanatory and Feedback layer*, allows users to access and query the historic data, and contains effectors that allow users to steer the system's decision-making if they consider it necessary. The argument is that by engaging users and other stakeholders in the decision-making, their trust and understanding of the adaptive behaviour should improve [48]. The middleware architecture and components are presented and then applied to a network management case study. The validation was performed by a SAS expert who was able to extract explanations based on the system' historical behaviour and was able to steer the decision-making, if considered necessary, through the set of effectors available.

The rest of the document is structured as follows. Section 2 presents the foundations that underlie the research. Section 3 illustrates the approach to enable history-aware, human-in-the-loop self-adaptive systems. Section 4 describes the case study. Section 5 compares this work with other similar ones. Finally, Section 6 presents the conclusions and future work.

2 BACKGROUND

2.1 Self-awareness and History-awareness in Self-adaptation

Self-awareness capabilities allow the system to access knowledge related to its own state and the environment, supporting a better understanding and reasoning of its own adaptive behaviour [8, 12]. Self-awareness can also be related to different specific capabilities such as goal-awareness [50], requirements-awareness [48] or time-awareness [5]. Specifically, time-awareness refers to the use of knowledge of historical but also future phenomena [5] for systems' reasoning. It is argued that time-awareness requires node-level memory, and capabilities for time series modelling and/or anticipation [45]. History-awareness is implied in time-awareness.

Existing work on self-awareness tends to leave history implicit in the formal model [12]. As claimed in [17], explicit representation of history provides a more transparent consideration of the performance of past actions in the decision process. Making the history implicit in the model also means leaving the storage and retrieval of this past history as an *ad hoc* effort, which changes from SAS to SAS [17]. In some cases, past history is "compressed" so much that it is unrecoverable: the user cannot see what the system has based its decisions upon.

In regard to accessing explicit representations of history to support reasoning and understandability, the authors of [48] argue that a SAS needs to garner confidence in its users by explaining its behaviour during execution. Developers also need explanations to avoid "surprises" during testing and maintenance [53]. The full potential of techniques based on AI and evolutionary computing may not be realised without explanation capabilities [3].

2.2 Explanations in Self-adaptive Systems

Explanations provide a key capability to shape the understanding that humans develop when observing the environment, especially when their perceptions diverge from their expectations [14]. Explanation-aware computing has received growing interest due to the ubiquity and complexity of AI-based systems, creating the notion of explainable AI (XAI) [27]. There are different arguments in favor of XAI. Adadi et. al. stated four main ones in [2]:

- *Explain to justify:* AI is involved in more and more areas of our everyday lives. People affected by AI-influenced decisions (e.g. when refused a loan) may demand a justification for the particular outcome. This transparency can justify a system's reasoning to ensure fair and ethical decisions [51] are being made.
- *Explain to control:* Explanations can often be used to keep agent actions inside an envelope of good behaviour. The explanations allow to discover the origin of a problem or to clarify misunderstandings between the system and the user [4]. Indeed, explanations can contribute to prompt identification of errors in non-critical situations [2].
- *Explain to discover:* AI systems can process large amounts of data that otherwise would be difficult for humans to process. Explanations are helpful to extract insights about the knowledge acquired by this processing [2].
- *Explain to improve:* In order to improve an AI system, it is key to discover its flaws. A system that can be explained and understood can be easier to enhance and use to the best advantage [47].

In SAS, explainability can be described as the capability of answering questions about the system's past, present and future behaviours. The answers to these questions can explain why a decision was made or a particular state was reached [17]. Systems able to explain their decisions, concepts, and information sources to users can demonstrate their trustworthiness and support users' understanding [44]. Understanding what the system did requires

tracking its decision history and explaining those decisions (either textually or graphically [4]) to the users coherently, which is part of the present work. We focus on *explanations to control* and *explanations to improve* for developers and knowledgeable users. These two groups of users are familiar with developing and/or using SAS and are, hence, interested in understanding, diagnosing, as well as refining such systems in a given application context [34].

2.3 Storage and Retrieval of Historic Data

Identifying historical patterns in the data produced by a system has been a topic of interest for a long time. A 2012 survey on time-series mining by Esling et al. [20] outlines more than two decades of research work on this topic. Typical tasks include finding time-points of interest, clustering similar regions, classifying time-points, finding anomalies or predicting future time-points.

In regard to industrial applications, the need to organise the large volumes of data generated by the Web and the Internet of Things has motivated the development of better time-series analysis capabilities in database technologies. For instance, the Elasticsearch search engine can index large document collections with numerical measurements over time, and then apply machine learning approaches to find anomalies [18]. Moreover, on healthcare context, the temporal history of some hospitalized patient can be described by the time series of the values of his/her temperature, blood pressure, and oxygenation [49].

Still, these time-series have the limitation that they simply track the evolution of a metric: they cannot track, for instance, the evolution of the relationships within a system. They cannot directly represent the relationships between multiple evolving metrics, either. Graph databases such as Neo4j [42] do a good job at tracking complex networks of relationships between many entities (e.g. social graphs). However, on the other hand, they do not model explicitly the time dimension.

The above has motivated the development of *temporal graph databases* (TGDBs), which can track the evolution of a labelled attributed graph (where both nodes and edges can have collections of arbitrary *key-value* pairs). One implementation of this approach is Greycat [28] which uses a copy-on-write approach to efficiently represent a graph which evolves over a timeline of *instants*, by only storing the changed nodes at each instant. The timeline may also branch off into parallel *worlds*, where one world may reflect the actual measurements, and the rest may reflect *what-if* scenarios derived from simulations. In the present work, by using TGDBs, it is possible to track the evolution of certain metrics at each node, as well as the changes in the relationships of the various entities in the system, or their appearance and disappearance.

2.4 Human-in-the-loop: Implications

Human-in-the-loop feedback control systems offer exciting opportunities to a broad range of cyber-physical system applications including energy management, healthcare and automobile systems [36]. For example, explicitly introducing human-in-the-loop models for autonomous driving could improve safety [36]. Nunes et al. in [38] presented a taxonomy of human-in-the-loop applications (Fig. 1). The authors outline several cases for human-machine interaction: (i) systems manually controlled by users, (ii) systems

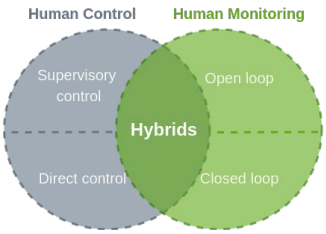

Figure 1: Taxonomy of human-in-the-loop applications [38]

that make decisions while monitored by humans, and (iii) hybrids of (i) and (ii). The present approach belongs to the hybrid category, i.e. the human is part of the system's context, but can also either supervise and/or influence it.

Although having humans in the loop has its advantages, modelling human behaviours can be challenging due to complex physiological, psychological and behavioural aspects of human beings [36]. There have been different initiatives to study human collaboration in the context of SAS [9, 26, 32]. One common approach to involve humans in self-adaptation is the Opportunity-Willingness-Capability (OWC) model [19]. This is a modelling approach used to quantify how much a participant who is performing a given task can affect systems. In this context, *Opportunity* describes the prerequisites to attempt a task (i.e Does the participant have access to the system?). *Willingness* captures the factors that could affect the intention of a human to attempt a task (i.e. motivation, mood). *Capability* describes the feasibility of a human to perform the task accurately (i.e. level of training for the task). To achieve an acceptable human-in-the-loop collaboration, the human must achieve states for which the OWC model can be evaluated as true [26]. The true state values selected for our OWC implementation are depicted in Section 4.

3 PROPOSAL: EXPLANATORY AND FEEDBACK LAYER

This paper targets the level 3 of the staged roadmap for history-awareness in SAS proposed in [40], considering the human as an external entity able to steer the system' decision-making actively. While monitoring a SAS, the user may not completely approve the system's current behaviour. The user may even disagree with the history-based explanations given by the running system. Allowing users to take part in the decision-making may increase their level of confidence in the system. However, achieving this may prove to be demanding as the explanations will need to be adequate and concise. Additionally, new agreements between the running systems and human should be allowed. For example, a SAS may allow or encourage users to change their preferences at runtime depending on unknown rewards that the SAS may uncover during runtime, and which were not foreseen before. Situations like those would create a loop between the system, its explanations, and the ensuing feedback from the user. Based on the above, this paper proposes to extend the original MAPE-K architecture by introducing an *explanatory feedback layer* to enable human-in-the-loop capabilities in a SAS. A two-way communication between the human and the system should be available, based on compliance with the OWC

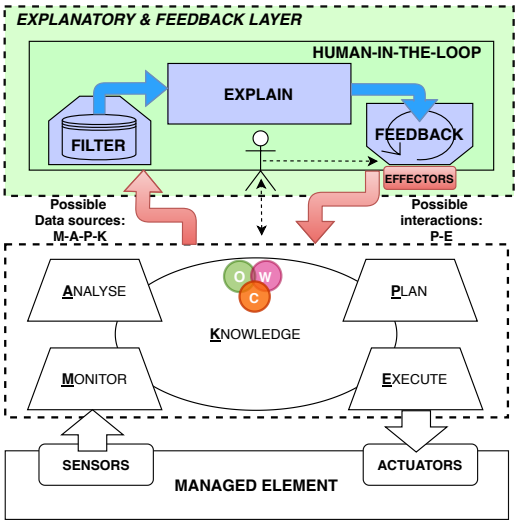

Figure 2: Explanatory and feedback layer extension to MAPE-K to enable Human-in-the-loop.

model described in Section 2.4. This layer introduces three new components to the MAPE-K architecture, as shown in Figure 2 and are described next.

3.1 Filter Component

This component receives and processes the data coming from the system. Data can come from the *Monitor*, *Analyse* or *Plan* stages of the MAPE-K loop. In order to build an explanation, *Monitor* can provide raw information about sensor readings and/or data that the system is collecting. *Analyse* can feed the filter with the information that is currently being used by the system for making decisions, and *Plan* can inform the user about the system's intentions for decision-making. Initially, it will be the developer who defines the focus of interest, i.e. the subset of the data that will feed the filter component and produce the explanations. However, the ultimate goal is that users will define the object of interest based on their own needs. This filter component is made-up by a temporal graph database (TGDB) built with information provided by the system in the form of a log. This log is reshaped to conform with the trace-metamodel presented in the authors' previous work in [39, 40]. The model represents relationships between the elements within a system. The trace-metamodel links the system' goals and decisions to its observations and reasoning. It has been updated from [40] to include the concept of *Agent* to keep track if decisions and observations are performed by the system or the human for accountability. The information is stored in the TGDB creating a new snapshot at the current point in time: all relevant versions are kept. We use a model indexer to automatically compare the trace-model as an object graph against the current version of the temporal graph. It creates a new version which only updates the temporal graph where needed, for efficient storage.

3.2 Explain Component

This component is where the explanations are constructed and presented. The final shape of a satisfying explanation partly depends on the understanding shown by the human recipient. Therefore, a rigid system for which developers or domain experts have defined explanations with no awareness of the needs and expectations of the recipients may be not convenient for users with different backgrounds. Allowing the user to have access to the system's behavioural history and request specific explanations would help to complete their mental model or test hypotheses over system behaviour. The explain component can run a query on the TGDB using the time-aware query language presented in [23], an extension of the Epsilon Object Language (EOL) to define temporal patterns that traverse the history of a model. The result of this query contains the information that will be used to construct the explanations. These explanations could be presented in textual or graphical ways, e.g. plots of various kinds, yes/no answers, or specific examples of matches of a certain temporal pattern. In relation to the explanation phases defined by Anjomshoae et al. [4], this work tackles the first two: i) the *explanation generation* is the construction of the causally connected TGDB (performed on the previous component), and ii) the *explanation communication* is the extraction of the information using the temporal query language (what information will be provided) and the presentation of explanations either textually or graphically (how will it be presented).

3.3 Feedback Component

The feedback component allows users to make changes in the system through a set of effectors if they consider it necessary. At the *Plan* stage of the MAPE-K architecture, a human could influence the high-level goals of the system by guiding the system and their priorities, or the internal parameters governing their algorithm: in both cases, the users would need an appropriately abstracted explanation to allow them to make an informed decision about the relevance and impact of the intended changes. At the *Execute* stage, effectors could allow users to explicitly select certain actions; for example, on an autonomous car the user could decide to go in a specific direction that goes against the system's reasoning. In this case the system may need to reconfigure its decision-making to meet a new preference introduced by a user. The full functionality of the effectors will be only available when the OWC model is evaluated as true which is introduced as part of the Knowledge in the MAPE-K loop.

4 CASE STUDY

4.1 RDM: Remote Data Mirroring

This section will evaluate the approach through a case study on the Remote Data Mirroring (RDM) SAS [7, 29], which uses Bayesian Learning an AI approach [24]. RDM manages data servers and network links, and it aims to protect from data loss by replicating data across servers. Its overall structure is shown in Figure 3. Uncertainty exists due to different unexpected situations such as delayed or lost messages, noise in sensors, or network link failures. RDM self-adapts to these situations by reconfiguring itself. Specifically, RDM can use two topologies: `Minimum Spanning Tree (MST)`,

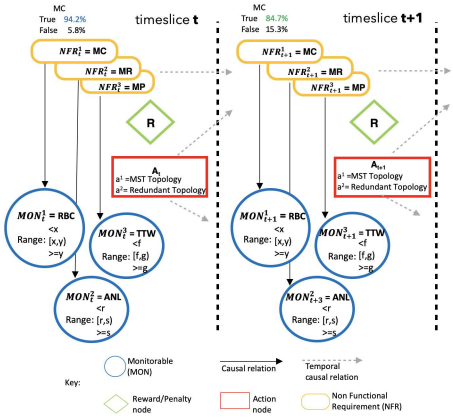

Figure 3: RDM SAS - case study

and `Redundant Topology (RT)`. Both topologies provide their own levels of reliability, performance and cost which are taken into account while estimating the observed levels of satisfaction of the non-functional requirements (NFRs): `Minimization of Cost (MC)`, `Maximization of Reliability (MR)`, and `Maximization of Performance (MP)`. `MST` is more efficient in terms of energy consumption and performance, whereas `RT` is more reliable.

The RDM SAS has been configured with Service Level Agreements (SLAs) for the satisfaction levels of the NFRs. The identified SLAs were: P(`MC`=True) ≥ **0.8** (the observed level of satisfaction of `MC` is greater than or equal to 0.8 out of 1), P(`MR`=True) ≥ **0.9**, and P(`MP`=True) ≥ **0.75** respectively. Initial stakeholders' preferences about the NFRs and adaptation topologies have also been provided. They are represented by the Reward/Penalty node shown in Fig. 3. In RDM, the initial preferences provided by the domain experts favour the MST topology under *stable conditions* [24]. Stable conditions represent a system context where the average (since the system started the execution) satisfaction levels of the NFRs meet their SLAs. In this work, we will showcase how a user can take a more active role in the decision-making based on history-aware explanations provided by the RDM SAS under unexpected contexts (i.e. *unstable conditions*) detected at runtime.

4.2 Enabling History-aware Human-in-the-loop in RDM

As mentioned in Section 2.2, the history-aware explanations in this case study targeted RDM SAS developers or operators as SAS users. The user interacts with the RDM under ideal conditions. Ideal refers to the fact that the SAS developer fulfils the requirements of the OWC model [19], for achieving effective human-machine interaction. Table 1 shows the selected values for each OWC element under the ideal conditions identified for this experiment. Accordingly, we can state that the SAS developer: (i) *has access* to the RDM SAS, (ii) is *willing to interact* with it and (iii) *has the capacity* to perform an action in an effective way.

- Opportunity:
  - $human.currentAction \in$ { system monitoring (SM), break (B), out of work (OW) }
  - $human.location \in$ { workstation (W), meeting room (MR), using personal computer (PC) }
  - $human.accessControl \in$ {available (A), not available (NA)}
- Willingness:
  - $human.confidence \in$ { high (H), medium (M), low (L)}
  - $human.attention \in$ { high (H), medium (M), low (L)}
  - $human.stressLevel \in$ { high (H), medium (M), low (L) }
  - $human.stamina \in$ { high (H), medium (M), low (L) }
- Capacity:
  - $human.experience \in$ { high (H), medium (M), low (L)}
  - $human.support \in$ { available (A), not available (NA) }

Dimension	Element	Value	True Value
Opportunity	$h.currentAction$	SM	T
	$h.location$	W	T
	$h.controlAccess$	A	T
Willingness	$h.confidence$	H	T
	$h.attention$	H	T
	$h.stressLevel$	L	T
	$h.stamina$	H	T
Capability	$h.experience$	H	T
	$h.support$	A	T

Table 1: OWC model for RDM (ideal)

Regarding to the system, RDM SAS is extended with human-in-the-loop capabilities and a graphical user interface (GUI) for the interaction. Specifically, the components of the *explanatory and feedback layer* presented in section 3 are integrated to the RDM SAS as follows:

Filter component: to store the history of execution, a TGDB is employed. The filter component records data of the execution of the system and allows for flexible querying when analyzing the evolution of the system. In this experiment, the information is filtered for meeting the RDM developer explanations' requirements about:

- initial stakeholder preferences about the NFRs and SLAs for each one.
- adaptation strategies selected by the SAS (based on the stakeholder preferences) and their impact on the satisfaction levels of the NFRs.
- situations detected at runtime, where initial stakeholder preferences may drive the SAS to unsuitable adaptation strategies and thus to a negative impact on the satisfaction levels of the NFRs.

Explain component: after the relevant information is available, the explanations are constructed, either textually or graphically. Both would allow the developer to understand the system's behaviour. In the RDM case study, for each time slice during execution, a GUI is updated with information from the TGDB maintained by an Eclipse Hawk model indexing server [22]. Specifically, relevant history-aware explanations related to the current satisfaction levels of NFRs, adaptation topologies and SLAs are presented (Fig. 5). The implemented temporal query can be found in Listing 1.

Feedback component: users consume the explanations and can improve their mental model about the system's current behaviour. If system behaviour does not agree with the mental model of the

```
var result : Sequence;

var nfrs = NFRBelief.latest.all;
var dec=Decision.latest.all.first;
var blfs=dec.nfrBeliefsPre;

var mecblfs=blfs.first.versions;
var mrblfs=blfs.second.versions;
var mpblfs=blfs.third.versions;

var aveMEC=mecblfs.collect(b|b.
    estimatedProbability).average();
var aveMR=mrblfs.collect(b|b.
    estimatedProbability).average();
var aveMP=mpblfs.collect(b|b.
    estimatedProbability).average();

for(nfr in nfrs){
    var currentNFR = nfr.latest;
    result.add(Sequence{
      currentNFR.eContainer.eContainer.
    timesliceID,
      currentNFR.nfr.name,
      currentNFR.satisfied,
      currentNFR.estimatedProbability,
      currentNFR.eContainer.actionTaken.name
    ,
      aveMEC,aveMR,aveMP
      });
}
return result;
operation Sequence average(){
  return self.sum()/self.size();
}
```

Listing 1: EOL query to find NFRs, NFR averages, SLAs and topologies in RDM based on the trace-metamodel structure for each instant of time.

developer, changes can be requested. Since the decision-making in the RDM is driven by the satisfaction levels of the NFRs, the effectors exposed by the system (the buttons "+" and "-" shown in Fig. 5) will allow the developer to manipulate the system's preferences (in this case, the relative priorities or weights of the NFRs).

The sequence diagram in Fig. 4 shows the communication between the various components for level 3 (human-guided history-aware decision-making with explanation capabilities) of the RDM case study. The user initializes the simulation and the GUI assuming that the model indexer Hawk is running, the trace metamodel is registered and the log file to be indexed is defined. At the end of each timesclice, the system will update the log file with the corresponding information. The GUI is fed by the temporal graph that is being build in Hawk (*filter component*). At any point in time, the user can run a query based on the current state of the temporal graph in order to obtain an explanation about how it got there (*explanation component*). If the user determines that the system is not fulfilling the user's preferences or that an external action could improve the system performance, reconfiguration can be made using the effectors/controls in the GUI (*feedback component*).

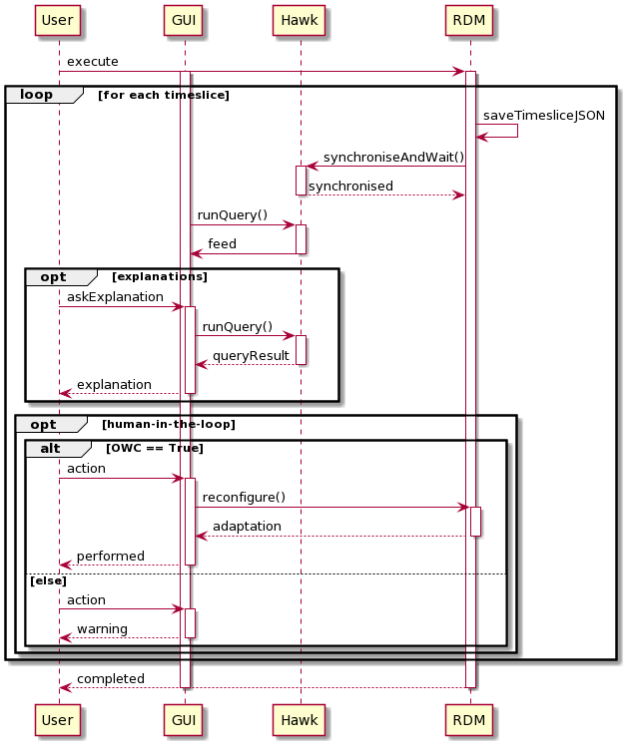

Figure 4: UML sequence diagram for interaction between components (RDM case study)

A reconfiguration in the system will be made if the OWC model is evaluated as true, otherwise the system should generate a warning message.

4.3 Evaluation and Discussion of Results

The experiments were conducted on a Lenovo Thinkpad T480 with an Intel i7-8550U CPU with 1.80GHz, running Ubuntu 18.04.2 LTS, Oracle Java version 1.8.0_201 and 15.6GB RAM. For the experiment, a simulation of the RDM SAS was run over 1000 time slices.

4.3.1 Experimental evaluation. The information provided by the filter component can be used to generate visual explanations such as those in Figure 5, which summarise the behaviour of the RDM SAS as it runs. It is observed that initially, the satisfaction levels of the NFRs `Minimization of Cost (MC)`, `Maximization of Reliability (MR)`, and `Maximization of Performance (MP)` are in general over their Service Level Agreements (SLAs), with some values below their thresholds, but this noise is considered to be normal for the system. Later, from time slice 324 (See Fig 5, Maximization of Reliability), a period of consecutive and unexpected data packet losses while using the `MST topology` reduces the observed reliability of the system. Data packet loss may represent link failures in a RDM, which can be caused by problems with the equipment (e.g. failures in a switch or router, or power failures [29]). Despite the new detected conditions and based on the initial RDM configuration, the selected topology continues to be MST (See Fig. 5, Maximization of Reliability: time slice 324). Specifically, under

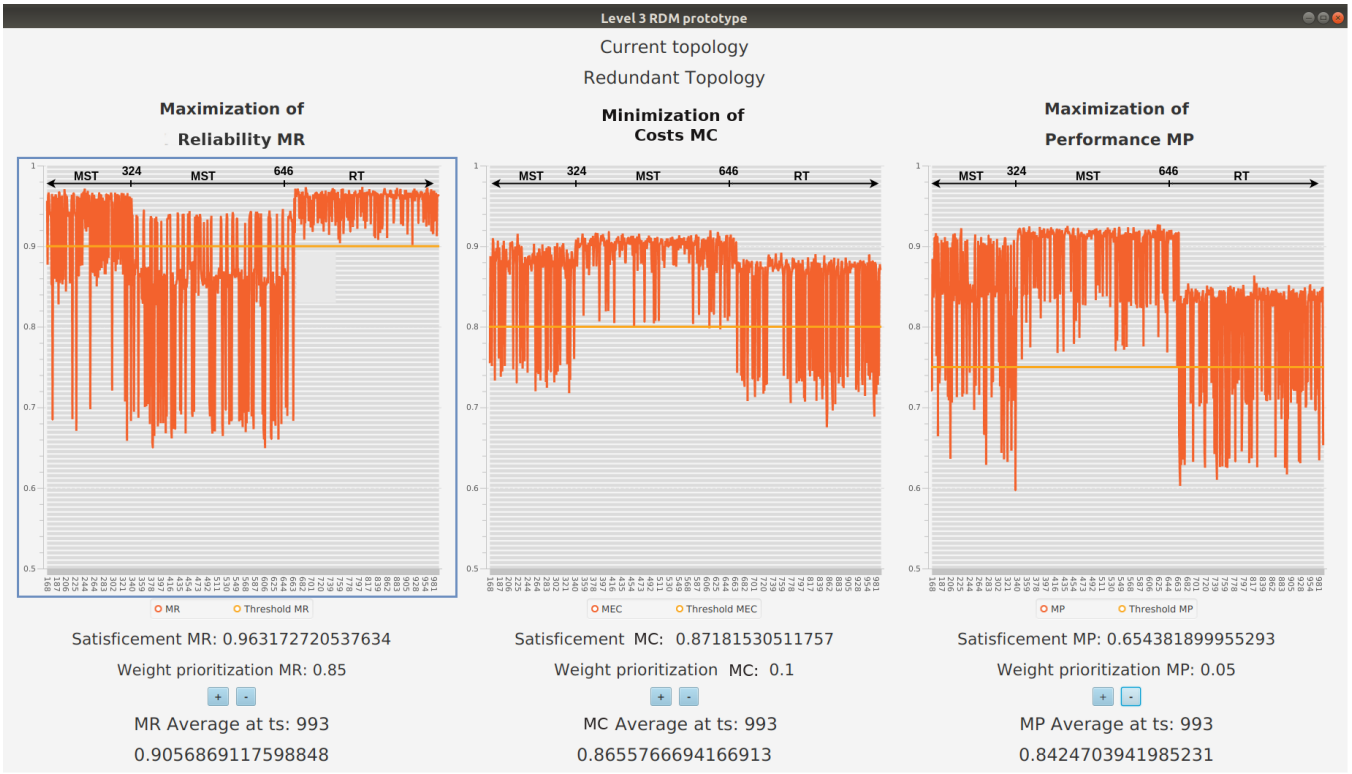

Figure 5: GUI showing the system's historical behavior. At time slice 646, the user set a higher priority to the MR NFR (left chart).

the current context, initial stakeholder preferences are not suitable anymore as they continue favouring the use of a topology that does not contribute to improve the satisfaction level of MR, which is mainly under its tolerance threshold (See Fig. 5, Maximization of Reliability: time slices 324 - 646). The preferences should be eventually reassessed and updated to assign higher importance to NFRs with poor satisfaction levels (e.g. MR, the reliability of the system) and to improve the selection of the topology in the RDM SAS.

Complementing the scenario presented above, through the integration of the human-in-the-loop based on the RDM components of the *explanatory feedback layer* the developer is able to explore the history and steer the decision-making. History-aware explanations are presented to the SAS developer through the GUI. Under the current runtime context, special attention is paid to: (i) the NFRs satisfaction levels from time slice 324 onwards, (ii) the current preferred topology (MST), and (iii) the current preferences about the NFRs. These explanations help the developer refine their "hypotheses" or mental models about the current state of the system.

Next, based on the information provided by the *explanation component*, the user is allowed to potentially improve the current behaviour of the system. If the user considers that the system is not fulfilling the intended behaviour or that an external action could improve its performance, a reconfiguration can be made using the effectors/controls in the GUI (*feedback component*). In order to reconfirm the external action selected by the user, Fig. 6 shows a *pre-adaptation explanation* of how relevant the effector "increase the priority of MR" can be. After the change is applied, Fig. 5 shows how the satisfaction level of MR increases from time slice 646 onward as a result. The satisfaction levels of MC and MP went down, but they still met their SLAs.

4.3.2 ***Discussion.*** The experiment has covered how some unforeseen dynamic contexts may affect negatively the NFR satisfaction levels when initial assumptions, e.g. stakeholder preferences, are not updated in response. The "NFRs *without* update of preferences" series in Fig. 7 shows this behaviour in the satisfaction levels of the NFRs from time slice 324 to time slice 646. The average satisfaction level of MR is always below the SLA given in the initial stakeholder preferences, despite the new detected context, and the RDM SAS continues favouring the MST topology as shown in Fig. 5. In contrast, by including the human-in-the-loop in the decision-making of the RDM SAS, it is possible to improve the general performance of the system and the NFR trade-offs. Going back to Fig. 7, "NFRs *with* update of preferences" shows this new behaviour from time slice 646 onward. It can be seen that after the user intervened the average satisfaction level of MR started meeting its SLA. There is also a slight reduction on the satisfaction levels of MC and MP, but they still meet their SLAs.

Through this experiment, we have shown how external entities (e.g. human stakeholders) are able to evaluate and update the parameters of a SAS based on live explanations of the SAS behaviour, participating in the tradeoffs between the NFRs in a SAS. Currently we have focused on explanations based on the evolution of a metric (eg. NFR average). However, explanations based on relationships between metrics and events spanned over the time can be obtained by exploiting the full potential of the causally connected TGDB

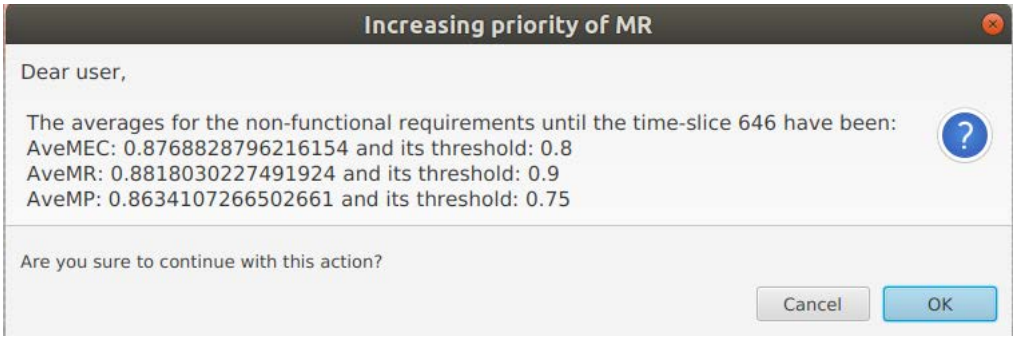

Figure 6: Textual pre-adaptation explanation from the system at time slice 646, when user shows interest in increasing priority of MR.

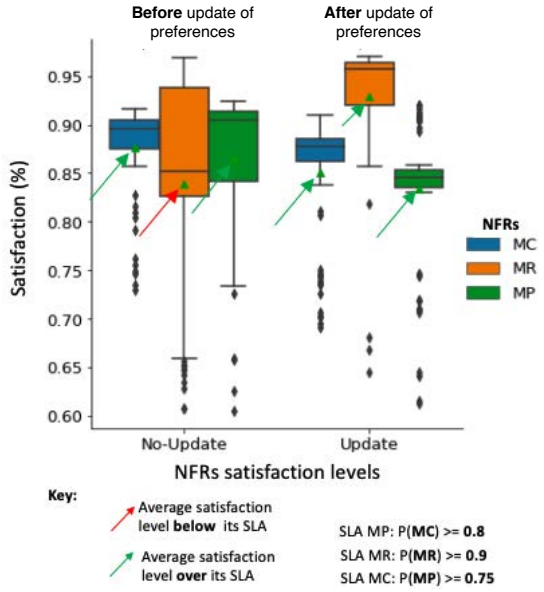

Figure 7: NFRs average satisfaction levels before and after human interaction

as shown in [40]. For example, the causes of the packet loss in the case study can be tracked. Future experiments will focus on these types of explanations. Additionally, further studies on the impact of human interventions conflicting with the perspectives or requirements of the SAS are required.

Regarding to the explanations stages mentioned in section 3, the present work has tackled the first two, explanation construction and communication. The third stage, explanation reception, has been assumed as ideal for the SAS developer based on the compliance of the OWC model. Under these assumptions, an explanation is deemed to be useful if the recipient understands the system's behaviour and feels confident to interact with it either passively or actively. However, the evaluation of the explanations reception and the impact of human interaction in SAS performance while considering different levels of user expertise outside of ideal OWC conditions, is still required. Furthermore, the study of explanations for non-expert stakeholders (e.g., novice users) is necessary.

5 RELATED WORK

Related work is categorised as human-in-the-loop for decision making systems, and explanations for autonomous systems.

5.1 Human-in-the-loop in Autonomous Systems

In [41], the authors propose a cooperative human-machine approach for continuous planning in self-adaptive systems. The work focuses on how the information from the real world (i.e. with unstructured data) is inserted into the system, and how it is subsequently integrated into planning. In [1], an approach for making autonomous systems benefit from human expertise is presented, using human guidance during the learning process of a Reinforcement Learning agent. The authors of [10] also study the pros and cons of involving humans in the adaptation loop. They propose a framework to reason about humans in a self-adaptation loop as system-level effectors during the execution adaptation stage. These approaches focus on how humans affect system decision-making and learning. Different from this paper, they do not consider human preferences or trust as in our case.

The work in [21], as in this paper, focuses on user preferences, proposing the MUSIC middleware for user participation. The aim is providing user control and trust while still maintaining adaptivity. However, different from the present work, it does not target the user's understanding of the behaviour of the system, and it does not take into account how the information is being perceived by the users, as it is done through the history-aware explanations in the present paper.

5.2 Explanations for Autonomous and Self-adaptive Systems

There are also emerging research initiatives related to explanations in self-adaptive systems to tackle the challenges posed by the use of AI and ML. In [6], the authors propose an architecture for building self-explainable systems. As in the case of the present paper, their MAB-Ex loop is a framework that can extend the SAS MAPE-K loop in order to support explanations.

Mouline et al. [35] also presented a temporal model to support interactive diagnosis of self-adaptive systems. The temporal model represents, stores and queries decisions, considering their context, requirements, and adaptation actions. However, these works lack

a feedback component to allow users to interact with the system based on the provided explanations.

Li et al. [32] propose explanations for human-in-the-loop as in this paper. Their target is to define when an explanation should be provided as a tactic of the SAS to support human interaction. However, their work does not focus on how explanations are built, which the present work supports through trace metamodels, temporal graphs, and a temporal query language. Further, they do not explicitly take into account the historical behaviour of the self-adaptive system, which is an advantage provided by the use of temporal graphs.

6 CONCLUDING REMARKS AND FUTURE WORK

This paper has proposed introducing human-in-the-loop capabilities into the MAPE-K architecture by adding an *explanatory and feedback layer* based on historical data. The proposed extension corresponds with the hybrid category of the taxonomy from Nunes [38], where the human is part of the system' context and can also interact with it. The definition, implementation and evaluation of the proposed extension are shown through a case study on an existing SAS that uses Bayesian Learning. The approach shown is human-centric, targeting trustworthiness with a two-way collaboration between the human and the SAS. The SAS provides the user with effectors to steer the decision-making: the effectors abstract away the details of the underlying decision algorithm from the user. These effectors for the interaction are available if the OWC model is evaluated as true. Through the feedback and explanatory layer, users can query the history of the system to improve their understanding about the behaviour exposed by the SAS. The impact of the user's decisions was examined using data collected from the same interface, with the ability to compare system performance pre- and post-adaptation. These human interactions are annotated in the history and tracked in the temporal graph for accountability.

The evaluation was performed by members of our team who are the developers of the RDM SAS and who focus on improving it. The type of explanations presented, either textual or graphical, fit the audience, who are able to understand the data representations and can extract knowledge from their system. However, this evaluation can represent a source of threats to validity. Therefore, additional evaluations are part of the immediate future work. We plan to test the approach to evaluate the user perceptions using some well-known techniques for software acceptance as the Technology Acceptance Model (TAM) [15] and its variants [52]. TAM will be used to evaluate the ease-of-use perception, usefulness perception and intention to use in the future of the proposed framework. Additionally, specific techniques to evaluate XAI as the one recently proposed by Rosenfeld, A., in [43] will be also explored. This work presents a methodology for evaluating XAI that focuses on metrics that quantify the appropriateness of the explanations provided given a specific goal. Additionally, as defined in the OWC model, the performance of humans in the decision making can vary according to various factors (e.g. training, workload, or stress levels): further investigation is needed on this topic as well.

There are other lines of work beyond these major ones. From the technical point of view, the explanations provided so far were mostly factual. However, explanations can be about proving or disproving hypotheses from the user, or presenting simplified predictors of the system behaviour. The development of these new types of explanations is another interesting line of further research. On the other hand, the current implementation is designed to be used in a central control node for a SAS. It would require further efforts to be applied to the case of a distributed SAS. Finally, the approach should be validated by using additional case studies, to measure how it improves explainability in a wider domain.

ACKNOWLEDGMENTS

This work has been partially sponsored by the EPSRC Research Project Twenty20Insight (Grant No. EP/T017627/1).

REFERENCES

[1] David Abel, John Salvatier, Andreas Stuhlmüller, and Owain Evans. 2017. Agent-agnostic human-in-the-loop reinforcement learning. *arXiv preprint arXiv:1701.04079* (2017).
[2] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018), 52138–52160.
[3] Robert Andrews, Joachim Diederich, and Alan Tickle. 1995. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems* (1995). https://doi.org/10.1016/0950-7051(96)81920-4
[4] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. 2019. Explainable agents and robots: Results from a systematic literature review. In *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*. International Foundation for Autonomous Agents and Multiagent Systems, 1078–1088.
[5] T. Becker, A. Agne, P. R. Lewis, et al. 2012. EPiCS: Engineering Proprioception in Computing Systems. In *IEEE 15th International Conference on Computational Science and Engineering*. https://doi.org/10.1109/ICCSE.2012.56
[6] Mathias Blumreiter, Joel Greenyer, Francisco Javier Chiyah Garcia, Verena Klös, Maike Schwammberger, Christoph Sommer, Andreas Vogelsang, and Andreas Wortmann. 2019. Towards self-explainable cyber-physical systems. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 543–548.
[7] Kate M. Bowers, Erik M. Fredericks, and Betty H. C. Cheng. 2018. Automated Optimization of Weighted Non-functional Objectives in Self-adaptive Systems. In *Search-Based Software Engineering*, Thelma Elita Colanzi and Phil McMinn (Eds.). Springer International Publishing.
[8] Javier Cámara, Kirstie L Bellman, Jeffrey O Kephart, Marco Autili, Nelly Bencomo, Ada Diaconescu, Holger Giese, Sebastian Götz, Paola Inverardi, Samuel Kounev, et al. 2017. Self-aware computing systems: Related concepts and research areas. In *Self-Aware Computing Systems*. Springer, 17–49.
[9] Javier Cámara, David Garlan, Gabriel A Moreno, and Bradley Schmerl. 2017. Evaluating trade-offs of human involvement in self-adaptive systems. In *Managing Trade-Offs in Adaptable Software Architectures*. Elsevier, 155–180.
[10] Javier Cámara, Gabriel Moreno, and David Garlan. 2015. Reasoning about human participation in self-adaptive systems. In *Proceedings of SEAMS*.
[11] Peter Carey. 2018. *Data protection: a practical guide to UK and EU law*. Oxford University Press, Inc.
[12] Tao Chen, Rami Bahsoon, and Xin Yao. 2018. A Survey and Taxonomy of Self-Aware and Self-Adaptive Cloud Autoscaling Systems. *ACM Comput. Surv.*, Article 61 (2018). https://doi.org/10.1145/3190507
[13] Betty HC Cheng, RogÚrio de Lemos, Holger Giese, et al. 2009. Software engineering for self-adaptive systems: A research roadmap. In *Software engineering for self-adaptive systems*. Springer.
[14] Michael T Cox. 2011. Metareasoning, monitoring, and self-explanation. *Metareasoning: Thinking about thinking* (2011).
[15] Fred D Davis. 1985. *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. Ph. D. Dissertation. Massachusetts Institute of Technology.
[16] Rogerio De Lemos. 2020. Human in the Loop: What is the Point of no Return?. In *Proceedings of SEAMS*.
[17] Antonio Garcia Dominguez, Nelly Bencomo, Juan Marcelo Parra Ullauri, and Luis Hernan Garcia Paucar. 2019. Towards history-aware self-adaptation with explanation capabilities. In *Proceedings of FAS* W 2019*. IEEE.
[18] Elastic. 2017. Introducting Machine Learning for the Elastic Stack. last checked: 2020-05-15.
[19] Douglas Eskins and William H Sanders. 2011. The multiple-asymmetric-utility system model: A framework for modeling cyber-human systems. In *2011 Eighth*

International Conference on Quantitative Evaluation of SysTems. IEEE, 233–242.
[20] Philippe Esling and Carlos Agon. 2012. Time-series data mining. *Comput. Surveys* (2012). https://doi.org/10.1145/2379776.2379788
[21] Christoph Evers, Romy Kniewel, Kurt Geihs, and Ludger Schmidt. 2014. The user in the loop: Enabling user participation for self-adaptive applications. *Future Generation Computer Systems* 34 (2014).
[22] Antonio Garcia-Dominguez, Konstantinos Barmpis, Dimitrios S. Kolovos, Ran Wei, and Richard F. Paige. 2017. Stress-testing remote model querying APIs for relational and graph-based stores. *Software & Systems Modeling* (2017). https://doi.org/10.1007/s10270-017-0606-9
[23] Antonio García-Domínguez, Nelly Bencomo, Juan Marcelo Parra-Ullauri, and Luis Garcia. 2019. Querying and annotating model histories with time-aware patterns. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*. IEEE.
[24] Luis Garcia-Paucar and Nelly Bencomo. 2019. Knowledge Base K Models to Support Trade-offs for Self-adaptation using Markov Processes. *13th IEEE Conference SASO, Sweden* (2019).
[25] Antonio García-Domínguez, Nelly Bencomo, and Luis Garcia. 2018. Reflecting on the past and the present with temporal graph-based models. In *Proceedings of MODELS 2018 Workshops*, Vol. 2245.
[26] Miriam Gil, Manoli Albert, Joan Fons, and Vicente Pelechano. 2019. Designing human-in-the-loop autonomous Cyber-Physical Systems. *International Journal of Human-Computer Studies* 130 (2019), 21 – 39. https://doi.org/10.1016/j.ijhcs.2019.04.006
[27] David Gunning. 2017. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web* 2 (2017), 2.
[28] Thomas Hartmann, Francois Fouquet, Matthieu Jimenez, Romain Rouvoy, and Yves Le Traon. 2017. Analyzing Complex Data in Motion at Scale with Temporal Graphs. In *Proceedings of SEKE'17*. https://doi.org/10.18293/SEKE2017-048
[29] Minwen Ji, Alistair C Veitch, John Wilkes, et al. 2003. Seneca: remote mirroring done write.. In *USENIX Annual Technical Conference, General Track*.
[30] J. O. Kephart and D. M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003). https://doi.org/10.1109/MC.2003.1160055
[31] Pierre Le Bras, David A Robb, Thomas S Methven, Stefano Padilla, and Mike J Chantler. 2018. Improving User Confidence in Concept Maps: Exploring Data Driven Explanations. In *Proceedings of CHI 2018*. ACM.
[32] Nianyu Li, Javier Cámara, David Garlan, and Bradley Schmerl. 2020. Reasoning about When to Provide Explanation for Human-involved Self-Adaptive Systems. In *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. IEEE, 195–204.
[33] Brian Y Lim, Anind K Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of CHI 2009*. ACM.
[34] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1, 1 (2017), 48–56.
[35] Ludovic Mouline, Amine Benelallam, François Fouquet, et al. 2018. A Temporal Model for Interactive Diagnosis of Adaptive Systems. In *ICAC 2018*. https://doi.org/10.1109/ICAC.2018.00029
[36] Sirajum Munir, John A Stankovic, Chieh-Jan Mike Liang, and Shan Lin. 2013. Cyber physical system challenges for human-in-the-loop control. In *Presented as part of the 8th International Workshop on Feedback Computing*.
[37] Richard Murch. 2004. *Autonomic computing*. IBM Press.
[38] David Sousa Nunes, Pei Zhang, and Jorge Sá Silva. 2015. A survey on human-in-the-loop applications towards an internet of all. *IEEE Communications Surveys & Tutorials* (2015).
[39] Juan Marcelo Parra-Ullauri, Antonio García-Domínguez, Nelly Bencomo, Changgang Zheng, Chen Zhen, Juan Boubeta-Puig, Guadalupe Ortiz, and Shufan Yang. 2022. Event-driven temporal models for explanations-ETeMoX: explaining reinforcement learning. *Software and Systems Modeling* 21, 3 (2022), 1091–1113.
[40] Juan Marcelo Parra-Ullauri, Antonio García-Domínguez, Luis Hernán García-Paucar, and Nelly Bencomo. 2020. Temporal Models for History-Aware Explainability. In *Proceedings of the 12th System Analysis and Modelling Conference*. 155–164.
[41] Colin Paterson, Radu Calinescu, Suresh Manandhar, and Di Wang. 2019. Using unstructured data to improve the continuous planning of critical processes involving humans. In *14th SEAMS*.
[42] Ian Robinson, James Webber, and Emil Eifrem. 2015. *Graph databases* (second ed.). O'Reilly. ISBN 978-1-4919-3089-2.
[43] Avi Rosenfeld. 2021. Better Metrics for Evaluating Explainable Artificial Intelligence. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 45–50.
[44] Thomas Roth-Berghofer, Stefan Schulz, David B Leake, and Daniel Bahls. 2007. Explanation-aware computing. *AI Magazine* (2007).
[45] Lucas Sakizloglou, Sona Ghahremani, Matthias Barkowsky, and Holger Giese. 2020. A Scalable Querying Scheme for Memory-efficient Runtime Models with History. In *2020 ACM/IEEE 23nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*.
[46] Mazeiar Salehie and Ladan Tahvildari. 2009. Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems* 4, 2 (2009), 14.
[47] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296* (2017).
[48] 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 *Proceedings of RE'10*. https://doi.org/10.1109/RE.2010.21
[49] Guido Sciavicco and Ionel Eduard Stan. 2020. Knowledge Extraction with Interval Temporal Logic Decision Trees. In *27th International Symposium on Temporal Representation and Reasoning (TIME 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
[50] Gabriel Tamura, Norha M. Villegas, Hausi A. Müller, Laurence Duchien, and Lionel Seinturier. 2013. Improving context-awareness in self-adaptation using the DYNAMICO reference model. In *Proceedings of SEAMS*. https://doi.org/10.1109/SEAMS.2013.6595502
[51] Matteo Turilli and Luciano Floridi. 2009. The ethics of information transparency. *Ethics and Information Technology* 11, 2 (2009), 105–112.
[52] Viswanath Venkatesh and Fred D Davis. 2000. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science* 46, 2 (2000), 186–204.
[53] Kristopher Welsh, Nelly Bencomo, Pete Sawyer, and Jon Whittle. 2014. Self-Explanation in Adaptive Systems Based on Runtime Goal-Based Models. *Trans. Computational Collective Intelligence* 16 (2014). https://doi.org/10.1007/978-3-662-44871-7_5
[54] Danny Weyns, M Usman Iftikhar, Sam Malek, and Jesper Andersson. 2012. Claims and supporting evidence for self-adaptive systems: A literature study. In *Proceedings of SEAMS*.