An Architectural Viewpoint for Benefit-Cost-Risk-Aware Decision-Making in Self-Adaptive Systems

DANNY WEYNS, Linnaeus University, Växjö, Sweden, Katholieke Universiteit Leuven, Leuven, Belgium, and NORCE, Bergen, Norway
SARA HEZAVEHI, University of Groningen, Groningen, The Netherlands and Linnaeus University, Sweden
PARIS AVGERIOU, University of Groningen, Groningen, The Netherlands
RADU CALINESCU, University of York, York, United Kingdom of Great Britain and Northern Ireland
RAFFAELA MIRANDOLA, KASTEL, Karlsruhe Institute of Technology, Germany
DIEGO PEREZ-PALACIN, Computer Science Department, Linnaeus University, Växjö, Sweden and Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Zaragoza, Spain

Self-adaptation equips a software system with a feedback loop that resolves uncertainties during operation and adapts the system to deal with them when necessary. Most self-adaptation approaches today use decision-making mechanisms that select for execution the adaptation option with the best-estimated benefit expressed as a set of adaptation goals. A few approaches also consider the estimated (one-off) cost of executing the candidate adaptation options. We argue that besides benefit and cost, decision-making in self-adaptive systems should also consider the estimated risk the system or its users would be exposed to if an adaptation option were selected for execution. Balancing all three concerns when evaluating the options for adaptation to mitigate uncertainty is essential for satisfying stakeholders' concerns and ensuring the safety and public acceptance of self-adaptive systems. In this article, we present a reference model for decision-making in self-adaptation that considers the estimated benefit, cost, and risk as core concerns of each adaptation option. Leveraging this model, we then present an ISO/IEC/IEEE 42010 compatible architectural viewpoint that aims at supporting software architects responsible for designing robust decision-making mechanisms for self-adaptive systems. We demonstrate the applicability, usefulness, and understandability of the viewpoint through a case study where participants with experience in the engineering of self-adaptive systems performed a set of design tasks in DeltaIoT, an Internet-of-Things exemplar for research on self-adaptive systems.

CCS Concepts: • **Software and its engineering** → **Software design engineering**; **Software design tradeoffs**; **Risk management**;

This work was partially funded by the EPSRC projects EP/V026747/1 "UKRI Trustworthy Autonomous Systems Node in Resilience" and EP/X040518/1 Communications Hub for Empowering Distributed Cloud Computing Applications and Research. The work was also partially supported by the pilot program Core Informatics at KIT (KiKIT) of the Helmholtz Association (HGF) and a ICM-Mid-Term-Fellowship at KIT.

Authors' Contact Information: Danny Weyns (corresponding author), Linnaeus University, Växjö, Sweden, Katholieke Universiteit Leuven, Leuven, Belgium, and NORCE, Bergen, Norway; e-mail: danny.weyns@lnu.se; Sara M. Hezavehi, University of Groningen, Groningen, The Netherlands, and Linnaeus University, Sweden; e-mail: s.mahdavihezavehi@uva.nl; Paris Avgeriou, University of Groningen, Groningen, The Netherlands; e-mail: p.avgeriou@rug.nl; Radu Calinescu, University of York, York, United Kingdom of Great Britain and Northern Ireland; e-mail: radu.calinescu@york.ac.uk; Raffaela Mirandola, KASTEL, Karlsruhe Institute of Technology, Germany; e-mail: raffaela.mirandola@polimi.it; Diego Perez-Palacin, Computer Science Department, Linnaeus University, Växjö, Sweden and Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Zaragoza, Spain; e-mail: diego.perez@lnu.se.

ACM 1556-4703/2025/3-ART2
https://doi.org/10.1145/3705612

Additional Key Words and Phrases: self-adaptive, risk, cost, benefit, viewpoint, decision-making

ACM Reference format:
Danny Weyns, Sara Hezavehi, Paris Avgeriou, Radu Calinescu, Raffaela Mirandola, and Diego Perez-Palacin. 2025. An Architectural Viewpoint for Benefit-Cost-Risk-Aware Decision-Making in Self-Adaptive Systems. *ACM Trans. Autonom. Adapt. Syst.* 20, 1, Article 2 (March 2025), 65 pages.
https://doi.org/10.1145/3705612

1 Introduction

Modern software systems are expected to deal with changing operating conditions, such as dynamic workloads, fluctuation in the availability of services, and changes in requirements. For system designers, these changes create uncertainties that may be difficult to anticipate before deployment [28, 36, 53, 58, 60]. Yet, without proper mitigation strategies, such uncertainties may jeopardize the goals of the system stakeholders at runtime. One approach to mitigate uncertainties is self-adaptation [21, 70]. Self-adaptation extends a software system with a feedback loop that monitors the system and its environment to identify and resolve the uncertainties by adapting the system to deal with changing conditions, or to gracefully degrade if necessary. Over the past two decades, researchers have extensively studied the problem of how to realize self-adaptation using feedback loops [70] and engineers have widely applied self-adaptation in industry [73].

A common self-adaptation approach is to equip the feedback loop with a decision-making mechanism that evaluates the relevant options for adaptation at runtime (i.e., the alternative system configurations considered for adaptation) and selects the option with the best expected outcome in terms of achieving a set of adaptation goals (e.g., performance or reliability). We refer to this as the *estimated benefit* that can be achieved by self-adaptation when a particular adaptation option is selected for execution. A few approaches also take into account the *estimated cost* for adapting the system when evaluating the options for adaptation [44, 68]. In this context, we use "cost" to refer to the one-off cost to perform an adaptation of the running system.[1] An example of a cost is the energy that is required to communicate the actions for adapting a self-adaptive system in a wireless network.

Besides the estimated benefit and cost, we argue that adaptation decisions should also take into account the *estimated risk* the system or its users would be exposed to if an adaptation option were selected and used to adapt the running system. With estimated risk we refer to the potential effects of uncertainties that may lead to positive or negative consequences on the system objectives as the result of selecting an adaptation option (based on the definition of risk in [13]). Considering the risk is particularly important in domains where decision-making may affect the safety and/or privacy of users, have an impact on the environment, or raise ethical or legal concerns [30]. Compared to other areas where risk is taken into account in decision-making, see for instance [3, 5, 12, 39], the decision-making of self-adaptive systems is lagging behind. Yet, in contrast to domains where risk analysis is done by humans supported by tools before the system is deployed, risk analysis in self-adaptive systems needs to be done automatically by the system itself, within the time window available for the adaptation decisions. This calls for solid preparation of decision-making by the system during design, such that adaptation decisions that take risk into account can be

---

[1] Note that the one-off cost for adaptation contrasts with the monetary cost such as the financial cost for the owner to run the application, or the price that may need to be paid by users when using the service. If deemed relevant for the stakeholders, these aspects need to be considered when determining the expected benefit of adapting the system.

made efficiently by the system during operation. One established design solution facilitating such preparation for autonomous runtime decision-making is the use of architectural viewpoints [37].

To address the aforementioned challenges, we introduce a reference model for decision-making in self-adaptation that considers the estimated benefit, cost, and risk as core concerns of each adaptation option considered in self-adaptation. Leveraging this reference model, we then present an architectural viewpoint for benefit-cost-risk-aware decision-making in self-adaptive systems that aims at supporting software architects responsible for designing such robust decision-making mechanisms for self-adaptive systems. The viewpoint is structured using the template recommended by the ISO/IEC/IEEE 42010 standard [43]. The viewpoint establishes the conventions for defining and using architecture views to deal with the concerns of stakeholders for decision-making in self-adaptation by taking into account the estimated benefit, cost, and risk of adaptation options. The reference model and viewpoint are grounded in a study of the literature and supported by extensive experiences in engineering self-adaptive systems across a wide range of domains. We validate the viewpoint through a case study where participants with experience in self-adaptation performed a set of design tasks in DeltaIoT [40], an **Internet-of-Things (IoT)** exemplar for research on self-adaptive systems.

The remainder of this article is structured as follows. Section 2 provides background on architectural viewpoints and discusses related work. In Section 3, we give a short overview of the scientific approach we followed in this research. Sections 4 and 5 then introduce the reference model and the viewpoint, respectively. In Section 6, we validate the viewpoint focusing on its applicability, usefulness, and understandability. Section 7 reflects on the evaluation results,and discusses threats to validity. Finally, we draw conclusions and look at future work in Section 8.

2 Background and Related work

In this section, we introduce the basic concepts necessary to lay out the foundation for our work and discuss relevant related research.

2.1 Self-Adaptive System

In this article, we consider a system to be self-adaptive if it complies with two basic principles [71]. The *external principle* states that a self-adaptive system can achieve its goals in the face of changes and uncertainties without or with minimal human intervention. While a human operator manages uncertainties in a regular computing system, a self-adaptive system can deal with uncertainties autonomously, possibly supported by an operator, taking a set of adaptation goals as input. The *internal principle* states that a self-adaptive system comprises two distinct parts: the first part, called the *managed system*, interacts with the environment and is responsible for the domain concerns for which the system is built; the second part, called the *managing system*, consists of a feedback loop that interacts with the first part (and monitors its environment) and is responsible for the adaptation concerns, i.e., concerns about the domain concerns. For instance, a domain concern for an IoT system may be to collect data obtained by its nodes deployed in the environment, and an adaptation concern may be to collect this data with minimum energy consumption. A human operator may be involved in realizing the functions of the managing system (human-in-the-loop), or in observing the system in operation and only taking action when needed (human-on-the-loop).

2.2 Architecture-Based Adaptation

Our particular focus is on *architecture-based adaptation* [33, 49, 57, 77], which is an established approach to realize self-adaptive systems. Architecture-based adaptation has a dual focus [71]: on the use of software architecture as an abstraction to *define* a self-adaptive system at design time [49]; and on the use of architectural models to *reason* about change and make adaptation decisions at runtime [33]. We are primarily concerned with the second aspect.

Aligned with the second principle of self-adaptation, architecture-based adaptation comprises a managed system (e.g., an IoT system with battery-powered nodes that are deployed in the environment and communicate wireless) that deals with the domain concerns, i.e., the functions or services that need to be provided to the users (e.g., the network collects data about the presence of humans, the status of locks, the temperature), and a managing system that deals with the adaptation concerns, i.e., how the domain concerns are achieved in terms of benefits (e.g., allowed packet loss, minimum energy consumption), costs (e.g., the extra energy required to communicate adaptation actions to the nodes in order to change the network settings), and risks (e.g., the level of privacy protection of the data communicated over the IoT network). A reference approach to realize the managing system is by means of a so-called MAPE feedback loop [47, 78]. MAPE refers to the basic functions that need to be realized by the feedback loop: Monitor the system and its environment, Analyze the situation and the options for adaptation, Plan the adaptation of the managed system for the best adaptation option, and Execute the actions of the plan to adapt the managed system. The MAPE functions share a repository with Knowledge that typically comprises different types of runtime models [11, 77] (MAPE is therefore also referred to as MAPE-K). A concrete architecture maps the MAPE-K functions to components, which can be one-to-one or any other mapping (e.g., analysis and planning may be mapped to a decision-making component).

In 1998, Oreizy et al. [57] presented a pioneering model for architecture-based adaptation that comprises two simultaneous processes: system adaptation that deals with detecting and handling changes, and system evolution that deals with the consistent application of change over time. Around the same time, IBM launched its legendary initiative on autonomic computing [46] that took inspiration from the autonomic nervous system of the human body to enable computing systems to manage themselves based on high-level goals. Garlan et al. [33] pointed out the central role of architectural models as first-class elements that enable a system to reason about system-wide change, a principle that aligns with “models at runtime” introduced by Blair et al. [11]. Since these pioneering efforts, a substantial body of knowledge has been developed in this field [71]; some characteristic examples include [16, 41, 81]. A recent large-scale survey [73] provided evidence that the principles of architecture-based adaptation are also widely applied in industry.

2.3 Uncertainty in Self-Adaptive Systems

Already in 2010, Garlan [32] pointed to the key role of uncertainty in software-intensive systems. Different researchers have provided taxonomies for uncertainty in self-adaptive systems, including Ramirez et al. [60], Perez-Palacin and Mirandola [58], Esfahani and Malek [28], and Musil et al. [56]. Leveraging these efforts, Mahdavi-Hezavehi et al. [53] provided a systematic overview of uncertainty dimensions (location, nature, level/spectrum, emerging time, sources) with their respective options. In this overview, 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.

Recently, Troya et al. [65] identified existing notations and formalisms used to represent the different types of uncertainty, together with the software development phase in which they are used and the types of analysis allowed. Hezavehi et al. [36] performed a survey in which over 100 members of the research community provided insights into how the concept of uncertainty is understood and currently handled in the engineering of self-adaptive systems.

According to Weyns [71], the fifth of seven waves of research in engineering self-adaptive systems focuses on “Guarantees Under Uncertainties” emphasizing uncertainty as a core driver for self-adaptation. In consolidating the existing work on self-adaptive systems, Weyns [71] defines uncertainty in self-adaptive systems as “any deviation of deterministic knowledge that may reduce the confidence of adaptation decisions made based on the knowledge.” The research presented in our article aligns with this definition. To that end, we focus on mitigating uncertainty in the decision-making of self-adaptive systems by taking into account benefit, cost, and risk when an adaptation option is considered and ultimately selected for execution.

2.4 Decision-Making in Self-Adaptive Systems Based on Estimated Benefit

Most existing approaches for runtime decision-making in self-adaptive systems focus on the benefit that can be obtained when applying an adaptation to the managed system. Estimated benefit refers to the expected advantage (or implied effects) that will be obtained by adapting the system from its current configuration to a new configuration. Estimated benefit is usually expressed in terms of quality properties of the system in the form of adaptation goals [76].

We summarize several prototypical approaches for decision-making in self-adaptive systems that are based on the estimated benefit. Moreno et al. [54] proposed a method for improving decision-making in a self-adaptive system that is based on maximizing the accumulated utility over the look-ahead horizon. In the example case of a server-based system used for evaluation, the utility was defined as a weighted sum of the average time to serve a request and the cost to be paid for the adaptation options. Ramirez and Cheng [59] present a goal-based requirements model-driven approach for automatically deriving state-, metric-, and fuzzy logic-based utility functions for relaxed goal models. This approach was evaluated for an intelligent vehicle system. Cámara et al. [25] presented a formal approach based on a stochastic game that considers uncertainty in sensing when reasoning about the best way to adapt and improve system utility. Utility in a client-server case used for evaluation is defined based on user annoyance and the portion of malicious clients obtained when selecting a configuration for adaptation. Calinescu et al. [17] presented Engineering Trustworthy Self-Adaptive Systems, a framework that combines (1) design-time and runtime modeling and verification, with (2) industry-adopted assurance cases. The approach employs probabilistic verification to verify stochastic models of the adaptation options of the managed system and its environment to comply with a set of rules. In the case of an unmanned underwater vehicle used for evaluation, the rules included a minimum threshold for measurement accuracy and minimizing sensing energy. Weyns et al. [75] verified adaptation options using statistical model checking to ensure that the selected option complies with a set of adaptation goals that are defined as rules. In the example of an IoT case used for the evaluation, rules included thresholds on allowed packet loss and network latency, and minimization of the energy consumed for communication. Purandare et al. [62] applied adaptation with three strategies to ensure the safety of **Small Uncrewed Aerial Systems (sUAS)**: stabilize the sUAS so that they can complete their flight, send the sUAS into loiter mode, and land the sUAS in place. An algorithm evaluates large sets of data to determine the stability of the system and its current context and based on that applies one of the strategies if needed. For the vast body of work on decision-making in self-adaptation based on estimated benefit, we refer to the literature.

In summary, while each of these approaches provides valuable contributions to the research on self-adaptive systems, they all consider only the estimated benefit of adaptation options when selecting a new configuration for adapting the managed system.

2.5 Decision-Making in Self-Adaptive Systems Based on Estimated Benefit and Cost

A number of self-adaptation approaches also consider the estimated cost, in addition to the estimated benefit, when making adaptation decisions. The estimated cost refers to the expected (one-off) cost of executing the option selected for adaptation. Merely focusing on estimated benefit and disregarding the impact of the cost implied by adapting the system may adversely affect the expected benefit of adaptation, and hence the quality goals of the system [68]. Cost may refer to time or resources that are required to apply adaptation options. Resources may refer to CPU cycles, network bandwidth, disc, memory, and battery energy to provide services to the user. Yet, there is currently no clear view in the literature on different types or classes of estimated cost; cost is often domain-specific and varies from system to system. We highlight a number of representative approaches.

Chen et al. [20] point to costs for self-adaptation such as planning delay and extra resource/energy consumption. To that end, they address the problem of how to make a binary decision at each point in time: whether to adapt the self-adaptive system, considering dynamic and uncertain monetary cost-benefit of adapting the system or not. The approach leverages principles from technical department and online machine learning. Bertolli et al. [10] presented an approach that dynamically selects components according to an adaptation strategy to achieve a required level of quality of service. The approach includes both a performance model and a cost model that quantifies the overhead for reconfiguring a component (e.g., when switching between versions). The approach is evaluated on a flood management application. Cámara et al. [18] considered cost in terms of adaptation tactic latency, i.e., the interval between the time when a tactic's execution is triggered and the time when its effects are observed in the state of the system due to application of adaptation. Similarly, Cheng et al. [23] considered cost as the time difference between enacting an adaptation tactic and the time when its effects can be observed. Van Der Donckt et al. [66, 67] considered cost as a first-class concern in selecting adaptation options at runtime. Their approach leverages the **Cost-Benefit-Analysis Method (CBAM** [19]**)**, transferring this classic architecture selection method to runtime. The approach is applied to an IoT application where the cost corresponds to the energy required to send adaptation actions from the managed system deployed at the gateway to the nodes of the IoT network.

In summary, several approaches have demonstrated the value of considering the cost of applying adaptation actions. Their main focus is on the expected time between enacting an adaptation action and observing its effects, and on the resources required to apply the adaptation actions.

2.6 Estimated Risk in the Decision-Making of Self-Adaptive Systems

Estimated risk is rarely considered in the decision-making process of self-adaptive systems. With estimated risk, we refer to potential effects of uncertainties on system objectives in terms of their likelihoods and consequences (positive, negative, or both) [13]. Traditionally, risk estimation in software is a human-driven activity performed by system designers and architects at design time. However, in self-adaptive systems, the risk may change over time possibly impacting users as well as the system itself. To address this issue, we argue that the risk should be addressed throughout the entire life cycle of the system. This implies that in addition to cost and benefit, risk estimation should be made an indispensable concern of runtime decision-making in self-adaptive systems.

Almeida and Vieira apply in [4] a risk-based approach to define change loads for resilience benchmarks for self-adaptive systems. In [4], Reichstaller and Knapp target testing of self-adaptive systems, focusing on risk-based goals and the detection of hazardous failures. Cailliau and van Lamsweerde highlight in [14] that the satisfaction rate of the goals of self-adaptive systems depends on the rate at which adverse conditions prevent their satisfaction. Obstacle analysis is a goal-oriented approach to risk analysis where obstacles to system goals are identified, assessed, and resolved through countermeasures yielding new goals. The selection of appropriate countermeasures relies on the assessed likelihood and criticality of obstacles together with environmental assumptions. To meet the system's goals under changing conditions, the authors proposed an approach for runtime obstacle resolution. The approach relies on a model where goals and obstacles are refined and specified in a probabilistic linear temporal logic. The approach allows for (a) monitoring the satisfaction rate of probabilistic leaf obstacles; (b) determining the severity of their consequences by up-propagating satisfaction rates through refinement trees from leaf obstacles to high-level probabilistic goals; and (c) dynamically shifting to alternative countermeasures that better meet the required satisfaction rate of the system's high-level goals under imposed cost constraints.

In summary, estimated risk in the decision-making process of self-adaptive systems has been largely ignored so far. Given the growing importance of risk mitigation in terms of safety/privacy of users, environmental impact, and ethical or legal concerns, it is crucial to incorporate estimated risk as a first-class concern in the decision-making of self-adaptive systems.

2.7 Reference Model

There is no common definition of what constitutes a reference model. In this article, we refer to a reference model as a set of functional entities with relationships between these entities that together solve a given problem. A reference model is abstractly defined, and is domain and technology agnostic. A concrete architecture maps the functions to concrete components.

The pioneering reference model in the field of self-adaptation is MAPE-K [47]. MAPE-K defines the essential functions of a feedback loop of a self-adaptive system and their relationships. We highlight two other well-known reference models for self-adaptation. First, **Formal Reference Model for Self-Adaptation (FORMS)** [77] provides a small set of formally specified modeling elements that correspond to the key concerns in the design of self-adaptive software systems, and a set of relationships that guide their composition. FORMS provides three complementary perspectives: computational reflection, distributed coordination, and MAPE-K. Second, **Dynamic Adaptive, Monitoring and Control Objectives (DYNAMICO)** [69] model aims at addressing: (i) the management of adaptation properties and goals as control objectives; (ii) the separation of concerns among feedback loops required to address control objectives over time; and (iii) the management of dynamic context as an independent control function to preserve context-awareness in the adaptation mechanism. To that end, the DYNAMICO reference model integrates three types of feedback loops that focus on the control objectives, the target system adaptation, and dynamic monitoring, respectively.

In summary, the field of self-adaptation has produced a number of reference models that define core functions of self-adaptive systems. By consolidating existing knowledge in the decision-making of self-adaptive systems, our article aims to outline a reference model for decision-making in self-adaptive systems that take into account benefit, cost, and risk as core concerns.

2.8 Architectural Viewpoint and View

Architectural viewpoints and views are a common approach to documenting the architecture of software-intensive systems [1, 38]. The IEEE 1471 and ISO/IEC 42010 standards [1] offer widely accepted conceptual definitions of architectural viewpoints and views. Specifically, an *architectural viewpoint* is defined as "a work product establishing the conventions for the construction, interpretation, and use of architecture views to frame specific system concerns." An architectural viewpoint gives architects the means to express a coherent set of concerns, the stakeholders interested in these concerns, and model kinds (i.e., meta-models) that frame the concerns, each defining notations, modeling templates, analytical methods and possibly other useful operations on models of the model kind. A viewpoint can be instantiated for a domain at hand, resulting in an *architectural view*. An architectural view comprises architectural models that are developed using the conventions and methods established by its associated viewpoint. An architectural model may participate in more than one view. Although viewpoints have become increasingly popular for describe software architectures, their adoption in the domain of self-adaptive systems is rather limited.

Galster and Avgeriou [31] proposed a variability viewpoint for constructing views of enterprise software systems. The viewpoint comprises several complementary models that address detailed

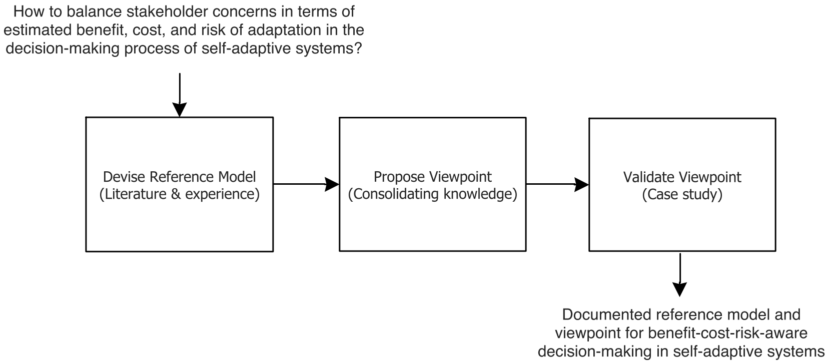


Fig. 1. Research approach used in this study.

variability concerns. The viewpoint facilitates the representation and analysis of variability (i.e., the ability of software to be adapted for a specific context to enable multiple deployments of a software system) in the architecture of an enterprise software system. Musil et al. [55] documented an architecture viewpoint for continuous adaptation management in **Collective Intelligence Systems (CIS)**. The viewpoint frames concerns of stakeholders with an interest in handling CIS-specific adaptation across the entire system's life cycle and includes a set of four model kinds for identifying, designing, and realizing adaptation in CIS key elements. The viewpoint provides support for software architects to deal with self-adaptive systems in CIS.

In summary, architectural viewpoints and views offer the means to document and analyze software designs from the perspective of stakeholder concerns. Leveraging a reference model (pointed out above), this article aims to present an architectural viewpoint for the decision-making component of self-adaptive systems. This viewpoint should consider the estimated benefit, cost, and risk of adaptation options in the decision-making process of self-adaptive systems.

3 Research Approach

In this section, we start by outlining the three-stage research approach we used in this research. Then, we introduce an example application that we use as a running case throughout this article.

3.1 Three-Stage Research Approach

The research presented in this article aims to answer the following research question:

> *How to balance stakeholder concerns in terms of estimated benefit, cost, and risk of adaptation in the decision-making process of self-adaptive systems?*

To answer this research question, we devised a three-stage research approach shown in Figure 1, leveraging principles of design science [79, 80]. The first two activities consolidate knowledge through engineering activities, the third activity validates the results though an empirical study.

Synthesize Reference Model. Driven by the research question and based on a thorough analysis of the literature including [53, 58], a number of literature reviews of the community such as [36], and our own experience in designing and developing self-adaptive systems in a variety of domains, e.g. [17, 68, 74, 75], we synthesize a reference model for decision-making in self-adaptive systems that integrates benefit, cost, and risk as core concerns.

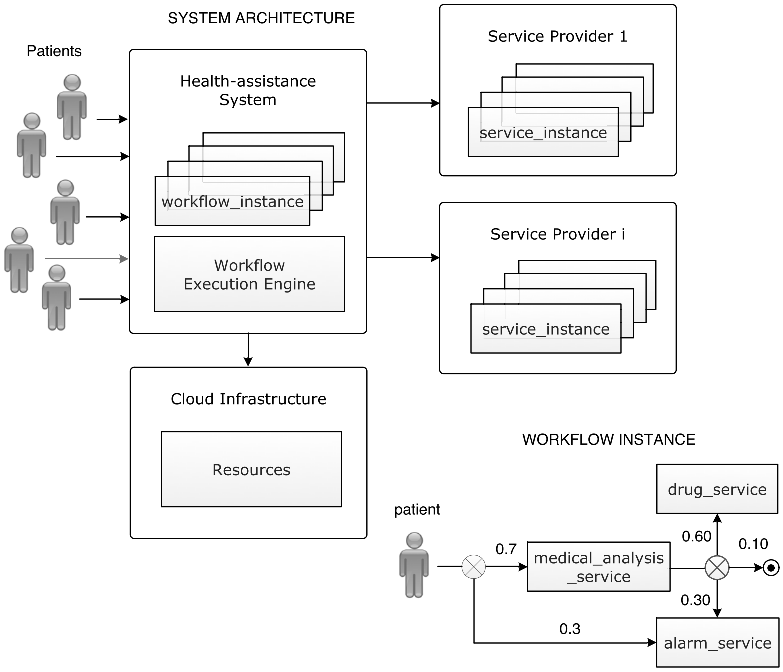


Fig. 2. Service-based system: system architecture and workflow instance.

Propose Viewpoint. Consolidating the knowledge of the reference model and leveraging experience with architectural design in general and viewpoints in particular, we then propose a viewpoint for benefit-cost-risk-aware decision-making in self-adaptive systems. The viewpoint frames the essential concerns of stakeholders with an interest in the runtime decision-making of self-adaptive systems that are subject to uncertainties. In particular, the viewpoint defines a set of model kinds for identifying, designing, and realizing a decision-making module for self-adaptation taking into account the estimated benefit, cost, and risk of the relevant adaptation options. The viewpoint should support integrating existing methods for the analysis of adaptation options as well as be open for future extensions.

Validate Viewpoint. To validate the architectural viewpoint, we conducted a case study evaluating the usefulness, understandability, and applicability of the viewpoint from the perspective of a group of participants with experience in the design and development of self-adaptive systems. The case study consisted of two design tasks and a survey which were carried out by the participants. Where applicable, the results from the case study were cross-validated against the survey results using triangulation. Our study resulted in a documented reference model and a validated viewpoint for benefit-cost-risk-aware decision-making in self-adaptive systems.

3.2 Running Example: A Service-Based e-Health Assistance System

Consider a simple service-based system as shown in Figure 2. This system offers a remote health assistance service to elderly patients and relies on data collected via wearable devices. The health-assistance service is realized by a set of specific services that are executed in a workflow as shown in the figure.

The core of the application exploits the resources of a cloud infrastructure. Each request of a patient instantiates a new instance of the health-assistance service workflow. This workflow then interacts with the services, following the invocation pattern defined by the workflow. A *Medical Service* receives messages with values of vital parameters from the patient's health device. The service analyzes the data, and depending on the analysis results either requires no action, or it instructs a *Drug Service* to notify a local pharmacy to deliver new medication to the patient, or change the dose of medication, or it instructs an *Alarm Service* in case of an emergency to request that the patient is visited by medical staff. The alarm service can also directly be invoked by a user via a panic button. The numbers associated with arrows in the workflow express probabilities that actions are invoked. These numbers represent uncertainties that may change over time. Each service can be implemented by a number of service providers that offer concrete services according to a service-level agreement that specifies the reliability and accuracy of the service, among other aspects. Some of the properties of services may change at runtime. For example, due to the changing workloads on the provider side or unexpected network failures, the reliability of a service may deviate from the one specified in its service-level agreement. Each service provider also offers a privacy policy that specifies how patient's data is managed.

At runtime, it is possible to pick any combination of the services offered by the service providers. The adaptation goals that express the benefit of adaptation are to keep the average failure rate low while minimizing the resources required to run the e-health service. Switching services in the system may imply a cost in terms of the extra resources that are required to test newly selected services before they can be used. Given that service providers may use different privacy policies on how patient data is managed, selecting a service from a service provider implies a risk to the confidentially of the data of patients within legal constraints (e.g., kept strictly local, stored with partners, shared with partners, or non-specified). Finally, medical analysis services perform their analysis based on the measurements of a limited period using bounded computational resources, and therefore, there is a risk that the diagnoses derived from the analysis results may not be 100% accurate. This may indirectly affect the health conditions of the patients.

4 Reference Model

To address the problem of selecting adaptation options for a self-adaptive system that balances stakeholder concerns in terms of estimated benefit, cost, and risk we devised the reference model shown in Figure 3. The reference model is grounded in a thorough analysis of the literature complemented with our own experiences in designing and developing self-adaptive systems in a wide variety of domains, including [17, 36, 53, 58, 68, 74, 75].

The reference model defines the primary functions that are required to select adaptation options taking into account estimated benefit, cost, and risk. We explain the functions for one adaptation option. This process is repeated for all adaptation options that are considered for decision-making. The first function is benefit estimation which takes as input the current configuration of the managed system, an adaptation option, the benefit concerns, and the adaptation goals resulting in an estimated benefit for that adaptation option. The current configuration represents the aspects of the managed system and the environment that are relevant to adaptation at the time of decision-making. Benefit concerns are stakeholder issues regarding the qualities of the system that can be obtained (positively or negatively) by adapting the system. Adaptation goals represent the quality requirements that need to be achieved by the managing system. Examples of benefit concerns for the service-based system are the reliability and performance of service invocations. The adaptation goals are keeping the failure rate low, while minimizing the resource usage for running the service.

The second function is cost estimation. Besides the current configuration of the system and an adaptation option, this function takes as input cost concerns and cost metrics resulting in an

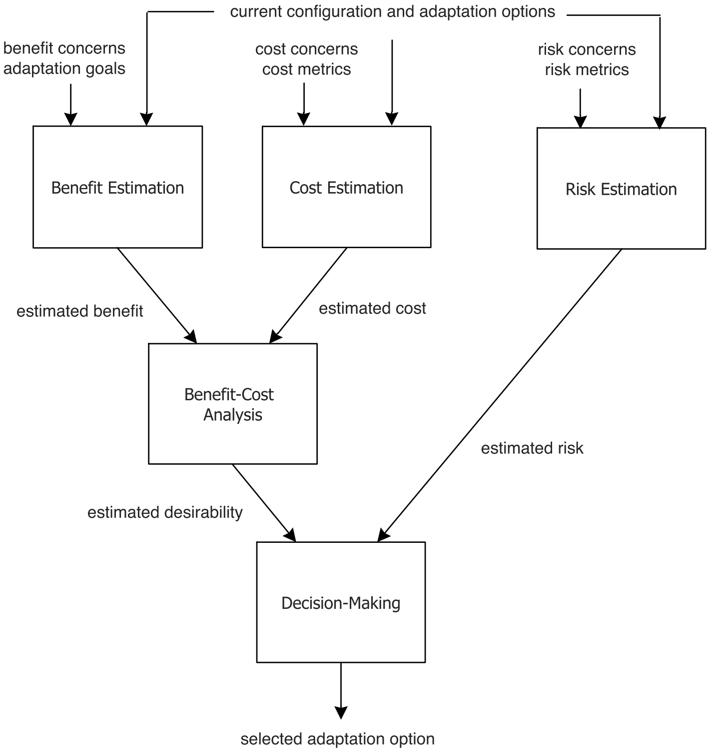


Fig. 3. Reference model for benefit-cost-risk-aware decision-making in self-adaptive systems.

estimated cost for the adaptation option. Cost concerns are stakeholder issues regarding the cost implied by adapting the system. Cost metrics define measures to quantify the cost concerns. For instance, properly initiating new services when adapting the service-based system may imply a cost to test these new services. This cost may be measured by the resources required for these tests that may depend on the **Service Level Agreements (SLAs)** made with the service providers.

The third function is a benefit-cost analysis which takes as input the estimated benefit and estimated cost for an adaptation option and produces an estimated desirability for that option. The estimated desirability expresses the degree to which stakeholders prefer one adaptation option over the other options by comparing the overall estimated benefit with the overall estimated cost. For the service-based system, the estimated desirability of adaptation options can for instance be represented by the **Value-for-Cost (VFC)**, that is, the benefit determined by the failure rate and resource usage divided by the cost determined by the resources required for testing newly required services.

The fourth function is risk estimation which takes as input the current configuration, an adaptation option, risk concerns, and risk metrics, and produces an estimated risk for the adaptation option. Risk concerns are stakeholder issues regarding the risks that may be implied by adapting the system. Risk metrics define measures to quantify risk. A consequence/likelihood matrix is an example to represent risk metrics. Such a matrix enables specifying an estimated risk according to its consequence and likelihood based on stakeholder input. In the service-based system example, the likelihood of data exposure (for the risk of confidentiality of data) may range over a 4-point scale from rarely to almost certain, while the consequences can range from negligible effect to significant impact. In general, risks may arise either from incomplete data or from an imperfect method or algorithm that deals with data even if the data itself is perfect.

The fifth and last function is decision-making which takes as input the estimated desirability and estimated risk of the adaptation options considered for adaptation and selects an adaptation option that balances benefit, cost, and risk according to the concerns of stakeholders of the system. In the service-based system example, the decision-making mechanism may select the adaptation option that maximizes the difference between weighted desirability and risk. The selected adaptation option represents the new configuration that is selected for adaptation and will be applied to the managed system. In the service-based system example, the selected adaptation option comprises the set of services that the workflow needs to invoke; some of the current services may remain in use, others may need to be replaced by newly selected services.

Together, the five functions of the reference model with their interactions select adaptation actions in self-adaptive systems that balance stakeholder concerns in terms of estimated benefit, cost, and risk. These functions are domain and technology-agnostic. The concrete realization of the functions requires architectural solutions for the domain and problem at hand. Such solutions may have a one-to-one mapping of these functions to components or any other type of mapping.

Benefits, Costs, and Risks as First-Class Citizens. While the reference model (Figure 3) may suggest differently, the viewpoint treats benefits, costs, and risks as equal first-class concerns. Indeed, the risks appear after benefit-cost analysis, yet the final selection of an adaptation option (decision-making step) considers the three concerns for each adaptation option as equal factors. The intermediate step of determining the desirability of each adaptation option based on benefits and costs aims to make the specification of the decision-making process easier for the architect and the other stakeholders. In particular, the proposed model enables the stakeholder to consider "what they desire" with "what risks are they willing to accept" when adapting the system. In principle, the intermediate step to determine desirability could be omitted and decision-making could directly take the estimates of benefit, cost, and risk as input to select an adaptation option. Yet there are good arguments to use the intermediate step. First, directly considering benefit, cost and risk as input for decision-making would complicate the decision-making mechanism. In particular, it requires the stakeholders to express their preferences about how to balance the three types of concerns into a single decision-making mechanism. Second, direct integration of the three types of concerns would add complexity to test and tune the integrated decision-making mechanism. Third, considering benefit-cost analysis as a distinct step adds to the understandability of the decision-making of a self-adaptive system and supports communication with the stakeholders. A recent survey with practitioners has shown that the understandability of adaption decisions is important for the trustworthiness of self-adaptive systems [73]. Note that the result of integrating the three concerns in a single step would yield the same results of adaptation decisions if the appropriate weight values are calculated and applied to each of the three concerns. We refrained from offering the option to architects to use or not use the intermediate step of determining desirability as we believe that the flexibility of this generalization may not necessarily improve the usability of the viewpoint.

5 Viewpoint

We now present the viewpoint for benefit-cost-risk-aware decision-making in self-adaptive systems. This viewpoint establishes the conventions for defining and using architecture views to deal with the concerns of stakeholders for decision-making in self-adaptation by taking into account the estimated benefit, cost, and risk of adaptation options.

The viewpoint leverages the reference model for benefit-cost-risk-aware decision-making in self-adaptive systems. It maps the functions of the reference model to design artifacts, offering architects and designers the means to specify concrete views for benefit-cost-risk-aware decision-making modules of self-adaptive systems.

We start with outlining the scope of the viewpoint. Then we provide a specification of the viewpoint, including the stakeholders with their concerns, model kinds, and viewpoint analysis. We illustrate the different parts using the running example of the service-based system.

5.1 Scope

Given the wide range of approaches used to realize self-adaptation in general and decision-making in self-adaptive systems in particular, it is essential that we clarify the scope of the viewpoint before we present its specification. In this work, we adopt the widely accepted conceptual definitions of architectural viewpoints and models from the IEEE 1471 and ISO/IEC 42010 standards [1].

The viewpoint is centered on *architecture-based adaptation* [33, 49, 57, 71, 77], which is an established approach for engineering self-adaptive systems. Furthermore, the viewpoint aligns with the standard MAPE-K reference model [47, 78]. In particular, the focus of the viewpoint is on the decision-making of self-adaptation, which maps to the analysis function and the first part of the planning function [71]. The analysis function evaluates the relevant adaptation options and the first part of planning selects the best option based on the estimated values for benefit, cost, and risk of all analyzed adaptation options. The planner function then determines the best option to adapt the running system. As such, architects can combine the viewpoint with additional architectural approaches, such as complementary viewpoints or patterns, to deal with other concerns of realizing a self-adaptive system, such as monitoring the system and its environment, keeping runtime models up to date, generating adaptation plans, and executing the adaptation actions of a plan.

The focus of the viewpoint is on the adaptation concerns, in particular, the selection of a configuration from the possible configurations to adapt the system. We refer to the set of possible configurations to select from as the *adaptation options* and refer to the complete set as the *adaptation space*. With *relevant adaptation options* we refer to the adaptation options that are deemed to be relevant and are actually analyzed, which can be the complete adaptation space or a subset of it, determined using some heuristic or selection mechanism.

The viewpoint is concerned with uncertainties related to *anticipated change*, i.e., the architect has knowledge of the types of changes that may occur, but not when these changes occur and in what way they may occur [36] (for instance the frequency of changes or their intensity). Uncertainties related to unanticipated change are not supported by the viewpoint. Decision-making in the viewpoint is defined based on abstract functions associated with the estimated benefit, cost, and risk of adaptation options. This allows to support different types of decision-making mechanisms, for instance, based on rules, utilities, or softgoals.

The approaches used for estimating benefit, cost, and risk build on the CBAM [19] and the IEC 31010:2019 standard on risk management and risk assessment techniques [13]. CBAM is an established method for analyzing the benefits and costs of architectural designs of software-intensive systems. CBAM takes into account the uncertainty factors regarding benefits and costs, providing a basis for informed decision-making about architectural design or upgrade. In contrast to CBAM, we require an automated approach that makes adaptation decisions for the system at runtime to deal with uncertainties. On the other hand, the IEC 31010:2019 standard on risk management and risk assessment provides guidance on the selection and application of various techniques that take into account risk in the decision-making process when mitigating uncertainty. Whereas the standard focuses on techniques that are used to aid decision-making under uncertainty in general, in this viewpoint we require techniques that can be applied automatically by a system at runtime to make adaptation decisions on the fly under uncertainty.

5.2 Specification of the Viewpoint

The viewpoint frames the essential concerns of stakeholders of the system with an interest in the runtime decision-making of self-adaptive systems that are subject to uncertainties. In particular, the viewpoint defines a set of model kinds for identifying, designing, and realizing a decision-making module for self-adaptation taking into account the estimated benefit, cost, and risk of possible relevant adaptation options. The model kinds show the relevant architectural information that is essential to guide a successful design of the decision-making module and can be instantiated for a problem at hand to handle uncertain but anticipated changes in the system, its goals, and the environment. It is important to note that the instantiated model kinds fit within the context of the design of a feedback loop, adhering to the widely known and applied MAPE-K reference model.

5.3 Stakeholders and Concerns

Table 1 shows an overview of the viewpoint as well as the stakeholders and their concerns.

Stakeholders of the viewpoint are *architect(s)*, *owner(s)*, *user(s)*, and *other(s)*. Architects are primarily interested in technical aspects, in particular, the design and behavior of the decision-making module of the self-adaptive system taking into account the estimated benefit, cost, and risk. Owners have a primary interest in the benefit of the self-adaptive system, the cost that may be induced by adaptation, and the risk that may be implied by adaptations of the system.[2] Architects and owners have also an interest in the desirability of system configurations; a configuration with a high desirability has a high benefit and a low cost. Users are primarily interested in the benefit provided by the self-adaptive system as well as the risk they may be exposed to. Others are those people who may be exposed to potential risks implied by the system, directly or in the environment.

In summary, the viewpoint addresses the following adaptation concerns of the stakeholders: *benefit* of adaptation (architects, owners, and users), *cost* of adaptation (architects and owners), *desirability* of adaptation (architects, owners), *risk* implied by adaptation (architects, owners, users, and others), and *decision-making* for adaptation (architects).

Example. The architects of the health assistance system are the persons who design and oversee the realization of the decision-making module of the feedback loop that selects adaptation options to adapt the services used by the workflow. The architects' main concern is to ensure that the system makes proper adaptation decisions, i.e., select adaptation options balancing their estimated benefit, cost, and risk. The system owners are the persons who operate the health-assistance service. The main concerns of the system owners are to provide a good service to the users, optimize the resources to operate the system, keep the cost required for adaptation low, and minimize the exposed risk. The users are the patients who use the service via a wearable device either to analyze vital parameters or to directly alarm a medical team in case of an emergency. Their main concerns are the reliability of the service and risks in terms of data privacy and health risks of the application. Finally, others are people who may be exposed to risks implied by the system directly or indirectly, in particular risks with respect to data privacy and decisions made by the system. Others can be relatives and friends of patients, and healthcare professionals.

5.4 Model Kinds

The viewpoint comprises five model kinds. Table 2 presents the first three model kinds: *benefit estimation*, *cost estimation*, and *benefit-cost analysis*. Table 3 presents the last two model kinds: *risk*

---

[2] In this viewpoint, we use the term *benefit attribute* to refer to different dimensions of estimated benefit; similarly we use the terms *cost attribute* and *risk attribute* to refer to different dimensions of estimated cost and risk respectively.

Table 1. Viewpoint—Overview, Stakeholders, and Concerns

Overview

The architecture viewpoint deals with the main stakeholder concerns related to the decision-making of self-adaptive systems that need to handle uncertain but anticipated changes in the environment, the system, and its goals. The viewpoint takes into account the estimated benefit, cost, and risk as first-class citizens when selecting adaptation options. The viewpoint offers model kinds that can be instantiated for a problem at hand. The model kinds show the relevant architectural information that is essential to guide a successful design of benefit-cost-risk-aware decision-making modules of self-adaptive systems.

Stakeholders

Architect(s) who design the decision-making module of a self-adaptive system.

Owner(s) who operate the system and offer its service to users.

User(s) who use (and pay) for the service of the system.

Other(s) who may be exposed to potential risk implied by the system.

Concerns

C1—Benefit. What are the adaptation goals of the self-adaptive system? What are the estimated values of the quality attributes, i.e., benefit attributes, for an adaptation option that corresponds with the adaptation goals? How can the adaptation goals be combined to determine the overall estimated benefit of an adaptation option?

C2—Cost. What are the different types of costs associated with performing adaptation? How can the cost for each type be quantified? How can the cost for each type be estimated for an adaptation option? How can the cost for different types be combined to determine the overall estimated cost of an adaptation option?

C3—Desirability. How to balance the benefit against the cost of an adaption option in order to express its estimated desirability? How to compare and rank desirability estimates?

C4—Risk. What are the relevant types of risk for the system? How can the risk for each type be quantified? How can the risk for each type be estimated for an adaptation option? How can the estimates of different types of risk be combined to determine the overall estimated risk of an adaptation option?

C5—Decision-Making. What options are available for adapting the system from the current configuration? What elements need to be considered when selecting an adaptation option to adapt the current configuration? How to balance the desirability of adaptation options with estimated risk when making adaptation decisions?

estimation and *decision-making*. We start by explaining the utilization process of the model kinds. Then, we explain the model kinds in detail and provide illustrations using the running example.

Model Kinds Utilization Process. Figure 4 shows an overview of the model kinds with the utilization process that takes as input the well-defined concerns C1 to C5 (see Table 1).[3] Leveraging the reference model for benefit-cost-risk-aware decision-making in self-adaptive systems, the process starts with the design of the benefit estimation model using the benefit estimation model kind. This model deals with concerns regarding the benefit of adaptation options (C1 in Table 1). Next—or

---

[3] Appendix E provides an overview of the main abbreviations used in this article.

Table 2. Viewpoint—Model Kinds MK1, MK2, and MK3

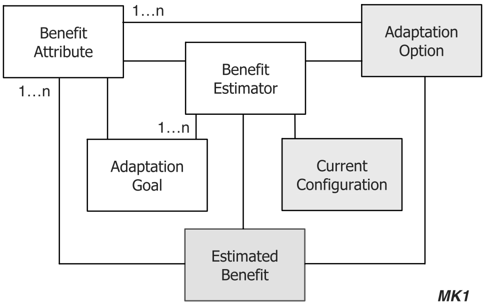
Model Kinds (Description)

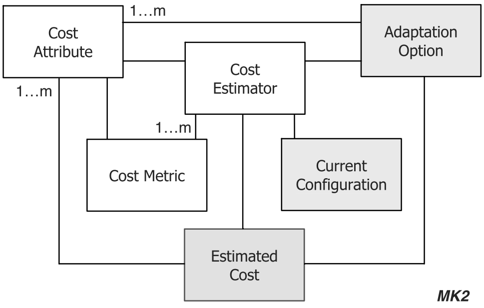
MK1—Benefit Estimation (Deals with Concern C1). Describes per adaptation option how each quality attribute is estimated based on the associated adaptation goal and what the overall estimated benefit is based on the combined adaptation goals.

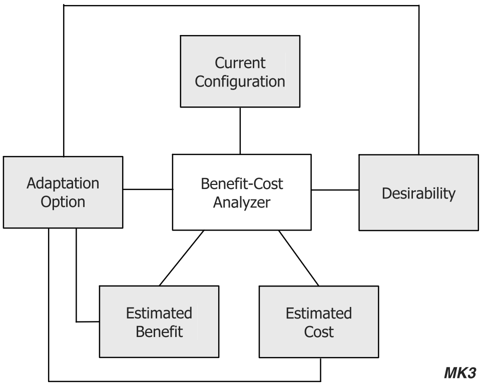
MK2—Cost Estimation (Deals with Concern C2). Describes per adaptation option how each cost attribute is estimated based on the associated cost metric and what the overall estimated cost is based on the combined cost metrics.

MK3—Benefit-Cost Analysis (Deals with Concern C3). Describes per adaptation option how the desirability is estimated based on the analysis of the estimated benefit and the estimated cost of that adaptation option.

Model Kinds (Meta-Models)







Key: UML (grey boxes represent model elements shared among model kinds).

in parallel—the cost estimation model can be designed using the cost estimation model kind. This model deals with concerns regarding the cost of adaptation options (C2 in Table 1). The next step in the process is the design of the benefit-cost analysis model. This model takes as input the benefit-cost analysis model kind, the benefit estimation model, and the cost estimation model. The benefit-cost estimation model deals with the concerns of the desirability of adaptation options

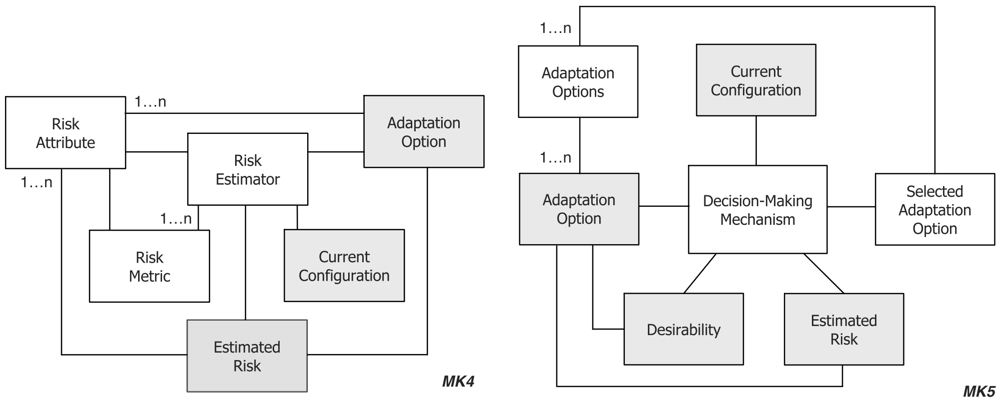
Table 3. Viewpoint—Model Kinds MK4 and MK5

Model Kinds (Description)

MK4—Risk Estimation (Deals with Concern C4). Describes per adaptation option how each risk attribute is estimated based on the associated risk metric and what the overall estimated risk is based on the combined risk metrics.

MK5—Decision-Making (Deals with Concern C5). Describes how an adaptation option is selected from the set of adaptation options to adapt the system from its current configuration based on the desirability and estimated risk of the adaptation options.

Model Kinds (Meta-Models)



Key: UML (grey boxes represent model elements shared among model kinds).

(C3 in Table 1). Then—or in parallel—the risk estimation model is designed using the risk estimation model kind. This model deals with the concerns of risk of adaptation options (C4 in Table 1). In the last step, the decision-making model is designed. This step takes as input the decision-making model kind, the cost-benefit estimation model, and the risk estimation model. The decision-making model deals with the concerns of decision-making, i.e., selecting the best adaptation option (C5 in Table 1). The result of the process is a decision-making model that addresses the stakeholder concerns C1 to C5. This model allows developers to implement a decision-making component that can then be used by the managing system to select adaptation actions at runtime.

Benefit Estimation Model Kind (MK1). This model kind describes how the benefit of each relevant adaption option is estimated (see Table 2 top left). The *current configuration* is a representation of the aspects of the managed system and the environment that are relevant to adaptation at that time. These aspects include the current component configuration of the managed system, the settings of relevant system parameters, the values of the quality properties of interest, and the values of uncertainties that are relevant to adaptation. An *adaptation option* is a possible configuration that can be reached from the current configuration by adapting the system. An *adaptation goal* represents a requirement that needs to be achieved by the managing system. Adaptation goals usually refer to quality requirements. A *benefit attribute* of an adaptation option represents the estimated value for a system property of the managed system that corresponds with an adaptation goal. Benefit attributes are usually quality properties. With each benefit attribute, there is one

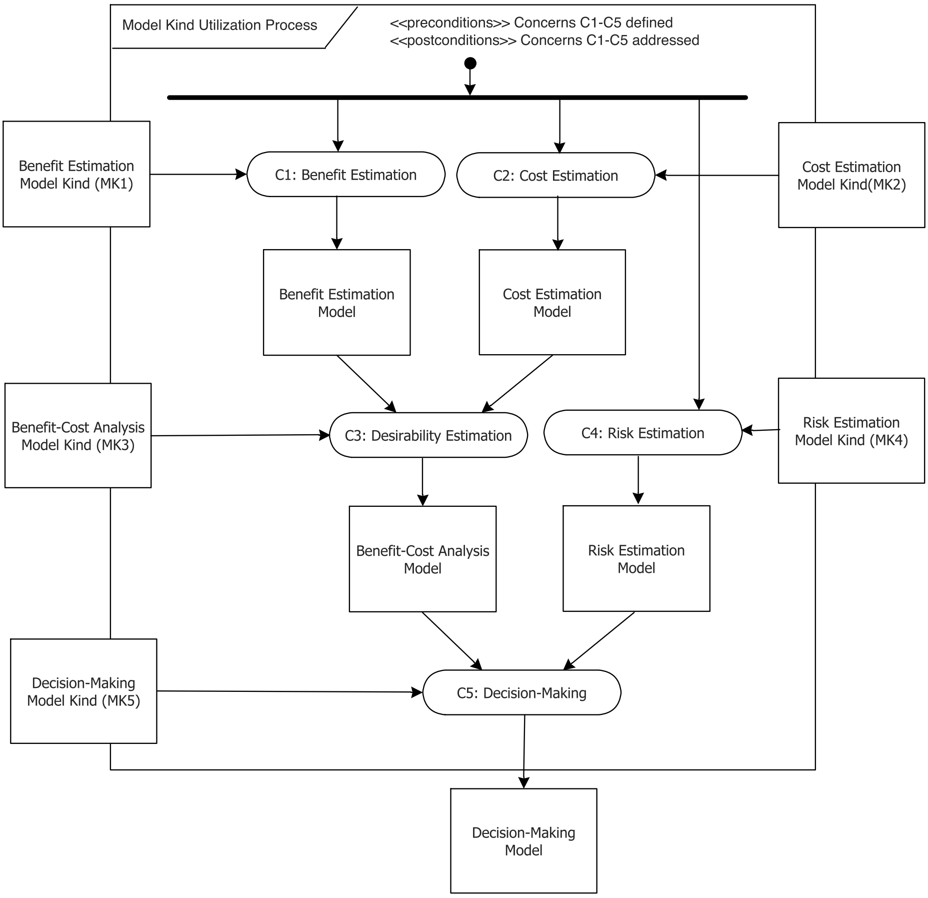


Fig. 4. Overview of model kinds with the process to use them.

corresponding adaption goal. A *benefit estimator* is a mechanism that enables estimating the benefit attributes of the adaptation options. The *estimated benefit* represents the overall estimated benefit of an adaptation option based on the estimated benefit attributes and combined adaptation goals.

Example. Figure 5 instantiates the benefit estimation model kind for the health assistance system. The current configuration of the system consists of the workflow with a set of services currently in use, a set of properties referring to uncertainties including the actual values associated with the different paths exercised in the workflow, the current values of the failure rate, and resource usage of the system and service-level agreements. An adaptation option corresponds to a particular selection of concrete services provided by service providers to be executed by the workflow. The service-based system has two benefit attributes: failure rate and resource usage. The corresponding adaptation goals are represented with utility responses.

Figure 6 shows the utility response curves for both failure rate and resource usage goals. A utility-response curve depicts how the utility derived from a particular response varies as the response varies. Utility encodes a measure of goodness or badness of the result of choosing a particular adaptation option in a self-adaptive system [34]. The utility can vary linearly, non-linearly, as a step-function, or any combination of these. Utility response curves are elicited from stakeholders;

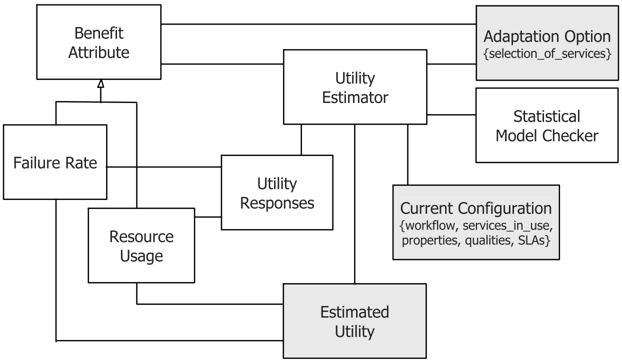


Fig. 5. Example instance of benefit estimation model kind.

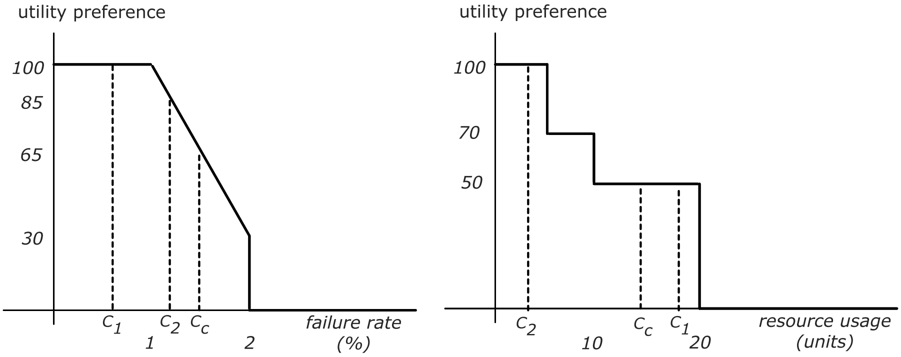


Fig. 6. Example of utility response curves.

it is well known that this can be a tedious process [8]. As shown in the left diagram, the utility preference for configurations with failure rates below 1% is 100%. For configurations with failure rates between 1% and 2% the utility preference gradually decreases to 30%. The utility preference of configurations with failure rates above 2% is zero. The right-side diagram shows the utility preference for resource units. As an example, configuration C_1 has a 100% utility preference for a failure rate of 0.7% and 50% for a resource usage of 15 units. Adaptation option C_2 on the other hand has a utility preference of 65% for a failure rate of 1.5% and 70% for a resource usage of 7 units. The utility estimator determines the estimated utilities for each relevant adaptation option. To that end, the estimator configures a runtime model of the workflow for each combination of concrete services (i.e., adaptation options) together with the actual probabilities that paths are selected. This model is then analyzed by a statistical model checker that runs a number of simulations for each adaptation option. The result of the analysis is an estimate of the failure rate and resource usage for each adaptation option with the required accuracy and confidence. The estimated benefit is then determined using a utility function that computes the sum of the weighted values of the estimated quality attributes for each adaptation option. Weights express the relative importance of the benefit attributes for the stakeholders.

In particular, to determine the estimated benefit for an adaptation option we take the sum of the difference between the estimated utility for that adaptation option and the utility of the current configuration for each benefit property taking into account the respective weights. As an example,

Table 4. Cost Attributes with Cost Metrics

Cost Type	Cost Attribute	Cost Metric
Resources	Communication	Required bandwidth
Resources	Computation	Required processing resources
Resources	Storage	Required memory
Resources	Power	Required energy
Overhead	Availability	Degree of reduced service
Overhead	Performance	Degree of degraded user experience
Overhead	Security	Cost to manage exposed vulnerability
Economic	Financial	Monetary price

the estimated benefit of adaptation options can be computed as follows:

$$EB_{C_i} = (U^F_{C_i} - U^F_{C_c}) * W^F + (U^R_{C_i} - U^R_{C_c}) * W^R, \tag{1}$$

with $U^F_{C_i}$ the utility of adaptation option C_i for failure rate F, $U^F_{C_c}$ the utility of the current configuration C_c for the failure rate, and $W^F = 0.7$ the weight for failure rate; and the second term with a similar structure referring to resource usage R, where $W^F = 0.3$. Applied to adaptation option C_1 in Figure 6 the estimated benefit is:

$$EB_{C_1} = (100 - 65) * 0.7 + (50 - 50) * 0.3 = 24.5. \tag{2}$$

Similarly, the benefit of adaptation option C_2 in Figure 6 is:

$$EB_{C_2} = (85 - 65) * 0.7 + (100 - 50) * 0.3 = 29.0. \tag{3}$$

In this particular case, adaptation option C_2 has a higher estimated benefit compared to adaptation option C_1. Hence, if an adaptation decision was made based on estimated benefit only, adaptation option C_2 would be selected for adaptation.

Cost Estimation Model Kind (MK2). This model kind describes how the estimated cost of applying each adaption option is determined (see Table 2 top right). A *cost attribute* represents a particular type of cost that is implied by adaptation of the managed system. Different cost attributes can be associated with one adaptation option, such as communication overhead implied by adaptation, extra resources required to perform adaptation, and temporal restrictions in the availability of the system functionality as a consequence of adaptation. *Cost metrics* define measures for quantifying cost attributes. A *cost estimator* is a mechanism that enables estimating the cost attributes of the adaptation options. The *estimated cost* represents the overall estimated cost of an adaptation option based on the estimated cost attributes and the combined cost metrics. To support architects with identifying cost attributes for a problem at hand, we devised a primary list of common cost attributes with associated cost metrics (see Table 4). This table is based on examples extracted from the literature and our own experiences, see for instance [44, 48, 68]. We distinguish three groups of adaptation costs. The first group refers to resources that are required to realize adaptation, such as bandwidth, processor time, memory, and power. The second group relates to the overhead of the system in terms of quality properties that are affected by adaptation, which can be reduced availability, a performance penalty, or the cost to deal with additional security vulnerabilities implied by adaptation. The third group refers to a monetary price that is implied by the realization of adaptation. Table 4 is not meant to be exhaustive and can be easily refined or extended as needed.

Example. Figure 7 instantiates the cost estimation model kind for the health assistance system.

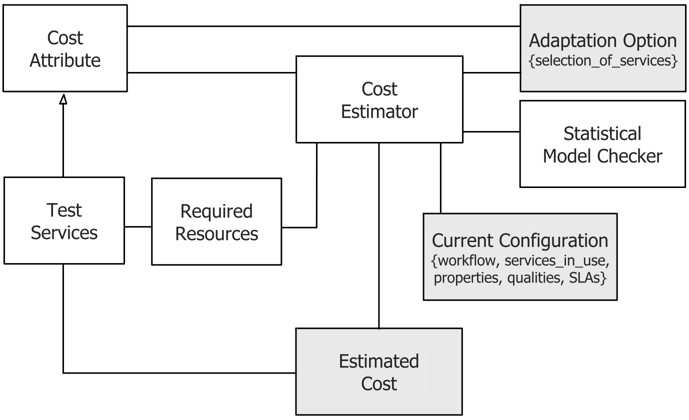


Fig. 7. Example instance of cost estimation model kind.

Table 5. Example Cost Model for e-Health System with Required Resources for Services

		Test Overhead		
Provider	**SLA**	**Medical Analysis Service**	**Drug Service**	**Alarm Service**
SP1	Silver	5	6	2
SP2	Gold	3	2	2
SP3	Bronze	8	4	4

In this example, we consider one cost attribute: the test overhead of new services that are considered by the adaptation options [15]. Testing overhead requires extra resources, which corresponds to computation overhead in Table 4. The amount of testing that is needed may depend on the trustworthiness of service providers which is specified in the SLA. In this example, gold represents high, silver medium, and bronze low trustworthiness levels. Table 5 illustrates a possible cost model to test new services for the e-health system.

Assume that the current configuration comprises the following set of services:

$$C_c = \{S^{MAS}_{SP1}, S^{DS}_{SP3}, S^{AS}_{SP1}\}. \tag{4}$$

MAS is a medical analysis service provided by service provider $SP1$, DS is a drug service provided by $SP3$, and AS is an alarm service provided by $SP1$. Consider now two adaptation options C_1 and C_2 composed as follows:

$$C_1 = \{S^{MAS}_{SP1}, S^{DS}_{SP2}, S^{AS}_{SP2}\}; \quad C_2 = \{S^{MAS}_{SP1}, S^{DS}_{SP1}, S^{AS}_{SP1}\}. \tag{5}$$

The estimated cost to test the adaptation option of configuration C_1 is:

$$EC_{C_1} = cost(C_c, C_1) = cost_{adapt}(S^{DS}_{SP3}, S^{DS}_{SP2}) + cost_{adapt}(S^{AS}_{SP1}, S^{AS}_{SP2}) = 2 + 2 = 4. \tag{6}$$

The estimated cost to test the adaptation option of configuration C_2 is:

$$EC_{C_2} = cost(C_c, C_2) = cost_{adapt}(S^{DS}_{SP3}, S^{DS}_{SP1}) = 6. \tag{7}$$

Despite the fact that adaptation option C_1 requires testing two new services and C_2 requires testing only one new service, the estimated cost of C_1 is smaller than the estimated cost of C_2. The reason is that the new services of C_1 are provided by a service provider with a gold SLA, requiring less extensive testing of services. In sum, if an adaptation decision would be made based on estimated cost only, adaptation option C_1 would be selected for adaptation.

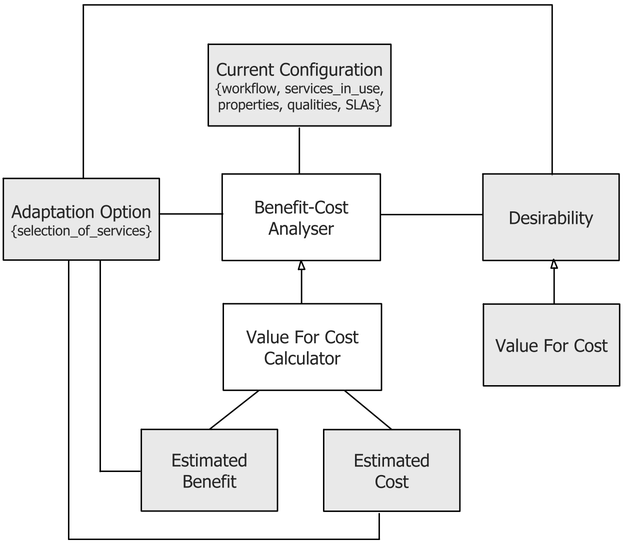


Fig. 8. Example instance of benefit-cost analysis model kind.

Benefit-Cost Analysis Model Kind (MK3). This model kind describes how the *estimated desirability* of each adaptation option is determined (see Table 2 bottom). Estimated desirability ED_{C_i} expresses the degree to which stakeholders prefer the selection of an adaptation option C_i over other options by comparing the total estimated benefit with the total estimated cost. The estimated desirability of an adaptation option increases with higher estimated benefit and lower estimated cost. The *benefit-cost analyzer* computes the desirability of an adaptation option. This computation requires that the estimated cost and benefit are expressed in a common metric and are scaled to be comparable. Different approaches exist to determine desirability. One common approach is the so-called "Value-For-Cost," which defines desirability as the ratio of the estimated overall benefit to the estimated overall cost (both scaled).[4] For example, consider two adaptation options with the same estimated benefit, say performance. The adaptation option with the lowest cost, say resources required to adapt the system, will have the highest estimated desirability of the two options. We say that for these two options with the same estimated performance, the option with the lowest estimated cost is desirable over the other option. Another approach subtracts the total estimated cost from the total estimated benefit to determine the desirability of an adaptation option. More advanced approaches may include regression and forecasting techniques to determine desirability.

Example. Figure 8 shows the benefit-cost analysis model kind instantiated for the health assistance system. We use VFC as a concrete instance to express the estimated desirability of adaptation options. The VFC calculator computes VFC for adaptation option C_i as follows:

$$VFC_{C_i} = \frac{s_b(EB_{C_i})}{s_c(EC_{C_i})}, \tag{8}$$

with s_b a function that scales estimated benefit EB_{C_i} and s_c a function that scales the estimated cost EC_{C_i}. In this example, we use trivial scaling functions that return the values of the original

---

[4] VFC was introduced in self-adaptive systems by Van Der Donckt et al. [67]. More generally, in the context of the cost-benefit analysis of architecture choices, VFC is sometimes considered as a basis to determine the Return on Investment, see for example [45].

Table 6. Generic Risk Attributes with Possible Metrics

Risk Attribute	**Risk Metrics**
Health	Fatalities, aid required
Safety	Fatalities, aid required
Security	Vulnerability, impact
Privacy	Data loss, impact
Community	Outrage, damage
Environment	Harm, damage
Financial	Loss, costs

estimates, i.e., $s_b(EB_{C_i}) = EB_{C_i}$ and $s_c(EC_{C_i}) = EB_{C_i}$. Applied to the two adaptation options C_1 and C_2, we obtain the following values for VFC.

$$VFC_{C_1} = \frac{24.5}{4} = 6.125; \quad VFC_{C_2} = \frac{29.0}{6} = 4.83. \tag{9}$$

Although adaptation option C_2 has a higher estimated benefit than adaptation option C_1, the desirability of adaptation option C_1 in terms of VFC is clearly better than adaptation option C_2. The reason is that the estimated cost associated with adapting the current configuration to the new configuration is higher for adaptation option C_2 compared to C_1. Hence, if an adaptation decision would be made using VFC based on the estimated benefit and cost of adaptation options C_1 and C_2, adaptation option C_1 would be selected for adaptation.

Risk Estimation Model Kind (MK4). This model kind describes how the risk of each adaptation option is estimated (see Table 3 left). Risk in general refers to the potential effects of uncertainties on system objectives in terms of their likelihoods and consequences (positive or negative or both) [13]. A *risk attribute* represents a particular type of risk that is implied by adapting a managed system with a given adaptation option. Different risk attributes can be associated with applying one adaptation option, such as safety, environment, finances, and so on. *Risk metrics* define measures for quantifying risk attributes. A *risk estimator* is a mechanism that enables estimating the risk attributes of the adaptation options. A variety of techniques have been established in different domains to estimate the risk. Essential to these techniques is that they capture the stakeholders' concerns and intent [30]. Example mechanisms include consequence/likelihood matrix, cause-consequence analysis, and decision-tree analysis [13]. These methods differ in the purpose of the assessment, the information that is required/available, the importance of the decision, and the time available to make a decision, among other criteria. Hence, the choice of a mechanism should be tailored to the context and requirements at hand. The *estimated risk* represents the overall expected risk of applying an adaptation option based on the estimated risk attributes and the combined risk metrics. Combining risk metrics accounts for the interactions and dependencies between risks. To support architects with identifying risk attributes for a problem at hand, we devised a primary list of high-level risk attributes with associated risk metrics, see Table 6. This table is extracted from the literature on risks, including [9, 13, 24, 27, 30, 64].

Example. Figure 9 shows an instance of the risk estimation model kind for the health assistance system with two risk attributes: risk on the confidentially of the data of patients based on exposure of data, and risk on the health of patients based on inaccurate analysis results. We illustrate risk estimation for the health assistance system using a *consequence/likelihood matrix*.

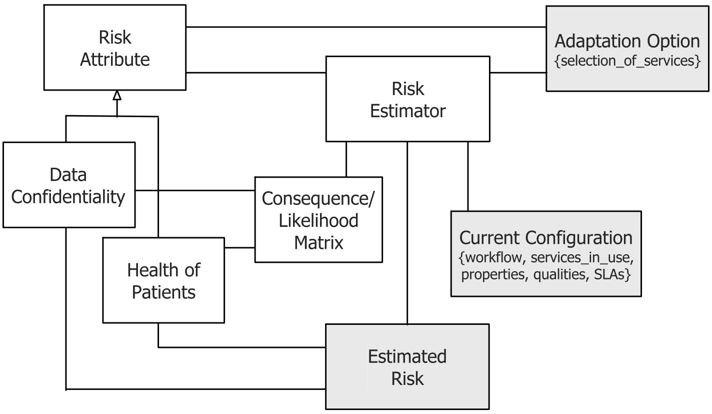


Fig. 9. Example instance of risk estimation model kind.

Table 7. Risk Metrics for Confidentially of Data of the Health Assistance System

SP Label	Data Policy	Likelihood	Consequence
Gold	Data stored local	Rarely (1/3)	Negligible effect (1)
Silver	Stored with partners	Possibly (2/3)	Limited impact (2)
Bronze	Shared with partners	Likely (3/3)	Sensitive data loss (3)
No label	Not specified	Almost certain (4/3)	Significant impact (4)

A consequence/likelihood matrix (or risk matrix) enables specifying an estimated risk according to its consequence and likelihood. Determining a risk matrix requires the involvement of the key stakeholders. The process for determining a concrete matrix for self-adaptive systems can leverage insights obtained from project management, where the process comprises four steps: (i) identify the risks, (ii) define and determine risk criteria, (iii) analyze the identified risks, and (iv) prioritize the risks and determines how to handle them.[5] Yet, the specification of a complete methodology for determining a risk matrix for a self-adaptive system requires additional research. The matrix requires customized scales for consequence (Y-axis) and likelihood (X-axis). A common approach is to use discrete scales with three to five points that can be qualitative or quantitative. The scales are determined based on the concerns of the stakeholders and their objectives. The scale for consequences can have positive or negative consequences.

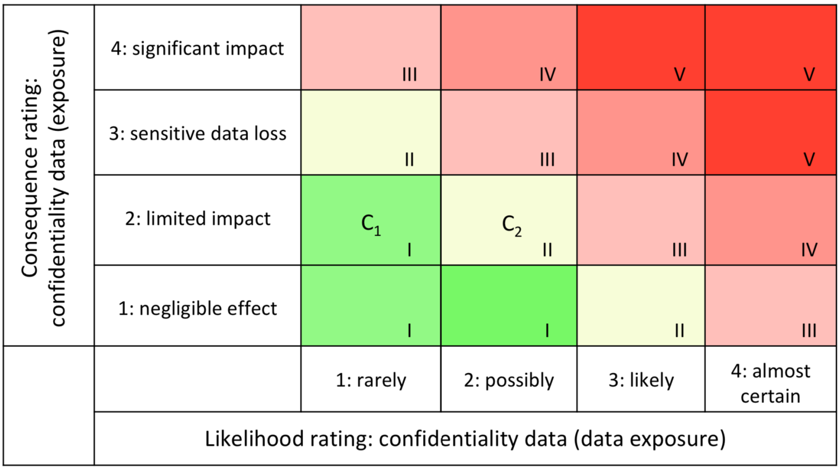
Table 7 gives an overview of the risk metrics for confidentially of data elicited from the stakeholders of the system.[6] For instance, a service provider with a gold label will store patient data locally. Exposure of data is expected to happen rarely, and if it happens, it will have a negligible effect. On the other hand, a service provider with a bronze label will share patient data with partners. Consequently, exposure of data is likely, and if it happens, it may lead to sensitive data loss.

The different options of likelihood and consequence have associated values that allow to determine risk when multiple services are combined (likelihood 1/3 to 4/3, and consequence 1–4). We apply

---

[5] See for instance: https://www.wrike.com/blog/what-is-risk-matrix/

[6] The metrics used in this example are inspired by examples provided in the Risk Standard IEC 31010:2019 [13]. Yet, this is just an example for illustration purposes, a wide variety of scales and values may apply depending on the situation at hand.



Consequence rating: confidentiality data (exposure)	4: significant impact	III	IV	V	V
	3: sensitive data loss	II	III	IV	V
	2: limited impact	C_1 I	C_2 II	III	IV
	1: negligible effect	I	I	II	III
		1: rarely	2: possibly	3: likely	4: almost certain
	Likelihood rating: confidentiality data (data exposure)				

Fig. 10. Example of a consequence/likelihood matrix for confidentially of the data.

the following rules in the example:

$$LC_{C_i} = round(LS_{SPi}^{MAS} + LS_{SPi}^{DS} + LS_{SPi}^{AS}), \tag{10}$$

$$CC_{C_i} = max(CS_{SPi}^{MAS}, CS_{SPi}^{DS}, CS_{SPi}^{AS}). \tag{11}$$

The *likelihood* of a combination of services LC_{C_i} of a configuration (adaptation option) is computed by rounding the sum of the likelihood of the individual services. On the other hand, the *consequence* of a combination of services CC_{C_i} is determined by the maximum consequence of any of the services of the configuration.

Figure 10 shows a consequence/likelihood matrix for the risk attribute confidentially of the data. In the example, the scale for consequences for the confidentiality of data in terms of "data exposure" has a 4-point scale from "negligible effect" to "significant impact." Likelihood in terms of "data exposure" also have a 4-point scale from "rarely" to "almost certain." Each cell of the matrix expresses the estimated risk at one of five possible levels, with level I corresponding to the lowest risk and level V the highest. A risk estimator will map each adaptation option to one cell of the matrix based on the risk metrics determined by the stakeholders (see Table 7) and the rules defined above. As an example, adaptation option C_2 with a medical analysis service, a drug service, and an alarm service provided by provider 1 is mapped as follows:

$$LC_{C_2} = round(LS_{SP1}^{MAS} + LS_{SP1}^{DS} + LS_{SP1}^{AS}) = round(2/3 + 2/3 + 2/3) = 2, \tag{12}$$

$$CC_{C_2} = max(CS_{SP1}^{MAS}, CS_{SP1}^{DS}, CS_{SP1}^{AS}) = max(2, 2, 2) = 2. \tag{13}$$

Value of $LC_{C_2} = 2$ corresponds to likelihood of data exposure "possibly" and value of $CC_{C_2} = 2$ corresponds to an estimated consequence of data exposure "limited impact." As a result, the estimated risk for confidentiality of adaptation option C_2 is level II, i.e., $ERL_{C_2}^{Data} = 2$. Likewise, adaptation option C_1 can be mapped to likelihood "rarely" ($round(2/3+1/3+1/3)$) and consequence "limited impact" ($max(1, 2, 2)$), with estimated risk for confidentiality C_1 level I, i.e., $ERL_{C_1}^{Data} = 1$.

In a similar way, risk metrics and a consequence/likelihood matrix can be defined for risk to the health of patients based on the accuracy of analysis results.

The overall estimated risk for each adaptation option is then determined by combining the estimated risk per risk attribute. To that end, different approaches can be applied, from basic adding or multiplying elements to providing a magnitude for a risk using a weighting factor to either the consequence or likelihood [13]. Regardless of the method used, it is important to ensure that the units are consistent and that the impact of a very high risk of one attribute should be treated properly as it may be "hidden" by a very low risk of the other attributes when combined. Therefore, adaptation options with estimated risks above certain thresholds may be ruled out before composing risk attributes.

Let us determine the overall estimated risk of adaptation options for the health assistance system using a weighted sum as an example. Assume we have a 4 × 4 consequence/likelihood matrix for the health of patients with risk levels I to V similar to the consequence/likelihood matrix for data confidentiality. Furthermore, assume that the estimated risk for the health of patients of adaptation option C_1 is level II, i.e., $ERL_{C_1}^{Health} = 2$, and the risk of adaptation option C_2 is $ERL_{C_2}^{Health} = 1$. With a weight factor $W^{Data} = 0.2$ and $W^{Health} = 0.8$, the overall estimated risk can then be determined as follows:

$$ER_{C_1} = ERL_{C_1}^{Data} * W^{Data} + ERL_{C_1}^{Health} * W^{Health} = 1 * 0.2 + 2 * 0.8 = 1.8, \tag{14}$$

$$ER_{C_2} = ERL_{C_2}^{Data} * W^{Data} + ERL_{C_2}^{Health} * W^{Health} = 2 * 0.2 + 1 * 0.8 = 1.2. \tag{15}$$

Hence, in this example, selecting adaptation option C_2 for adaptation would result in a lower estimated risk than selecting adaptation option C_1. Therefore, C_2 would be preferred over C_1 if only the estimated risk would matter in decision-making.

Decision-Making Model Kind (MK5). The fifth model kind describes how adaptation decisions are made (see Table 3 right). A *decision-making mechanism* provides a means to select an adaptation option from the set of available options taking into account the estimated desirability, and estimated risk. The *selected adaption option* C_s represents the new configuration that will be applied to the managed system to adapt it. In general, decision-making realizes the following abstract function:

$$C_s = select(C_c, \{(C_i, ED_{C_i}, ER_{C_i})\}), \tag{16}$$

$\{(C_i, ED_{C_i}, ER_{C_i})\}$ represents the set of triples of all adaptation options C_i together with their associated estimated desirability ED_{C_i} and estimated risk ER_{C_i}. The *select* function can be implemented in different ways, from a simple weighted combination of the parameters up to an integrated computation of benefit-cost-risk that may require additional or more detailed data [2]. In the following subsections, we use an illustrative example (i.e., a service-based e-health system [72]) to present different parts of our proposed architectural viewpoint in a practical setup.

Running Example—Decision-Making Model Kind. Figure 11 shows the decision-making model kind instantiated for the health assistance system. We illustrate decision-making using a mechanism that determines the best combination of services from the possible service configurations based on a weighted combination of estimated VFC and risk.

The *select* function is implemented as follows:

$$EBCR_{C_s} = \max \sum_{i=1}^{n} (ED_{C_i} * W_{VFC} - ER_{C_i} * W_R) \tag{17}$$

The adaptation option C_s that is selected for adaption is the option that maximizes the estimated benefit-cost-risk $EBCR_{C_s}$. $EBCR_{C_s}$ is computed as the difference between the weighted estimated desirability $ED_{C_i} * W_{VFC}$ and the weighted estimated risk $ER_{C_i} * W_R$.

The weights W_{VFC} and W_R determine the importance stakeholders put in the estimated desirability of the selected adaptation option, i.e., its VFC in the health assistance system, and the estimated risk. If we assume equal weights, i.e., $W_{VFC} = 0.5$ and $W_R = 0.5$, and we use the values for estimated

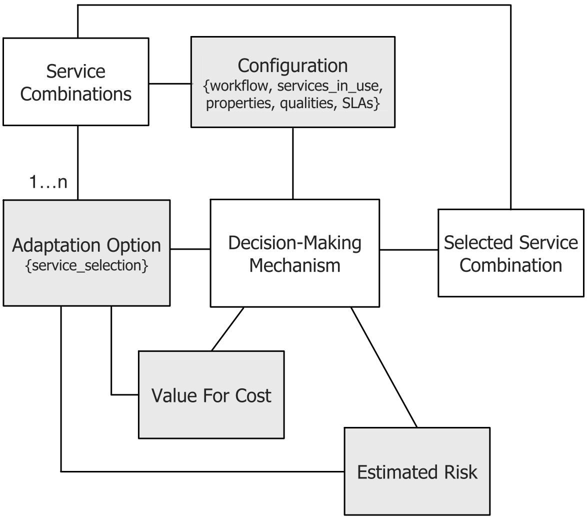


Fig. 11. Example instance of decision-making model kind.

desirability and risk for C_1 and C_2 from the previous examples, the decision will be made as follows:

$$EBCR_{C_1} = ED_{C_1} * W_{VFC} - ER_{C_1} * W_R = 6.125 * 0.5 - 1.8 * 0.5 = 2.16, \tag{18}$$

$$EBCR_{C_2} = ED_{C_2} * W_{VFC} - ER_{C_2} * W_R = 4.83 * 0.5 - 1.2 * 0.5 = 1.81. \tag{19}$$

If C_1 and C_2 would be the only two adaptation options, the decision-making mechanism would select adaptation option C_1. However, if the stakeholders would prefer different weights for desirability and risk, the outcome may be different. We elaborate on this in the next section.

Note on the Scalability of the Approach. Throughout the explanation of the model kinds, we considered only a few adaptation options to explain the different steps to make adaptation decisions based on estimated benefits, costs, and risks. This way, we aimed to keep the explanation simple and clear. Yet, this does not imply that the approach would be restricted to settings with only a few adaptation options. The complexity of the computations of the different steps to make adaptation decisions depends on the concrete techniques used in each step. As can be seen in the example case the complexity of the computations is rather low and the proposed solution would surely scale for systems with a huge number of adaptation options. Actually, previous research has shown that rule- and utility-based decision-making (at least for estimated benefits and costs) scales very well, see for instance the cases described in [22, 67].

5.5 Viewpoint Analysis

The viewpoint defines two types of analysis presented in Table 8: *benefit-cost tradeoff analysis* and *desirability-risk tradeoff analysis.*

Benefit-Cost Tradeoff Analysis. This analysis is applied to a selection of relevant adaptation scenarios, each comprising a current configuration with a selection of adaptation options. The analysis then assesses the effects of assigning different weights to the estimated benefit and estimated cost on the desirability of the adaptation options. The results can then be checked with domain knowledge obtained from stakeholders, historical information, field tests, or any other relevant data sources. The analysis results help balance the tradeoffs between estimated benefit and estimated cost when designing the benefit-cost analyzer. This analysis is usually performed at design time, but may also

Table 8. Viewpoint—Analysis

Analyses

A1—Benefit-Cost Tradeoff Analysis (Using MK3). Assesses the effects of different weights for benefit and cost on the overall estimated desirability of adaptation options of a given configuration with a given benefit-cost analysis mechanism.

A2—Desirability-Risk Tradeoff Analysis (Using MK5). Assesses the effects of different weights for desirability and risk on the selection of adaptation options of a given current configuration with a given decision-making mechanism.

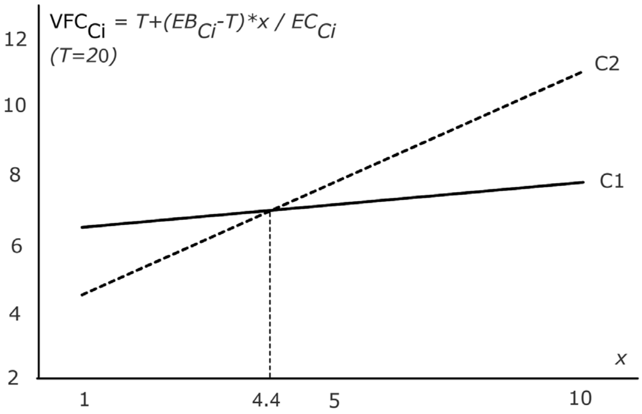


Fig. 12. Change of desirability (VFC) based on weights for estimated benefit and cost.

be useful in the context of a system evolution; for instance when new goals or risks are identified that need to be incorporated into the self-adaptive system.

Example. We illustrate the benefit-risk tradeoff analysis for the simple scenario of the health assistance system that we used to illustrate the benefit-cost model kind. In that example, we used trivial scaling functions that return the values of the original estimates for benefit and cost (i.e., $s_b(EB_{C_i}) = EB_{C_i}$ and $s_c(EC_{C_i}) = EC_{C_i}$), and we applied that to determine estimated VFC of two adaptation options (C_1 and C_2). We look now at how a parametric scaling function for benefit gives preference to adaptation options with high benefit as follows:

$$VFC_{C_i} = \frac{T + (EB_{C_i} - T) * x}{EC_{C_i}}. \tag{20}$$

The scaling function of estimated benefit is determined based on T, a threshold for estimated benefit, and a multiplier x. The concrete values for T and x need to be determined in consultation with the stakeholders as they express domain-specific preferences of the key people with an interest in the self-adaptive system. The scaling function of estimated cost remains trivial, returning the values of the original estimates for cost. The overall estimated benefit of an adaptation option is determined by taking the sum of the threshold and the fraction of the estimate above the threshold multiplied by a factor x. Figure 12 shows how estimated VFC (VFC_{C_i}) is determined for two adaptation options C_1 and C_2.

In this particular setting, threshold T is set to 20, and x is changed in a range from 1 to 10. Note that the setting with $x = 1$ corresponds to the original setting we used to illustrate the

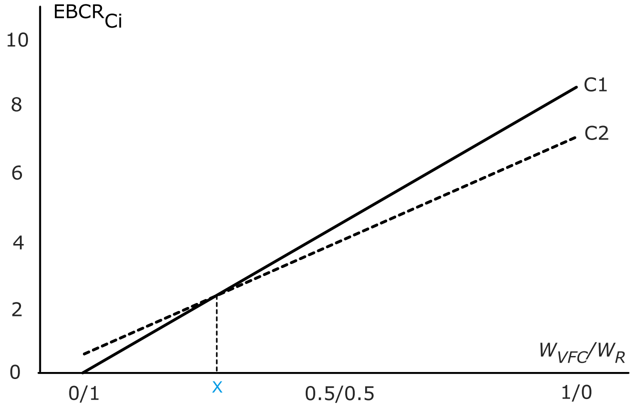


Fig. 13. Changing decision based on weights for estimated desirability (VFC) and risk.

desirability model kind ($VFC_{C_1} = 6.125$ and $VFC_{C_2} = 4.83$). As we can see, for values of x equal to or smaller than 4.44 the VFC of adaptation option C_1 would be preferred over that of C_2. For the complementary range of values, the opposite choice would be preferred. This means that above $x = 4.44$ the higher contribution of estimated benefit of C_2 above the threshold (namely 29.0 compared to 24.5 for C_1), would compensate the higher cost (6 for C_2 vs. 4 for C_1). The final choice for the threshold and multiplication factor x is something the stakeholders need to make.

Desirability-Risk Tradeoff Analysis. This analysis is applied to a selection of relevant adaptation scenarios, each comprising a configuration and a set of adaptation options. The analysis assesses the effects of assigning different weights to estimated desirability and risk on the selection of adaptation options by the decision-making mechanism. The results can then be compared with domain knowledge. The analysis results help balance estimated desirability (based on estimated benefit and cost) and estimated risk in the decision-making process. The results will help determine the knowledge required by the decision-making mechanism. Desirability-risk tradeoff analysis is usually performed at design time, but it can also be useful during system evolution.

Example. We illustrate the desirability-risk tradeoff analysis for the simple scenario of the health assistance system that we used to illustrate the decision-making model kind. In that example, we assumed equal weights, i.e., $W_{VFC} = 0.5$ and $W_R = 0.5$ to select one of two possible adaptation options (C_1 and C_2).

We look now at how a change in the weights has an effect on decision-making. Figure 13 shows how estimated desirability-risk of the two adaptation options ($EBCR_{C_1}$ and $EBCR_{C_2}$) change with different weights. For values of W_{VFC} equal or smaller than threshold x (e.g., 0.24 and hence, values of W_R approximately equal or larger than 0.76) adaptation option C_2 would be preferred over C_1. This means risk-averse stakeholders would give more attention to risk as to desirability in terms of VFC. For the complementary range of values for the weights, the opposite choice would be made. The final choice for the weights is something the stakeholders need to make.

6 Validation of the Viewpoint

The value of a viewpoint can be established if the viewpoint proves to be effective in practice [6]. To this end, it is important to select a suitable validation method to evaluate the value of the benefit-cost-risk-aware decision-making viewpoint for the self-adaptive systems community.

Table 9. Evaluation Questions (EQs)

EQ1: How do senior researchers or practitioners perceive the *usefulness* of the viewpoint in self-adaptive systems domain?
EQ2: How do senior researchers or practitioners perceive the *understandability* of the viewpoint in self-adaptive systems domain?
EQ3: How do senior researchers or practitioners perceive the *applicability* of the viewpoint in self-adaptive systems domain?

To get a better grasp of the existing validation methods, we first conducted a small-scale review to identify commonly used validation methods for architectural viewpoints in the literature. Adopting the best practices from methods extracted from the literature and our own experiences [6, 31, 42, 51, 52, 55, 68], we selected a case study as validation methodology, which we discuss in detail in the following subsections. For a summary of the results obtained from the small-scale literature review, we refer to Appendix A.

6.1 Case Study

To validate the proposed viewpoint, we conducted a case study. Runeson and Höst [61] define a case study as an empirical method that aims at investigating contemporary phenomena in their context, emphasizing using multiple sources of evidence. A case study can be used for both exploratory and descriptive purposes. In our case, a group of participants used the viewpoint to perform and document design tasks in an IoT case named DeltaIoT [40]. This case study was confirmatory, i.e., testing existing theories [26]. The case study involved participants with sufficient experience in designing and developing self-adaptive systems, meaning having at least 2 years of experience in the software industry and have a training in software design or architecture and/or a training in self-adaptive systems. After the design tasks, we conducted a survey to collect additional data on the perception of the participants on the use of the viewpoint when performing the design tasks. We used the survey data also to cross-validate the data collected from the design tasks where applicable using the Concurrent Triangulation Strategy [35]. This strategy uses different methods to confirm, cross-validate, or corroborate findings, motivated by the fact that often “what people say” could be different than “what people do” and therefore collecting data from multiple sources helps to improve the validity of results [26]. We discuss triangulation of the relevant data sources in our study in Section 6.3.

6.1.1 Goal. We use the Goal Questions Metric approach to define the goal of the case study [7]:

> Analyze *the benefit-cost-risk-aware viewpoint* for the purpose *of evaluation with respect to its usefulness, understandability, and applicability* from the point of view of *senior researchers and practitioners* in the context of *design of self-adaptive systems.*

We focused on usefulness, understandability, and applicability to assess the viewpoint, as these have shown to be most appropriate properties for validating viewpoints [6, 31, 51, 52, 68]. We refined the goal of the case study in three main evaluation questions, listed in Table 9.

6.1.2 Case Description. DeltaIoT is an IoT system consisting of a collection of 15 motes. The motes collect data from the environment using different types of sensors. This data is relayed via a wireless network in a multi-hop way to a gateway. The gateway is connected to a central monitoring facility where security personnel monitor the status of various buildings and take appropriate actions (i.e., adapt the system) to continuously support stakeholders' concerns.

The stakeholders of the DeltaIoT system have three major concerns: (1) optimizing packet loss and energy consumption when transmitting data, (2) keeping the energy consumed by the extra transmissions that are required to adapt the network settings low, and (3) considering the risk of service interruption due to jamming attacks in the network. Instead of manual intervention by the staff to deal with these concerns, in the case study, we invited participants to design a solution that added self-adaptation capabilities to the system to deal with the three main concerns. For a detailed description of the evaluation case, we refer to Appendix B.

6.1.3 Subjects. We used the convenience sampling method (i.e., samples are drawn from a population that is easily accessible) to select participants. We targeted researchers and practitioners who have concrete expertise in the design and realization of self-adaptive systems and are familiar with the topic of uncertainty in self-adaptive systems.[7] To ensure high-quality input from the subjects, we defined a set of criteria to evaluate their level of expertise. To be included in the case study, the subjects had to be researchers or practitioners with a training on self-adaptive systems, and/or software design and architecture. To that end, we asked all the participants of the study a number of background questions to ensure their suitability to participate in this study. Besides their training experiences, we also asked then their current role, and the years in the current role. Furthermore, we asked the participants how many years of experience they have in the software industry. The answers to these questions were used to filter out impertinent participants and interpret some of the results based on the level of industrial expertise of the participants.

6.1.4 Data Collection. The data used for the case study was derived from (a) a series of design tasks using document analysis, (b) the answers to questions of a survey.

Design Tasks. The tasks were defined in the context of the DeltaIoT system. The participants received an introduction of the viewpoint, a specification of the system with the stakeholder concerns and system settings, as well as a high-level architecture of the system. The task consisted of two parts:

(I) Design a decision-making component that takes into account the stakeholder concerns, i.e., the utility for packet loss and energy consumption, the value for cost due to communication overhead for adaptation, and the risk due to potential jamming attacks of the network.
(II) Starting from a given configuration of the system consider two given adaptation options. Apply now your solution for the decision-making and determine whether adaptation would be useful, and if so which of the two adaptation options would be selected for adaptation.

Survey. The survey included six demographic questions (first section) and eight main questions (second, third, and fourth sections). We used closed (Likert-scale) and open-ended questions (free text). The first section focused on the background of the participants (see the description in Section 6.1.3). These answers were used to ensure the suitability of the participants for the study. The second section focused on the usefulness of the viewpoint. The participants were asked whether or not they had used the viewpoint during the tasks. If the answer was negative, we probed what could have been the reason for their choice. If the answer was positive, they were asked to answer four secondary questions to measure the usefulness of the viewpoint from their perspective. In the third section, we posed two main questions that probed the understandability of the viewpoint

---

[7] We invited 20 people to participate in the case of whom 14 accepted the invitation. Since the evaluation is qualitative in nature, 14 participants give us a wealth of data for the analysis.

Table 10. Usefulness Measures, Definitions, and Metrics

Measures	Definitions	Metrics	Means of Quantification
Effectiveness	Degree to which users can correctly complete tasks using the viewpoint.	(1) Completeness (per task): Part of the subtasks of a task completed. (2) Correctness (per task): part of the subtasks of a task correctly completed (i.e., completeness subtracted by error parts).	(1a) Task completeness = average of parts of the subtasks of a task completed (1b) Viewpoint completeness = average of task completeness of all tasks (2a) Task effectiveness = average of parts of subtasks of a task correctly completed (2b) Viewpoint effectiveness = average of task effectiveness of all tasks
Efficiency	Ratio of parts of correctly performed tasks and the time required to complete the tasks.	Correctness: See metric (2) of effectiveness. Time: time spent on finishing a task.	(a) Task efficiency = relative percentage of task effectiveness divided by the time used for a task (b) Viewpoint efficiency = average of the task efficiency of all tasks
Satisfaction	Quality of interactions with the viewpoint's model kinds.	Scale: Answer survey questions on a 5-point Likert-scale	Level of satisfaction from 1 (not satisfied) to 5 (exceptionally satisfied)

of the participant. The last section included two main questions assessing the applicability of the viewpoint. We present the concrete survey questions in the following sections (labeled SQ1 to SQ8).

6.1.5 Metrics. We define metrics and means of quantifying the usefulness, understandability, and applicability of the viewpoint, leveraging the book of Fenton and Biemans on software metrics [29].

Usefulness of the viewpoint (EQ1) is concerned with whether the viewpoint assists designers with designing a solution that enables systematic estimating of benefits, costs, and risks for the adaptation options of the system, performing tradeoff analysis between the benefits and costs, and making adaptation decisions at runtime. Usefulness can be measured in three aspects: (1) effectiveness, (2) efficiency, and (3) satisfaction [29] as summarized in Table 10.

To measure the usefulness of the viewpoint, we collected data from the design tasks and the survey. We start with effectiveness. For design task one, we looked at the design of the decision-making component, in particular sub-components for estimating the benefit, cost, and risk, the analysis of benefit-cost, and the decision-making. For design task two, we looked at concrete estimations of benefit, cost, and risk, the analysis of benefit-cost, and the decision-making. We refer to the different parts of each task as subtasks. We measured the effectiveness as follows:

Task Completeness. Average of parts of the subtasks of a task completed (percentage):

1. Low values ($\leq 35\%$): the main components and/or explanation are presented, however most of the details (i.e., attributes, functions, relations, values) are missing.
2. Medium values ($> 35\%$ and $< 70\%$): the main components and/or explanation are presented; however, half of the details (i.e., attributes, functions, relations, values) are missing.
3. High values ($\geq 70\%$): the main components and/or explanation are presented and more than half of the details (i.e., attributes, functions, relations, values) are fully provided.

Viewpoint completeness was then computed as the average of the task completeness of all tasks.

Table 11. Understandability Measures, Definitions and Metrics

Measures	**Definitions**	**Metrics**	**Means of Quantification**
Comprehensibility	The difficulty of grasping descriptions of concerns and model kinds.	Scale: Answer to survey question on a 5-point Likert-scale.	Median of ordinal scores from 1 (i.e., incomprehensible) to 5 (i.e., trivial).
Perceived correctness	The perceived degree of correctly produced tasks using the viewpoint.	Scale: Answer to survey question on a 5-point Likert-scale.	Median of ordinal scores from 1 (i.e., incorrect) to 5 (i.e., completely correct)

Task Effectiveness. Average of parts of the subtasks of a task correctly completed (percentage):

(1) Low values ($\leq$ 35%): the main components and/or explanation are presented, however most of the details (i.e., attributes, functions, relations, values) are incorrect.
(2) Medium values (> 35% and < 70%): the main components and/or explanation are presented; however, half of the details (i.e., attributes, functions, relations, values) are incorrect.
(3) High values ($\geq$ 70%): the main components and/or explanation are presented and more than half of the details (i.e., attributes, functions, relations, values) are correct.

Viewpoint effectiveness was then computed as the average of the task effectiveness of all tasks.

Efficiency refers to the ratio sub-tasks that are correctly performed and the time required for it. The effectiveness of task t by participant PT_i is computed relative to the highest effectiveness obtained for that task by a participant PT_j. Hence, the task effectiveness of PT_j for task t is 100% and that of PT_i is a fraction of it relative to the task effectiveness of PT_j.

Finally, to measure satisfaction we used data from the survey, in particular the level of satisfaction experienced by the participants from interacting with model kinds. The values participants could select from a range of "1 = not satisfied" to "5 = exceptionally satisfied."

Understandability of the viewpoint (EQ2) refers to the extent to which the participants can understand the proposed viewpoint including the concerns and model kinds and correctly generate instances of the model kinds for a concrete case at hand. Understandability can be measured in two aspects: (1) comprehensibility, and (2) perceived correctness [29], as summarized in Table 11.

To measure the understandability of the viewpoint, we formulated two main survey questions about comprehensibility and perceived correctness. Comprehensibility refers to the degree designers understand the viewpoint in general, i.e., the concerns and the model kinds. Perceived correctness refers to the degree of correctly produced design tasks using the model kinds or the viewpoint in general. We used the answers to the questions to cross-validate the data of the design tasks and collect possible additional insights from participants.

We measured comprehensibility as the median of ordinal scores with values from "1 = trivial" to "5 = incomprehensible." For measuring the perceived correctness of the model kinds we used the median of ordinal scores of answers on a scale with "1 = incorrect" to "5 = completely correct."

Applicability of the viewpoint (EQ3) refers to the inclusivity of the viewpoint in terms of covering the relevant concerns, and its relevance for use in the future, as perceived by the participants [29].

Measuring the applicability of the viewpoint may be subjective due to several factors, such as the expertise of the participants, their role and daily duties, and their future plans for software design. Since these factors make quantitative assessment of the applicability of the viewpoint a

Table 12. Applicability Measures and Definitions Used for Qualitative Analysis

Measures	**Definitions**
Inclusivity	Refers to how participants perceive the viewpoint's aptness in thoroughly covering stakeholders' concerns.
Relevance	Refers to how participants perceive the viewpoint's relevance to use it in practice in the future.

challenging task, we opted for qualitative analysis. The measures of applicability and the definitions of inclusivity and relevance are listed in Table 12.

To assess the applicability of the viewpoint, we posed two open questions allowing us to collect personal opinions of the participants with respect to the inclusivity of the viewpoint and its relevance for use in the future respectively. The answers of the participants may include suggestions for improving the applicability of the viewpoint and its specifications.

6.1.6 Pilot. To test the clarity of the case study, the duration of the design process, as well as the understandability of the survey questions, we invited a researcher with extensive experience in software architecture, design, and systems-of-systems to participate in a pilot test. We used his feedback to revise and improve the clarity of the design tasks and the efficiency of the case study process, where applicable. The data of the pilot was not used in the data analysis of the study.

6.1.7 Data Analysis. We used descriptive statistics for usefulness and understandability, and comparative analysis method to evaluate the applicability of the viewpoint. For usefulness of the viewpoint, we graded the solutions submitted by the participants and analyzed the survey data against a predefined set of metrics. For a detailed description of the approach we used for data analysis, we refer to Appendix C.

6.1.8 Conducting the Case Study. We summarize the actions that participants undertook during the case study. We provided detailed information about the study to the participants who agreed to participate in the study via e-mail. In this e-mail, we gave an overview of the study materials, a description of the order in which the participants are advised to use them, and a link to a repository with the material. Concretely, the participants were guided to conform to the following steps:

1. *Introduction*: Participants were asked to watch a short introductory video presentation. This video included a brief introduction to the basic concepts of architectural viewpoints, a description of the viewpoint and its model kinds, and an illustrative example (i.e., a self-adaptive service-based e-health system) to illustrate the application of the viewpoint.
2. *Design tasks*: To lay the groundwork for tackling the design tasks, participants were asked to watch the second video. This video started with an introduction to the DeltaIoT system and its specifications. Then, a simple example scenario was presented and the design tasks were explained. In addition to the video, participants had access to a "Task Description" sheet that included a summary of the system specifications and an introduction to how to tackle the two tasks provided in the design tasks. To fulfill the tasks, participants were required to fill in specific "Answers Sheets" that they had to submit to us after completing the tasks.
3. *Survey*: To conclude, the participants were asked to fill in a survey. In particular, the participants were provided with a link to an online survey and were asked to take part in the survey after finishing their design tasks.

Table 13. Demographic Information of Case StudyParticipants

	Current Role	**Years in Current Role**	**Years in Software Industry**	**Training on SAS**	**Training on Software Design and Architecture**
PT1	Ph.D. researcher	3	7	Yes	Yes
PT2	Ph.D. researcher	3	5	Yes	Yes
PT3	University student	5	5	Yes	Yes
PT4	Graduate student	2	5	No	Yes
PT5	Ph.D. researcher	3	3	Yes	Yes
PT6	Ph.D. researcher	4	10	Yes	Yes
PT7	Senior lecturer	2	19	Yes	Yes
PT8	Researcher	1	3	Yes	Yes
PT9	Research associate	1	4	Yes	No
PT10	Expert cloud architect	1	7	Yes	Yes
PT11	Ph.D. researcher	1	0	No	Yes
PT12	Ph.D. researcher	3	5	Yes	Yes
PT13	Ph.D. researcher	3	2	No	Yes
PT14	Project manager	1	10	Yes	Yes

Data collected from both sources (i.e., answer sheets, and the answers to the survey questions) were used for analysis purposes. A replication package with all the materials of these three steps is publicly available.[8]

6.2 Results

We start by presenting the demographic information collected from participants. Then we zoom in on the data collected through the design tasks and the survey to answer the evaluation questions.

6.2.1 Demographics. Table 13 gives an overview of the demographic data retrieved from the survey. The first column lists the ID numbers of the participants. The data confirms that all participants comply with our predefined criteria for case study subjects (i.e., had training on SAS and/or software design and architecture). Note that 9 of the 14 participants had at least 5 years of experience in the software industry, and 13 of them had at least 2 years of experience.[9] Since we include the participants' IDs when reporting results, the results can be immediately traced back to each individual participant. This allows us to interpret results based on the participants' years of experience in industry.

6.2.2 Design Tasks. The solutions to the design tasks were analyzed to evaluate the usefulness of the viewpoint. For task one, the submissions included manually or digitally drawn diagrams and in some cases additional descriptions. For task two, the solutions were manually or digitally written computations, sometimes with additional explanations. Figure 14 shows an excerpt of a solution.

Using the predefined metrics presented in Section 6.1.7, we computed the effectiveness and efficiency of the submitted design tasks for each participant.[10] Figure 15 summarizes the results.[11]

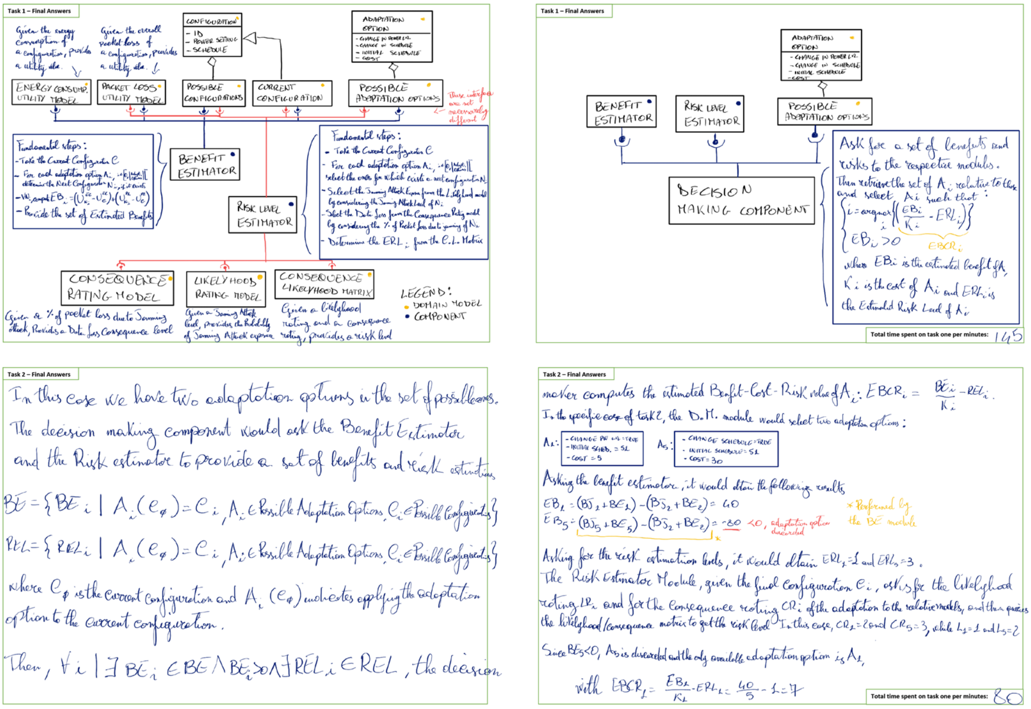
---

[8] https://zenodo.org/records/8316367

[9] The participant with identifier PT11 had no experience in the software industry. We kept this participant since she/he complied with the criteria for subjects, yet we kept an eye on the data collected from this participant to avoid an outlier.

[10] We determined the threshold values for outliers based on the mean plus/minus two times the standard deviation.

[11] We excluded three outliers for relative efficiency of task two (PT2, PT6, PT7).



In this case we have two adaptation options in the set of possibleons. The decision making component would ask the Benefit Estimator and the Risk estimator to provide a set of benefits and risk estimations

$BE = \{BE_i \mid A_i(C_\phi) = C_i, A_i \in \text{Possible Adaptation Options}, C_i \in \text{Possible Configurations}\}$

$REL = \{REL_i \mid A_i(C_\phi) = C_i, A_i \in \text{Possible Adaptation Options}, C_i \in \text{Possible Configurations}\}$

where C_ϕ is the current configuration and $A_i(C_\phi)$ indicates applying the adaptation option to the current configuration.

Then, $\forall i \mid \exists BE_i \in BE \wedge BE_i > 0 \wedge \exists REL_i \in REL$, the decision maker computes the estimated Benefit-Cost-Risk value of A_i: $EBCR_i = \frac{BE_i}{K_i} - REL_i$.

In the specific case of task2, the D.M. module would select two adaptation options:

Asking the benefit estimator, it would obtain the following results

$EB_1 = (BJ_1 + BE_1) - (BJ_2 + BE_2) = 40$

$EB_5 = \underbrace{(BJ_5 + BE_5) - (BJ_2 + BE_2)}_{*} = -80 < 0$, adaptation option discarded

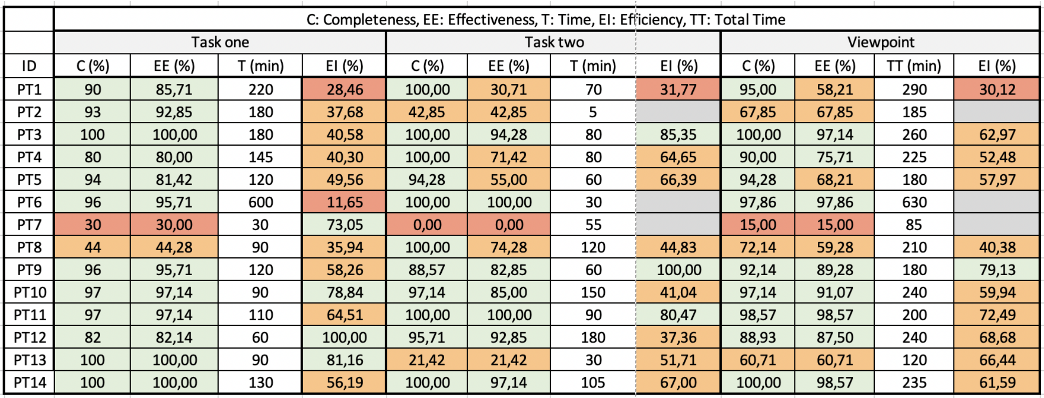
Asking for the risk estimation levels, it would obtain $ERL_1 = 1$ and $ERL_5 = 3$.

The Risk Estimator Module, given the final configuration C_i, asks for the likelihood rating LR_i and for the consequence rating CR_i of the adaptation to the relative models, and then queries the likelyhood/consequence matrix to get the risk level. In this case, $CR_1 = 2$ and $CR_5 = 3$, while $L_1 = 1$ and $L_5 = 2$

Since $BE_5 < 0$, A_5 is discarded and the only available adaptation option is A_1,

with $EBCR_1 = \frac{EB_1}{K_1} - ERL_1 = \frac{40}{5} - 1 = 7$

Fig. 14. Sample submissions for tasks one (top) and two (bottom) of case study (left: page 1, right: page 2).



	C: Completeness, EE: Effectiveness, T: Time, EI: Efficiency, TT: Total Time											
	Task one				Task two				Viewpoint			
ID	C (%)	EE (%)	T (min)	EI (%)	C (%)	EE (%)	T (min)	EI (%)	C (%)	EE (%)	TT (min)	EI (%)
PT1	90	85,71	220	28,46	100,00	30,71	70	31,77	95,00	58,21	290	30,12
PT2	93	92,85	180	37,68	42,85	42,85	5		67,85	67,85	185	
PT3	100	100,00	180	40,58	100,00	94,28	80	85,35	100,00	97,14	260	62,97
PT4	80	80,00	145	40,30	100,00	71,42	80	64,65	90,00	75,71	225	52,48
PT5	94	81,42	120	49,56	94,28	55,00	60	66,39	94,28	68,21	180	57,97
PT6	96	95,71	600	11,65	100,00	100,00	30		97,86	97,86	630	
PT7	30	30,00	30	73,05	0,00	0,00	55		15,00	15,00	85	
PT8	44	44,28	90	35,94	100,00	74,28	120	44,83	72,14	59,28	210	40,38
PT9	96	95,71	120	58,26	88,57	82,85	60	100,00	92,14	89,28	180	79,13
PT10	97	97,14	90	78,84	97,14	85,00	150	41,04	97,14	91,07	240	59,94
PT11	97	97,14	110	64,51	100,00	100,00	90	80,47	98,57	98,57	200	72,49
PT12	82	82,14	60	100,00	95,71	92,85	180	37,36	88,93	87,50	240	68,68
PT13	100	100,00	90	81,16	21,42	21,42	30	51,71	60,71	60,71	120	66,44
PT14	100	100,00	130	56,19	100,00	97,14	105	67,00	100,00	98,57	235	61,59

Fig. 15. Overview of participants' effectiveness and efficiency in design tasks (gray cells are outliers).

Key Observations from the Design Tasks on the Usefulness (Effectiveness and Efficiency) of the Viewpoint. The average completeness of the two tasks was 84% with similar results for task one and two. The average effectiveness was 76%, but the effectiveness (correctness) of the work was substantially better for task one as for task two: 84% vs. 68%. The participants spent on average 234 minutes on the design tasks. About 66% of this time was spent on task one and the rest on task two. Over the two tasks, one participant realized a high relative efficiency (PT9 with 79.13%), while one participant realized a low relative efficiency (PT1 with 30.12%). The other participants realized medium relative efficiencies.

Table 14. Survey Questions for EQ1

ID	SQs Focusing on Participants' Perception on the Usefulness of the Viewpoint
SQ1	In the example scenario, on a scale from "not effective" to "exceptionally effective," please rate the effectiveness of the viewpoint with respect to the following aspects: (a) correctly estimating benefits, costs, and risks, (b) correctly supporting the tradeoff analyses of benefits, costs, and risks, (c) correctly realizing decision-making, and (d) facilitating architects to make correct design decisions.
SQ2	In the example scenario, on a scale from "not useful" to "exceptionally useful," please rate the viewpoint's usefulness in saving time while performing the following tasks: (a) estimating the benefits, costs, and risks, (b) conducting tradeoff analysis, (c) realizing decision-making, and (d) making design decisions.
SQ3	On a scale from "not satisfied" to "exceptionally satisfied," how satisfied are you with respect to (a) the interaction with the benefit estimation model kind, (b) the interaction with the cost estimation model kind, (c) the interaction with the desirability estimation model kind, (d) the interaction with the risk estimation model kind, and (e) the interaction with the decision- making model kind.
SQ4	How do you perceive the overall impact of the viewpoint on the decision-making mechanism and its outcome? Please explain your answer.

6.2.3 Survey Results. We analyzed the survey data to evaluate the usefulness, understandability, and applicability of the viewpoint. The survey started with a general question about whether or not the participant used the viewpoint while working on the design tasks, and which specific model kinds they used. After the answer to this question, we zoom in on the analysis of the usefulness of the viewpoint, followed by the understandability, and finally the applicability.

6.2.3.1 Use of the Viewpoint. Answering the opening question of the survey, all participants reported that they used the viewpoint while working on the design tasks. The benefit estimation model kind and the risk estimation model kind were used by all participants. The cost model kind was used by 12 of the 14 participants (not used by PT4 and PT8). The decision-making model kind was not used by three participants (PT5, PT8, PT14). Finally, the benefit-cost analysis model kind was not used by seven participants (PT1, PT2, PT4, PT5, PT8, PT13, PT14). In summary, half of the participants used all five model kinds of the viewpoint. Of the other half, three participants used four model kinds, three others used three model kinds, and one participant used only two model kinds. Appendix D provides a detailed overview.

6.2.3.2 Usefulness. The survey included four questions to measure the usefulness of the viewpoint from the perspective of the participants (EQ1), see Table 14. Questions SQ1 to SQ3 collect the perceptions of the participants to effectiveness, efficiency, and satisfaction of the viewpoint, respectively. SQ4 enables the participants to share additional experiences using the viewpoint.

SQ1: Effectiveness of the Viewpoint. Question SQ1 investigated to what extent participants find the viewpoint effective with respect to performing the different steps of the design of a decision-making mechanism correctly (estimating benefits, costs, and risks, performing trade-off analyses, and realizing decision-making). Figure 16 summarizes answers to this question.[12]

---

[12] To compute averages, we used scores per participant answer of 1 = not effective ... 5 = exceptionally effective.

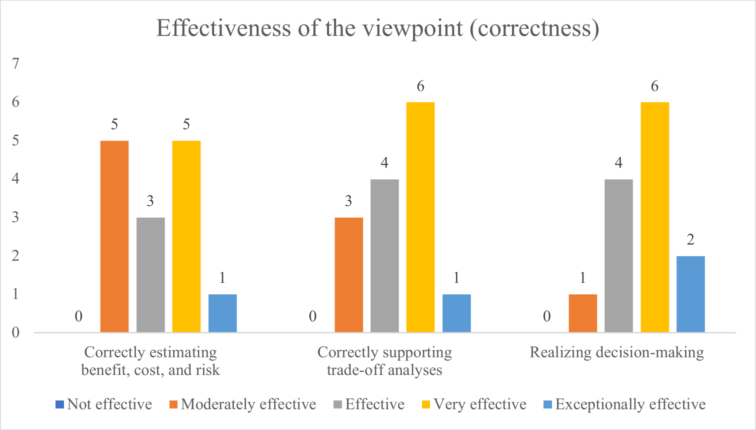


Fig. 16. Effectiveness of the viewpoint in supporting the correct design of decision-making mechanism.

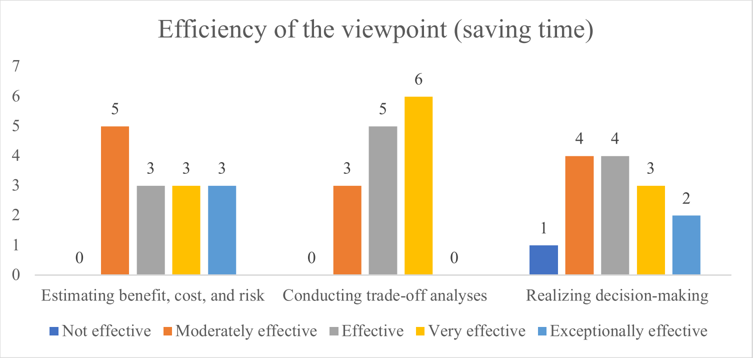


Fig. 17. Efficiency of the viewpoint in saving time when designing a decision-making mechanism.

Key Observations from SQ1. A majority of the participants found the viewpoint effective in correctly designing a decision-making mechanism, with an average score of 3.15 for estimating benefits, costs, and risks, performing 3.36 for performing trade-off analyses, and 3.43 for realizing decision-making. None of the participants found the viewpoint 'not effective' for any of the steps. With an overall average score of 3.31, we can conclude that the viewpoint is perceived as effective with respect to correctness.

SQ2: Efficiency of the Viewpoint. Question SQ2 investigated to what extent participants find the viewpoint efficient with respect to saving time when performing the different steps of the design of a decision-making mechanism (estimating benefits, costs, and risks, performing trade-off analyses, and realizing decision-making). Figure 17 summarizes answers to this question.

Key Observations from SQ2. A majority of the participants found the viewpoint efficient in saving time when designing a decision-making mechanism with an average score of 3.29 for estimating benefits, costs, and risks, 3.21 for performing trade-off analyses, and 3.07 for realizing decision-making. One participant found the viewpoint not efficient for realizing decision-making. With an overall average score of 3.19, we can conclude that the viewpoint is perceived as efficient in saving time.

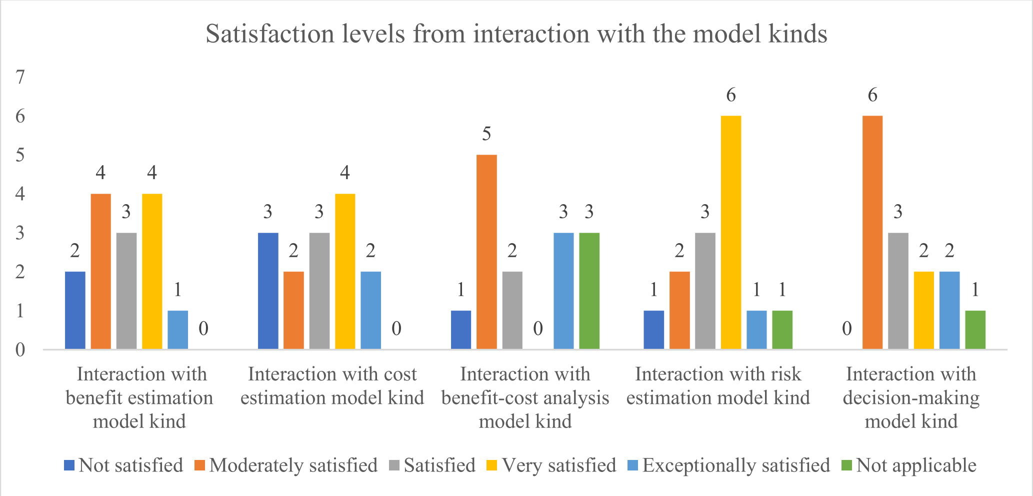


Fig. 18. Satisfaction level from interaction with the model kinds.

SQ3: Satisfaction of the Viewpoint. Question SQ3 investigated to what extent the participants were satisfied when interacting with the different model kinds of the viewpoint. Figure 18 shows a summary of participants' answers to this survey question per model kind.

Key Observations from SQ3. In general, a majority of participants were satisfied with their interactions with the model kinds of the viewpoint. Most appreciated are /the interactions with the risk model kind (average score of 3.31) followed by the decision-making model kind (3.00). Interactions with the other model kinds give mixed results (benefit model kind: 2.86, cost model kind: 2.91, and desirability model kind: 2.91). With an overall average score of 3.00, we can conclude that the participants perceived the interaction with the different model kinds of the viewpoint as satisfactory.

SQ4: Overall Impact of Viewpoint on Decision-Making. The last survey question on usefulness investigated how the viewpoint had an impact on realizing a decision-making mechanism. This open question enabled the participants to share additional experiences using the viewpoint. Table 15 gives an overview of sample quotes of the answers provided to this survey question.[13]

The answers show that the participants appreciate the viewpoint when addressing the benefits, costs, and risks in decision-making for self-adaptation: PT1 expressed the usefulness of the viewpoint as "tools to analyze and dissect decision-making"; PT2: "methodology provides an effective infrastructure"; PT3: "a general and effective abstraction"; PT4: "standardize the decision-making process"; PT5: "facilitates ... and systematically addresses the cost-benefit-risk analysis"; PT8: "good guidance"; PT9: "breaking down the assessment in multiple models"; PT11: "structured procedure to analyze"; PT13: "clearly group and summarize all the factors ... in decision-making"; PT14: "modelkinds collect all the necessary information that are needed." A few participants highlight potential improvements, in particular to further clarify the explanations and structure to use the viewpoint (PT10, PT12), and add guidance to deal with the weights of different concerns (PT14).

---

[13] The answers of two participants who did not provide clear answers to this survey question were excluded.

Table 15. Qualitative Analysis of SQ4 Illustrated with Sample Quotes

ID	Sample Quotes
PT1	"Introduces a level of confidence in the prediction and allows you to more explicitly state why some adaptation is selected." "It does give us tools to analyze and dissect the decision-making."
PT2	"The methodology provides an effective infrastructure. However, the architect has freedom in designing the decision-making policy, thus the "optimization" aspect results less convincing without an exhaustive analysis or analytical approach that does indeed provide guarantees that the solution is optimal. Maybe the combination of the viewpoint with other approaches can help on this matter."
PT3	"Using the viewpoint really helps the software architect to design the components involved in the process, providing a general and effective abstraction. It helps in reducing design time and effort and the outcome of the process can be easily determined."
PT4	"The impact is really good since the different viewpoints allow to "standardize" the decision-making process since it is reasonable to say that in a self-adaptive system we consider a benefit/cost tradeoff and perform some kind of risk analysis."
PT5	"The viewpoint facilitates the development of an architecture that systematically addresses the cost-benefit-risk analysis by providing a rigid set of model kinds. The model kinds have a positive impact on identifying factors that influence the benefit, cost, and risk of applying adaptation in a system."
PT8	"I found the viewpoint to offer a good guideline for the decision-making mechanisms; but as I have not worked a lot with it, you tend to implement your own line of reasoning which could cause you to leave it aside."
PT9	"The viewpoint is beneficial and enhances the decision-making mechanism by breaking down the assessment in multiple models. Each model focuses on a single aspect, e.g., risk, cost, etc., making it easier for practitioners to reason and identify the best solution."
PT10	"Moderate. The possible decisions are somehow framed but the explanations are not always clear enough."
PT11	"The viewpoint offers a structured procedure to analyze step-by-step each characteristic of the different available optimization options. Since its use adds consistent support in favor or against certain options when taking a decision, I perceive its impact as quite high."
PT12	"...the structure and viewpoint and interactions between the elements make sense. I would add directional arrows and follow a top-down approach for the model kinds."
PT13	"Viewpoint have the potential to clearly group and summarize all the factors that can impact in a decision-making model. I find it quite useful in implementing the model."
PT14	"The different model kinds collect all necessary data that are needed to prepare well all relevant data values on which basis a reasonable decision for adaptation or not is made. The only thing that I missed was a kind of guideline how to identify the weights for the different calculations of benefits, risks, cost, and desirability. "

Key Observations from SQ4. Overall the feedback from participants regarding the impact of the viewpoint on engineering a decision-making mechanism for a self-adaptive system was positive. Most participants highlight that the viewpoint offers systematic guidance to address the benefits, costs, and risks in realizing a decision-making mechanism.

6.2.3.3 Understandability. The second section of the survey comprised two questions to probe the understandability of the viewpoint (EQ2), see Table 16. SQ5 collected perceptions of the participants with respect to comprehensibility, and SQ6 with respect to the correctness of their solutions.

SQ5: Comprehensibility of the Viewpoint. This survey question investigated how easy or difficult it was for participants to understand the description of the viewpoint as a whole, the stakeholders' concerns, and each model kind. A summary of the responses to this question is shown in Figure 19.

Table 16. Survey Questions for EQ2

ID	SQs Focusing on Participants' Perception on the Understandability of the Viewpoint
SQ5	On a scale from "Incomprehensible" to "Trivial," please rate how easy was it for you to understand each of the following: (a) the viewpoint as a whole, (b) the description of stakeholders' concerns, (c) the description of the benefit estimation model kind, (d) the description of the cost estimation model kind, (e) the description of the desirability estimation model kind, (f) the description of the risk estimation model kind, and (g) the description of the decision-making model kind. Do you have any suggestions for improving the understandability of the viewpoint?
SQ6	On a scale from "Incorrect" to "Completely correct," how would you rate the correctness of the solutions you produced while performing the design tasks? Please use this space if you wish to add additional comments regarding the questions on the understandability of the viewpoint.

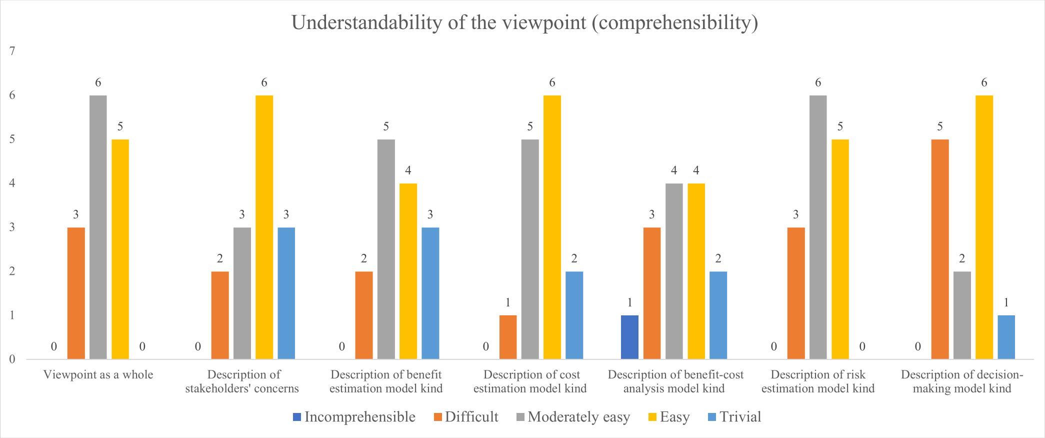


Fig. 19. Comprehensibility of the viewpoint as the easiness of understanding the viewpoint descriptions.

Key Observations from SQ5. The participants found it moderately easy to understand the viewpoint as a whole (average score of 3.14). Most easy to understand were the descriptions of the concerns (3.71) and the cost estimation model kind (3.64). Less easy were the descriptions of the model kinds of desirability estimation and decision-making (both 3.21). Only one participant found one model kind incomprehensible (the description of the desirability estimation model kind). With an overall average score of 3.38, we can conclude that the participants found it moderately easy to comprehend the viewpoint.

Only two participants provided suggestions for improving the viewpoint. The participant who found the description of the desirability estimation model kind incomprehensible suggested that a more concrete distinction between the benefit and desirability model kinds (or the relation between the two) would improve the comprehensibility of the viewpoint. Another participant suggested that adding more examples would benefit the comprehensibility of the viewpoint description.

SQ6: Correctness of Solutions. The second question on understandability investigated the confidence of the participants in the correctness of their solutions to the design tasks. On a Likert-scale of 1 = no confidence to 5 = high confidence, six participants selected a score of 4 implying that they had confidence in the correctness of their solutions, while seven participants selected a score of 3 indicating they were neither confident nor doubtful whether or not their solutions were

Table 17. Survey Questions for EQ3

ID	SQs Focusing on Participants' Perception of the Applicability of the Viewpoint
SQ7	Do you think the viewpoint thoroughly covers stakeholders' concerns in designing a Benefit-Cost-Risk-Aware self-adaptive system? (a) Yes, (b) No, (c) Maybe. Please explain your answer.
SQ8	Let us assume in the future you are taking part in a project where the use of the viewpoint may appear to be a logical option to you. Would you apply the viewpoint in your work? (a) Yes, (b) No, (c) Maybe. Which model kind(s) are more likely to be useful in your opinion? Please explain your answer.

correct. One participant selected a score of 2 showing low confidence in the solutions. For the additional comments, one participant reported that she/he was not certain about the definition of the term correctness, and another participant stated "While the concepts introduced in the case study were fairly easy to understand, I am not sure if I applied them correctly. Perhaps a more detailed description of the tasks or additional examples would have been beneficial."

Key Observations from SQ6. About half of the participants were confident about the correctness of the solutions of their design tasks, while the other half were neutral. The average score was 3.36.

6.2.3.4 *Applicability.*

The last section of the survey comprised two open questions to probe the applicability of the viewpoint (EQ3), see Table 17. The questions collected perceptions of the participants on the inclusivity of the viewpoint with respect to stakeholders' concerns (SQ7), and the perceived relevance of the viewpoint for future use in practice (SQ8).

SQ7: Covering Stakeholders' Concerns. This open question explored the perception of the participants on whether or not the viewpoint thoroughly addresses stakeholders' concerns. Table 18 gives an overview of sample quotes of the answers provided to this survey question.

The answers show that the participants consider the coverage of stakeholders' concerns of the viewpoint good. For the eight participants who answered "yes": PT1 stated it as "nicely dissects the requirements"; PT3: "covers all architectural concerns, together with the stakeholder concerns"; PT8: "guidance in all relevant dimensions"; PT9: "breaks it down it sub-parts"; PT11: "gives a quite thorough coverage"; PT12: "incorporates all information"; PT13: "There is no other concept"; PT14: "necessary steps to [...] design a component that can make the decision based on the available data". For the six participants who answered "maybe": PT2 highlighted "the architect has freedom [yet] the optimization aspect results less convincing without exhaustive analysis and analytical approach"; PT4: "reducing risk to a simple integer [...] maybe more represent a 100% degree of correctness and reality"; PT6: "not clear how utilities and functions have been defined, the interaction [...] is part of my concern"; PT10: "I think I need to implement a proof of concept."

Key Observations from SQ7. Overall, the feedback of the participants regarding the inclusivity of stakeholder concerns was positive. A majority of participants highlight that the viewpoint covers the concerns well. The participants highlighted that the viewpoint breaks the concerns into essential dimensions and guides designers to deal with them and combine them to make decisions. Feedback to potentially enhance the viewpoint relates to the clarification of utility and functions, strengthening the confidence of analysis, defining risk, the correctness of the models, and a proof of concept implementation.

Table 18. Qualitative Analysis of SQ7 Illustrated with Sample Quotes

ID	Answer	Sample Quotes
PT1	Yes	"...nicely dissect the requirements for such a system and through compartmentalization gives confidence that all aspects are accounted for in the design of system."
PT2	Maybe	"...the methodology provides an effective infrastructure." "...the architect has freedom in designing the decision-making policy, thus the "optimization" aspect results less convincing without an exhaustive analysis or analytical approach..."
PT3	Yes	"...it covers all the architect concerns, together with the stakeholders' ones. The only remaining concern should be the correctness of the given models..."
PT4	Maybe	"...maybe reducing risks to simple integer to be subtracted from a benefit/cost ratio may not represent at a 100 percent degree of correctness the reality. Also, defining the risk levels in the L/C matrix can be a tough task for stakeholders."
PT5	Maybe	"...forces me to consider all aspects related to the benefit, cost and risk of applying adaptation." "... gives an understandable description and models to take each of these aspects into account when making decisions for self-adaptive systems..."
PT6	Maybe	"It is not clear how these utilities and functions have been defined. The interaction behind making these corresponding functions is part of my concern..."
PT7	Maybe	"Not sure about it."
PT8	Yes	"...it offers guidance in all relevant dimensions of benefit-cost analysis which could help to standardize the process."
PT9	Yes	"...covers my concerns in designing a benefit-cost-risk aware SAS. It provides a thorough analysis and breaks it down in sub-parts for a better understanding." "...more examples or case studies are needed to further improve its applicability."
PT10	Maybe	"I think I would need to implement a proof of concept to be sure."
PT11	Yes	"...first time I'm approached to the Benefit-Cost-Risk evaluation in SAS and my impression is that the viewpoint gives a quite thorough coverage."
PT12	Yes	"...incorporates all information given by multiple stakeholders, it assess the benefit-cost and risk-awareness of the system, and combined them into a single answer that can be use for the system adaptation. The weights must also be provided by the stakeholders..."
PT13	Yes	"...the benefits are all the positive aspects in changing the current environment to a new configuration. The cost is actually what the system has to pay to change the configuration. The risk covers the cases in which exceptional events (in this case disasters) occur. There is no other concept..."
PT14	Yes	"It guides me through the necessary steps to think about and design mechanisms to calculate estimated benefits, cost, risk and desirability of relevant adaptation options and finally design a component that can make the decision based on the available data."

SQ8: Future Usage of the Viewpoint. The last survey question which was an open question investigated whether participants are likely to apply the viewpoint in a future project where it fits. Table 19 gives an overview of sample quotes of the answers provided to this question.

Out of 14 responses, 10 participants answered "yes" on the question of whether they would use the viewpoint in future projects if useful. The other four participants answered "maybe" indicating that they may use the viewpoint if useful. No participant answered negatively about potentially using the viewpoint. In total, seven participants stated that all model kinds would be useful (PT3, PT5, PT7, PT9, PT10, PT12, PT14). Of the other seven participants, the desirability model kind (benefit-cost analysis) was reported four times to be the most useful (PT1, PT2, PT5, PT11), followed

Table 19. Qualitative Analysis of SQ8 Illustrated with Sample Quotes

ID	Answer	Illustrative Quotes
PT1	Yes	"I think the cost-benefit together with risk estimation would enhance decision-making."
PT2	Yes	"Decision-making/benefit-cost (different customers may have different approaches to benefit/cost/risk estimation depending on their demands)."
PT3	Yes	"In my opinion all the model kinds are useful, but due to what I already expressed, the risk assessment model kind is more difficult and time consuming to define."
PT4	Maybe	"Probably the most useful ones are the benefit and the cost model kind, since the benefit/cost tradeoff is always one of the main concerns when designing engineering solutions."
PT5	Yes	"I think all model kinds are useful. To me they describe a logical view of the benefit-cost-risk analysis and encourage considering all aspects explicitly."
PT6	Maybe	"Decision-making."
PT7	Yes	"All model kinds provide useful insight, it is hard to pick one. Decision-making model is perhaps the most useful but it depends on other model kinds such as cost estimation model."
PT8	Yes	"Particularly I think that stakeholders concern, benefit estimation, and risk estimation model are the most useful because help a lot to shape the architecture of the decision-making problem and considers some of the most important aspects of it…"
PT9	Yes	"I think this depends on the applicability of the model kind(s) to the specific project. Otherwise, I would try to use all as I did in this case study."
PT10	Maybe	"All of them, probably."
PT11	Maybe	"I perceived the benefit-cost estimation model kind to be quite important in the economy of the viewpoint, with its usefulness residing in a convincing desirability analysis."
PT12	Yes	"The combination of them, although it would depends on the information available from the system."
PT13	Yes	"Cost Estimation, Benefit Estimation, Risk Estimation."
PT14	Yes	"I think all of the model kinds are useful. Every model kind covers different but relevant aspects and concerns that should be taken into account during the design of a self-adaptive system."

by the benefit, cost, and risk estimation model kinds reported three times (PT1, PT4, PT8, PT13). The decision-making model kind was reported twice as useful by participants who considered specific model kinds most useful (PT2, PT6). Motivations for using only specific model kinds included: customers' demands, importance of particular concerns, and relevance to shaping the architecture.

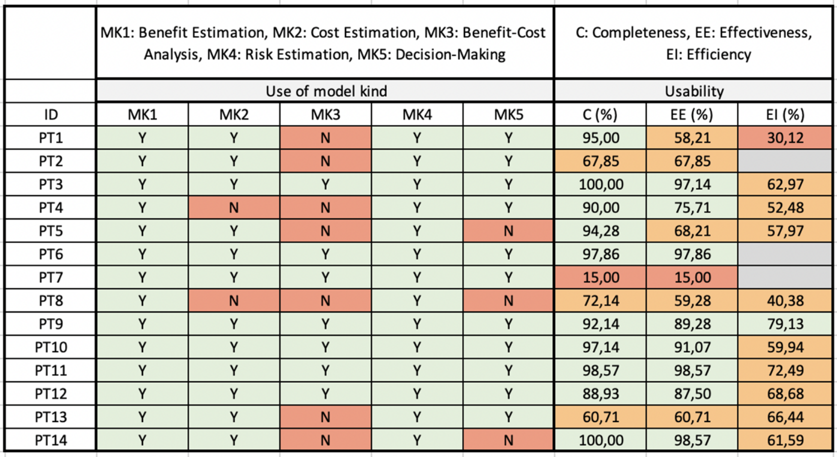
Key Observations from S8. Overall, half of the participants stated that they would use all model kinds of the viewpoint in future projects if it were an option, while the other half stated that they may use specific model kinds, dependent on the project, the requirements, and the context at hand. In conclusion, all participants would potentially apply the viewpoint in their future work.

6.3 Cross-Validation

We now cross-validate the results of the design task submissions and the survey answers. Table 20 shows details of results used for cross-validation and their respective data sources.

Table 20. Mapping of Results and Relevant Data Sources Used for Cross-Validation

Cross-Validated Results	**Data Sources**
Model kind use vs. Usefulness	Survey (opening question), Submitted design tasks.
Effectiveness vs. Perceived correctness	Submitted design tasks, Survey (SQ6).
Model kind comprehensibility vs. Usage	Survey (SQ5), Survey (opening question).



	MK1: Benefit Estimation, MK2: Cost Estimation, MK3: Benefit-Cost Analysis, MK4: Risk Estimation, MK5: Decision-Making					C: Completeness, EE: Effectiveness, EI: Efficiency		
	Use of model kind					Usability		
ID	MK1	MK2	MK3	MK4	MK5	C (%)	EE (%)	EI (%)
PT1	Y	Y	N	Y	Y	95,00	58,21	30,12
PT2	Y	Y	N	Y	Y	67,85	67,85	
PT3	Y	Y	Y	Y	Y	100,00	97,14	62,97
PT4	Y	N	N	Y	Y	90,00	75,71	52,48
PT5	Y	Y	N	Y	N	94,28	68,21	57,97
PT6	Y	Y	Y	Y	Y	97,86	97,86	
PT7	Y	Y	Y	Y	Y	15,00	15,00	
PT8	Y	N	N	Y	N	72,14	59,28	40,38
PT9	Y	Y	Y	Y	Y	92,14	89,28	79,13
PT10	Y	Y	Y	Y	Y	97,14	91,07	59,94
PT11	Y	Y	Y	Y	Y	98,57	98,57	72,49
PT12	Y	Y	Y	Y	Y	88,93	87,50	68,68
PT13	Y	Y	N	Y	Y	60,71	60,71	66,44
PT14	Y	Y	N	Y	N	100,00	98,57	61,59

Fig. 20. Use of model kinds vs. results obtained for the measures of usefulness.

We start by comparing the use of model kinds with the usefulness of the submitted design tasks. Then we look at the level of comprehensibility of the model kinds vs. their usage. Lastly, we compare the effectiveness of the submitted design tasks with the correctness perceived by the participants.

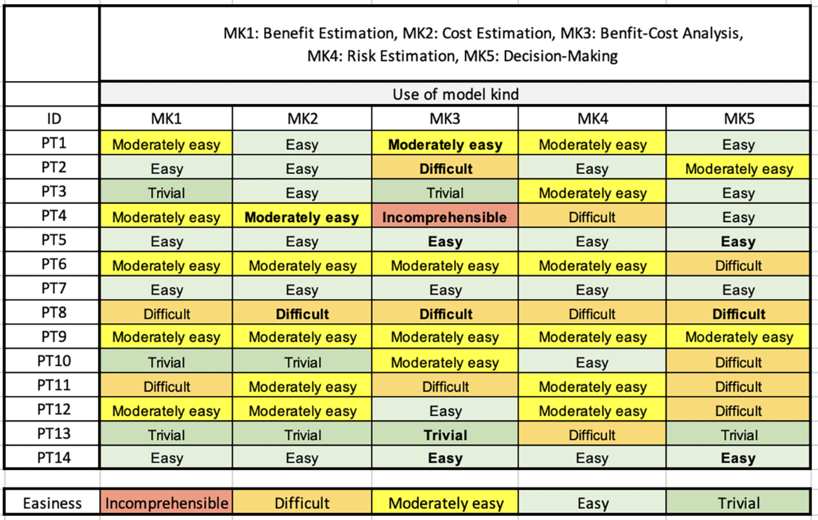
6.3.1 *Use of Model Kinds vs. Usefulness.*

We start by comparing the results obtained for the use of model kinds vs. the measures of usefulness. Figure 20 shows the cross analysis. The symbols "Y" and "N" refer to the use of the specific model kinds by each participant (yes and no respectively). For average completeness and efficiency, we observe no clear advantage when all model kinds are used (the average completeness is 84% vs. 81% when one to three model kinds are not used; the averages for efficiency are 65% and 63% respectively). On the other hand, the effectiveness (correctness) of the design tasks performed by participants who used all model kinds is substantially better compared to participants who did not use one to three model kinds (82% vs. 67%).

In conclusion, there is no clear benefit in the correctness and efficiency of the design tasks performed by participants who used all model kinds and the other participants, but, participants who used all model kinds for the design tasks were more likely to produce correct answers.

6.3.2 *Comprehensibility of Model Kinds vs. Their Usage.*

Next, we compare the level of comprehensibility of the model kinds vs. their usage. Figure 21 shows the cross analysis. Note that the corresponding model kind not used by the participant are marked in bold font. We observe that the comprehensibility of the cost estimation model kind and the benefit-cost analysis model kind are lower for participants who have not used the model kinds in their design compared to those who have used these model kinds (average score of 2.89 vs. 3.68 on a scale of incomprehensibility = 1 to trivial = 5). We do not observe this effect for the decision-making model kind (3.33 vs. 3.18).



	MK1: Benefit Estimation, MK2: Cost Estimation, MK3: Benfit-Cost Analysis, MK4: Risk Estimation, MK5: Decision-Making				
	Use of model kind				
ID	MK1	MK2	MK3	MK4	MK5
PT1	Moderately easy	Easy	**Moderately easy**	Moderately easy	Easy
PT2	Easy	Easy	**Difficult**	Easy	Moderately easy
PT3	Trivial	Easy	Trivial	Moderately easy	Easy
PT4	Moderately easy	**Moderately easy**	**Incomprehensible**	Difficult	Easy
PT5	Easy	Easy	**Easy**	Easy	**Easy**
PT6	Moderately easy	Moderately easy	Moderately easy	Moderately easy	Difficult
PT7	Easy	Easy	Easy	Easy	Easy
PT8	Difficult	**Difficult**	**Difficult**	Difficult	**Difficult**
PT9	Moderately easy	Moderately easy	Moderately easy	Moderately easy	Moderately easy
PT10	Trivial	Trivial	Moderately easy	Easy	Difficult
PT11	Difficult	Moderately easy	Difficult	Moderately easy	Difficult
PT12	Moderately easy	Moderately easy	Easy	Moderately easy	Difficult
PT13	Trivial	Trivial	**Trivial**	Difficult	Trivial
PT14	Easy	Easy	**Easy**	Easy	**Easy**
Easiness	Incomprehensible	Difficult	Moderately easy	Easy	Trivial

Fig. 21. Comprehensibility of model kinds vs. their usage.

In conclusion, the comparison of the results suggest that the participants consider the model kinds they have not used as more difficult to comprehend.

6.3.3 Effectiveness vs. Perceived Correctness. Lastly, we compare the effectiveness of the performed design tasks vs. their perceived correctness ("how correct are the submitted designs vs. how correct participants think their designs are"). Figure 22 shows the cross analysis. We observe a weak relation between the effectiveness and perceived correctness, that is, the seven participants that give a medium value of 3 on perceived correctness (a scale of 1 to 5) realized on average designs with an effectiveness of 72.34, while the six participants with a higher value of 3 on perceived correctness realize on average designs with an effectiveness of 80.47. The Kendall rank correlation coefficient between effectiveness and perceived correctness is 0.0995 with a p value of 0.71 suggesting that perceived correctness correlates only weakly with effectiveness. In conclusion, the cross-analysis indicates only a weak positive relationship between the beliefs of the participants about the correctness of their designs and the actual correctness.

7 Discussion of Results

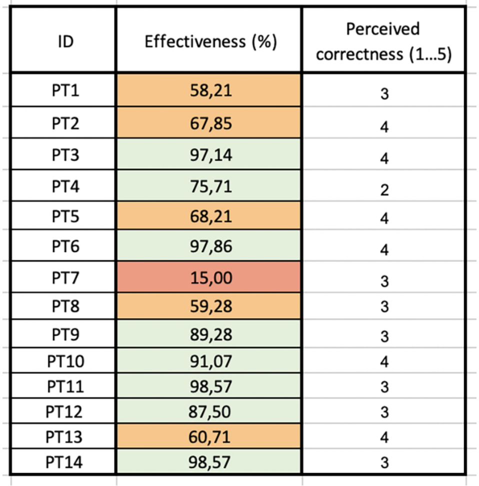
We start the discussion with a reflection on the study results and summarize key implications for researchers and practitioners. Then we discuss threats to the validity of the study.

7.1 Reflection on Results

Difference in Effectiveness between Tasks. We noted that the effectiveness (correctness) of the designs for task two was substantially lower as for task one (84% vs. 68%), while the time used for task one was substantially higher as for task two 155 min vs. 80 min). On the other hand, the average of the relative efficiency was similar for both tasks (54% vs. 61%). In absolute values, i.e., the effectiveness divided by time, the averages were also similar for both tasks, namely 74 and 84.[14] There may be different explanations for the differences in the correctness of the two tasks: task one may have been less complex than task two, or participants were less accurate when performing task two

---

[14] We refer to the replication package for detailed results.



ID	Effectiveness (%)	Perceived correctness (1…5)
PT1	58,21	3
PT2	67,85	4
PT3	97,14	4
PT4	75,71	2
PT5	68,21	4
PT6	97,86	4
PT7	15,00	3
PT8	59,28	3
PT9	89,28	3
PT10	91,07	4
PT11	98,57	3
PT12	87,50	3
PT13	60,71	4
PT14	98,57	3

Fig. 22. Effectiveness vs. perceived correctness.

because they spent already a substantial amount of time on the first task (i.e., fatigue and lack of time affected the execution of task 2).

Least "Popular" Model Kind. The benefit-cost analysis model kind was the least used model kind among the five model kinds. Four participants reported that this model kind was difficult to understand (one found it even incomprehensible). On the other hand, a majority of the participants reported a high level of satisfaction when interacting with this model kind. A possible reason for the low usage of the benefit-cost analysis model kind is that the quantifying estimated desirability (i.e., ratio of benefit to cost) in the case study was rather straightforward. As such, a number of participants may have dealt with the analysis intuitively, without guidance of the supporting model kind. Hence, the "popularity" of a model kind may be influenced by the needs for, and characteristics of, the application domain at hand. Another argument may be that the concept of desirability may not be easy to grasp by a number of participants. More validation of the viewpoint in general and the benefit-cost analysis model kind, in particular, will provide deeper insight into the usefulness. Such validation should consider other domains with different characteristics and stakeholder concerns.

Impact of Seniority and Experience of the Participants. By investigating the results of the design tasks and the survey, we noted that senior participants with experience in self-adaptation did not necessarily outperform junior participants with less experience. On the contrary, the latter group seemed to have performed better, even in the cases where participants expressed less confidence about their performance. This suggests that the viewpoint may provide a higher level of support to less experienced architects and designers. This was also mentioned by several participants in their answers to the open survey questions.

Added Value of the Risk Assessment. Enhancing traditional decision-making in self-adaptation based on estimated benefit and cost of adaptation options with estimated risk is one of the core contributions of this work. We looked into the perceptions of the participants on the added value of integrating risk in decision-making. Answers to open questions made clear that considering risks in the system and environment as first-class concern provides added value, in particular for systems with critical requirements, such as safety-critical systems. Some of the participants also highlighted that a systematic/standardized method to deal with risks as a first-class concern in self-adaptive systems supports the communication between the architects and stakeholders, and makes the logic behind the decision-making module of a self-adaptive system more transparent.

Impact of Viewpoint on the Quality of Design. The evaluation results of the usefulness of the viewpoint highlighted an interesting point: a "liberal" use of model kinds of a viewpoint (i.e., an architect decides to use a subset of model kinds based on the perceived needs) may have a negative effect on the correctness of the final design artifact. In fact, the results show that using the complete set of model kinds of the viewpoint increases the chances of higher quality of the final design artifact. This may simply be due to the fact that model kinds of a viewpoint are designed to be complementary to each other addressing related concerns. Therefore, omitting certain model kinds of the viewpoint disrupts the intentional integration between them and how they frame the different but related concerns, which may lead to more errors or slightly poorer designs. However, note that given the time constraints of practitioners, we emphasize that even using some model kinds is still much better than using an ad-hoc approach.

Appreciation of the Viewpoint. Results from the evaluation of the appropriateness and relevance of the viewpoint revealed that the viewpoint is perceived to provide an effective methodology assisting system architects to systematically identify and incorporate stakeholders' concerns with respect to benefit, cost, and risk into the design of a self-adaptive system. Providing such a viewpoint to formally address the key elements of designing a benefit-cost-risk aware decision-making component for a self-adaptive system was frequently appreciated by participants. This may be of added value for inexperienced architects, or those who are less familiar with self-adaptation.

Use of Model Kinds to Be Used in the Future. Results from the evaluation of the applicability of the viewpoint indicated that those participants who were open to using the viewpoint in the future were also more likely to try to apply all the model kinds in their work. This is a promising prospect as using the complete set of model kinds seems to have a positive impact on the quality of design artifacts, which aligns with the ultimate goal of architects and designers while opting for a viewpoint to construct and document the architecture of their system. Yet, here too, we highlight that architecting in real-life often comes with time constraints that may imply that architects can only consider the most relevant subset of model kinds when applying the viewpoint.

Application in Practice. Results from the evaluation of the relevance of the viewpoint suggested that more "real life" implementations and testing of the viewpoint are needed to strengthen the usefulness of the viewpoint in practice. To demonstrate its value, the participants who were slightly skeptical highlighted that the viewpoint may need to be applied and documented in a few systems of different domains, preferably with safety-critical demands. This would vigorously showcase the added value of the viewpoint in practice and demonstrate to practitioners the potential benefits of applying it in their work in the future. We acknowledge this as a key point for future work.

Implications. To conclude, we summarize key implications for researchers and practitioners.

Key Implications for Researchers.

- Additional research is required to obtain a deeper understanding of understanding of the concept of desirability in the decision-making of self-adaptive systems.
- Additional validation is required to provide deeper insight in the usefulness of the benefit-cost analysis model kind of the viewpoint for benefit-cost-risk aware decision-making.
- Additional applications of the viewpoint in practice will be pivotal to study the usefulness of the viewpoint in real world settings.

Key Implications for Practitioners.

- Incorporating risk as first-class concern in the decision-making of self-adaptive systems supports the communication between the architects and stakeholders.
- Using the complete set of model kinds of the viewpoint increases the chances of higher quality of the decision-making mechanism at hand. Yet, using a subset under time constraint is preferable over applying an ad-hoc approach.
- The evaluation shows that the viewpoint is particularly effective for less experienced architects and designers.

7.2 Threats to Validity

We discuss now threats to the validity of our research and strategies used to mitigate them.

7.2.1 Construct Validity. Refers to what extent the operational measures under study represent what researchers have in mind and what is investigated using the research questions [61].

The first potential threat to construct validity in validation studies is that the operational measures are not clearly and explicitly defined, and therefore, the collected data cannot be analyzed properly to answer the research questions. To tackle this threat and before conducting the case study, we created a protocol in which we explicitly explained the phenomena under study, defined the research questions addressing these phenomena, and listed a set of survey questions mapping to each research question. This ensured that relevant data was collected through interviews that can be traced back to answer the research questions in a meaningful way.

Another potential threat is that the participants may have a different understanding of the questions from what the researchers have in mind. This would invalidate the collected data to answer the research questions. To mitigate this risk, we applied two different strategies: (1) using a mix of open and closed survey questions, and (2) performing a pilot study. To prevent subjective interpretations of the survey questions, most of our questions were defined in terms of a Likert scale combined with follow-up open-ended questions. The close-ended part of the questions helped to better understand the questions by giving the participants an incentive on what a valid answer may be, while the open-ended part allowed them to provide additional explanations and examples. Another strategy to alleviate this thread is to run a pilot test. As explained above, we invited a Ph.D. student with a background in architecture and systems of systems to participate in a pilot test. The feedback received from the participant of the pilot helped us to revise and improve the procedure of the case study, the clarity of the survey questions, and any other ambiguity in the study.

7.2.2 External Validity. External validity concerns the generalizability of the case study results [61]. Since our study was limited to a small-sized sample in the context of a case in a particular domain, statistical generalization of our results is not possible. However, as suggested by [63], analytic generalization of results based on the similarities of relevant attributes between our sample and the population (e.g., experience and backgrounds of participants) is possible, hence the results are likely to hold for the general population. Nevertheless, to increase and strengthen the generalization of the study results additional validation is required in terms of participants and validation cases in other domains. We leave this as an option for future work.

7.2.3 Reliability. Reliability is concerned with minimizing errors and biases originating from the researchers involved in the study. In particular, to facilitate reliability, the data and results must be independent of the researchers who conducted the study [61]. To make sure that the study can be replicated, prior to conducting the study, we established a protocol in which we clearly explained the design of the study including the goal of the study, research questions, survey questions, and the concrete steps to conduct the study. The initial protocol was defined by one of the researchers and was then cross-checked, adjusted, and validated by the other researchers involved in the study. In addition, we crosschecked the results of the analysis and main findings by different researchers to check the correctness and validity of the results.

8 Conclusion and Future Work

Originating from the field of economics, cost-benefit tradeoff analysis has been widely used in software design and development for many years. Risk management on the other hand is also an established approach in designing and developing systems with critical requirements. Addressing benefit, cost, and risk concerns by means of architectural solutions raises a multi-faceted problem. Yet, the fluid nature of self-adaptive systems, where systems are subject to a variety of uncertainties that often can only resolved at runtime, designing a decision-making module that considers benefit, cost, and risk is a particularly challenging task for architects. The challenge is more acute in domains where the privacy or safety of users is at stake.

To tackle this challenge, we presented a reference model for benefit-cost-risk aware decision-making in self-adaptive systems that defines the primary functions that are required to select adaptation options taking into account benefit, cost, and risk. Leveraging this reference model, we then presented an architectural viewpoint that assists architects in identifying benefit, cost, and risk concerns for a system at hand and systematically designing a decision-making module for the self-adaptive system. We demonstrated the usefulness, understandability, and applicability of the viewpoint through a case study in which 14 participants performed two predefined design tasks using the viewpoint and then took part in a survey. By offering a standardized architectural approach, the viewpoint can be considered a first step towards tackling the multi-faceted problem of tackling benefit, cost, and risk and assisting architects in designing such systems.

To conclude, we outline several possible lines of future work. One of the major points of action for future work would be a real-life application of the viewpoint in a safety-critical system to offer vigorous evidence for the applicability of the viewpoint. Another action of future research could be conducting additional studies on the impact of the liberal use of model kinds vs. the complete set of model kinds. These studies should consider different domains with different characteristics (e.g., safety-critical vs. non-critical). The results of this study may shed further light not only on the impact of the liberal use of model kinds but also on the usefulness and importance of the different model kinds of the viewpoint. Realizing additional cases will offer deeper insights into the use of the viewpoint. This may then provide a basis for creating concrete guidelines for designing safety-critical self-adaptive systems where the application of adaptation requires caution with respect to imposed risks. Lastly, it would be interesting to explore the option of exploiting Generative AI techniques in realizing benefit-cost-risk-aware decision-making in self-adaptive systems; inspiration may be taken from [50].

References

[1] ISO/IEC Standard for Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems. 2017. *ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15*, 1–24. DOI: https://doi.org/10.1109/IEEESTD.2007.386501

[2] A. E. Boardman, D. Greenberg, A. Vining, and D. Weimer. 2011. *Cost-Benefit Analysis, 4th Edition.* Prentice Hall.

[3] Fatima A. Alali and Chia-Lun Yeh. 2012. Cloud computing: Overview and risk analysis. *Journal of Information Systems* 26, 2 (2012), 13–33. DOI: https://doi.org/10.2308/isys-50229

[4] Raquel Almeida and Marco Vieira. 2012. Changeloads for resilience benchmarking of self-adaptive systems: A risk-based approach. In *2012 9th European Dependable Computing Conference,* 173–184. DOI: https://doi.org/10.1109/EDCC.2012.30

[5] M. Almorsy, J. Grundy, and A. S. Ibrahim. 2013. Automated software architecture security risk analysis using formalized signatures. In *2013 35th International Conference on Software Engineering (ICSE),* 662–671. DOI: https://doi.org/10.1109/ICSE.2013.6606612

[6] Trosky B. Callo Arias, Pierre America, and Paris Avgeriou. 2009. Defining execution viewpoints for a large and complex software-intensive system. In *Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*, 1–10. DOI: https://doi.org/10.1109/WICSA.2009.5290784

[7] Victor R. Basili. 1992. *Software Modeling and Measurement: The Goal/Question/Metric Paradigm.* Technical Report CS-TR-2956. USA.

[8] Len Bass, Paul Clements, and Rick Kazman. 2021. *Software Architecture in Practice, 4th Edition.* Pearson, Addison-Wesley. Retrieved from https://people.ece.ubc.ca/matei/EECE417/BASS/index.html

[9] Paul Baybutt. 2015. Calibration of risk matrices for process safety. *Journal of Loss Prevention in the Process Industries* 38 (2015), 163–168. DOI: https://doi.org/10.1016/j.jlp.2015.09.010

[10] Carlo Bertolli, Gabriele Mencagli, and Marco Vanneschi. 2010. A cost model for autonomic reconfigurations in high-performance pervasive applications. *ACM International Conference Proceeding Series*, 3. DOI: https://doi.org/10.1145/1858367.1858370

[11] Gordon Blair, Nelly Bencomo, and Robert B. France. 2009. Models@ Run.Time. *Computer* 42, 10 (2009), 22–27. DOI: https://doi.org/10.1109/MC.2009.326

[12] B. W. Boehm. 1991. Software risk management: Principles and practices. *IEEE Software* 8, 1 (1991), 32–41. DOI: https://doi.org/10.1109/52.62930

[13] BSI. 2019. The British Standards Institution BS EN IEC 31010:2019 risk management—Risk assessment techniques. Retrieved from https://www.bsigroup.com/en-GB/standards/bs-en-iec-310102019risk-management—risk-assessment-techniques/

[14] Antoine Cailliau and Axel van Lamsweerde. 2017. Runtime monitoring and resolution of probabilistic obstacles to system goals. In *2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS),* 1–11. DOI: https://doi.org/10.1109/SEAMS.2017.5

[15] Radu Calinescu, Naif Alasmari, Colin Paterson, and Raffaela Mirandola. 2021. Quantitative verification with adaptive uncertainty reduction. *Journal of Systems and Software* 88 (2021).

[16] R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola, and G. Tamburrelli. 2011. Dynamic QoS management and optimization in service-based systems. *IEEE Transactions on Software Engineering* 37, 3 (2011), 387–409. DOI: https://doi.org/10.1109/TSE.2010.92

[17] R. Calinescu, D. Weyns, S. Gerasimou, U. Iftikhar, I. Habli, and T. Kelly. 2018. Engineering trustworthy self-adaptive software with dynamic assurance cases. *IEEE Transactions on Software Engineering* 44, 11 (2018), 1039–1069. DOI: https://doi.org/10.1109/TSE.2017.2738640

[18] Javier Cámara, Gabriel A. Moreno, David Garlan, and Bradley Schmerl. 2016. Analyzing latency-aware self-adaptation using stochastic games and simulations. *ACM Transactions on Autonomous and Adaptive Systems* 10, 4, Article 23 (Jan. 2016), 28 pages. DOI: https://doi.org/10.1145/2774222

[19] CBAM. 2018. Cost benefit analysis method, SEI, CMU. Retrieved from https://www.sei.cmu.edu/architecture/tools/evaluate/cbam.cfm

[20] Tao Chen, Rami Bahsoon, Shuo Wang, and Xin Yao. 2018. To adapt or not to adapt? Technical debt and learning driven self-adaptation for managing runtime performance. In *2018 ACM/SPEC International Conference on Performance Engineering*. ACM, New York, NY, 48–55. DOI: https://doi.org/10.1145/3184407.3184413

[21] Betty H. C. Cheng, Rogério de Lemos, Holger Giese, and Paola Inverardi. 2009. Software engineering for self-adaptive systems: A research roadmap. In *Software Engineering for Self-Adaptive Systems*. Springer, 1–26. Rerieved from https://link.springer.com/chapter/10.1007/978-3-642-02161-9_1

[22] Shang-Wen Cheng, David Garlan, and Bradley Schmerl. 2006. Architecture-based self-adaptation in the presence of multiple objectives. In *2006 International Workshop on Self-Adaptation and Self-Managing Systems (SEAMS '06).* ACM, New York, NY, 2–8. DOI: https://doi.org/10.1145/1137677.1137679

[23] Shang-Wen Cheng, David Garlan, and Bradley Schmerl. 2009. Evaluating the effectiveness of the rainbow self-adaptive system. In *2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, 132–141. DOI: https://doi.org/10.1109/SEAMS.2009.5069082

[24] Audrey J. Dorofee and Christopher J. Alberts. 2002. *Managing Information Security Risks: The Octave Approach.* Addison-Wesley.

[25] J. Cámara, W. Peng, D. Garlan, and B. Schmerl. 2018. Reasoning about sensing uncertainty and its reduction in decision-making for self-adaptation. *Science of Computer Programming* 167 (2018), 51–69. DOI: https://doi.org/10.1016/j.scico.2018.07.002

[26] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. *Selecting Empirical Methods for Software Engineering Research.* Springer London, London, 285–311. DOI: https://doi.org/10.1007/978-1-84800-044-5_11

[27] Mustafa Elmontsri. 2014. Review of the strengths and weaknesses of risk matrices. *Journal of Risk Analysis and Crisis Response* 4, 1 (2014), 49–57. DOI: https://doi.org/10.2991/jrarc.2014.4.1.6

[28] Naeem Esfahani and Sam Malek. 2013. Uncertainty in self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II.* Rogério de Lemos, Holger Giese, Hausi A. Müller, and Mary Shaw (Eds.), Springer.

[29] Norman Fenton and James Bieman. 2014. *Software Metrics: A Rigorous and Practical Approach, Third Edition.* CRC Press, Inc., USA.

[30] B. Fischhoff. 2015. The realities of risk-cost-benefit analysis. *Science* 350, 6260 (2015). DOI: https://doi.org/10.1126/science.aaa6516

[31] Matthias Galster and Paris Avgeriou. 2012. A variability viewpoint for enterprise software systems. In *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture,* 267–271. DOI: https://doi.org/10.1109/WICSA-ECSA.212.43

[32] D. Garlan. 2010. Software engineering in an uncertain world. In *FSE/SDP Workshop on Future of Software Engineering Research.* ACM, 125–128.

[33] David Garlan, Shang-Wen Cheng, An-Cheng Huang, Bradley Schmerl, and Peter Steenkiste. 2004. Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer* 37, 10 (2004), 46–54.

[34] T. J. Glazier, Bradley Schmerl, Javier Camara, and David Garlan. 2017. Utility theory for self-adaptive systems. In *CMU-ISR-17-119*.

[35] Jennifer Greene and Charles McClintock. 1985. Triangulation in evaluation: Design and analysis issues. *Evaluation Review* 9, 5 (1985), 523–545. DOI: https://doi.org/10.1177/0193841X8500900501

[36] S. Hezavehi, D. Weyns, P. Avgeriou, R. Calinescu, R. Mirandola, and D. Perez-Palacin. 2021. Uncertainty in self-adaptive systems: A research community perspective. *ACM Transactions on Autonomous and Adaptive Systems* 15, 4, Article 10 (2021), 36 pages. DOI: https://doi.org/10.1145/3487921

[37] R. Hilliard, P. Avgeriou, and P. Lago. 2010. Guest editors' introduction: Software architecture: Framing stakeholders' concerns. *IEEE Software* 27, 06 (Nov. 2010), 20–24. DOI: https://doi.org/10.1109/MS.2010.142

[38] Christine Hofmeister, Philippe Kruchten, Robert L. Nord, Henk Obbink, Alexander Ran, and Pierre America. 2007. A general model of software architecture design derived from five industrial approaches. *Journal of Systems and Software* 80, 1 (2007), 106–126. DOI: https://doi.org/10.1016/j.jss.2006.05.024

[39] Tobias Holstein, Gordana Dodig-Crnkovic, and Patrizio Pelliccione. 2018. Ethical and social aspects of self-driving cars. arXiv:1802.04103. Retrieved from https://arxiv.org/abs/1802.04103

[40] M. Usman Iftikhar, Gowri Sankar Ramachandran, Pablo Bollansée, Danny weyns, and Danny Hughes. 2017. DeltaIoT: A real world exemplar for self-adaptive Internet of things (artifact). *Dagstuhl Artifacts Series* 3, 1 (2017), 04:1–04:2. DOI: https://doi.org/10.4230/DARTS.3.1.4

[41] M. Usman Iftikhar and Danny Weyns. 2014. ActivFORMS: Active formal models for self-adaptation. In *9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems.* ACM, 125–134. DOI: https://doi.org/10.1145/2593929.2593944

[42] Didac Gil De La Iglesia and Danny Weyns. 2015. MAPE-K formal templates to rigorously design behaviors for self-adaptive systems. *ACM Transactions on Autonomous and Adaptive Systems* 10, 3, Article 15 (Sep. 2015), 31 pages. DOI: https://doi.org/10.1145/2724719

[43] ISO/IEC/IEEE 42010. 2011. *Systems and Softw. Engineering—Architecture Descr.*

[44] G. Jung, M. A. Hiltunen, K. R. Joshi, R. D. Schlichting, and C. Pu. 2010. Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In *2010 IEEE 30th International Conference on Distributed Computing Systems*, 62–73. DOI: https://doi.org/10.1109/ICDCS.2010.88

[45] Rick Kazman, Jai Asundi, and Mark Klein. 2002. Making architecture design decisions: An economic approach. Technical Report CMU/SEI-2002-TR-035.

[46] J. O. Kephart and D. M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50. DOI: https://doi.org/10.1109/MC.2003.1160055

[47] Jeffrey O. Kephart and David M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50.

[48] Narges Khakpour, Charilaos Skandylas, Goran Saman Nariman, and Danny Weyns. 2019. Towards secure architecture-based adaptations. In *14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '19).* IEEE Press, 114–125. DOI: https://doi.org/10.1109/SEAMS.2019.00023

[49] Jeff Kramer and Jeff Magee. 2007. Self-managed systems: An architectural challenge. In *Future of Software Engineering (FOSE '07).* IEEE Computer Society, 259–268.

[50] Jialong Li, Mingyue Zhang, Nianyu Li, Danny Weyns, Zhi Jin, and Kenji Tei. 2024. Generative AI for self-adaptive systems: State of the art and research roadmap. *ACM Transactions on Autonomous and Adaptive Systems* 19, 3, Article 13 (Sep. 2024), 60 pages. DOI: https://doi.org/10.1145/3686803

[51] Zengyang Li, Peng Liang, and Paris Avgeriou. 2015. *Architecture Viewpoints for Documenting Architectural Technical Debt,* 85–132. DOI: https://doi.org/10.1016/B978-0-12-802301-3.00005-3

[52] Zhendong Ma, Aleksandar Hudic, Abdelkader Shaaban, and Sandor Plosz. 2017. Security viewpoint in a reference architecture model for cyber-physical production systems. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW),* 153–159. DOI: https://doi.org/10.1109/EuroSPW.2017.65

[53] S. Mahdavi-Hezavehi, P. Avgeriou, and D. Weyns. 2017. A classification framework of uncertainty in architecture-based self-adaptive systems with multiple quality requirements. In *Managing Trade-Offs in Adaptable Software Architectures*, 45–77. DOI: https://doi.org/10.1016/B978-0-12-802855-1.00003-4

[54] G. Moreno, J. Cámara, D. Garlan, and B. Schmerl. 2015. Proactive self-adaptation under uncertainty: A probabilistic model checking approach. In *10th Joint Meeting on Foundations of Software Engineering*. ACM, 1–12. DOI: https://doi.org/10.1145/2786805.2786853

[55] Angelika Musil, Juergen Musil, Danny Weyns, and Stefan Biffl. 2019. Continuous adaptation management in collective intelligence systems. In *Software Architecture.* Tomas Bures, Laurence Duchien, and Paola Inverardi (Eds.), Springer International Publishing, Cham, 109–125.

[56] Angelika Musil, Juergen Musil, Danny Weyns, Tomas Bures, Henry Muccini, and Mohammad Sharaf. 2017. Patterns for self-adaptation in cyber-physical systems. In *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*. Springer, 331–368.

[57] Peyman Oreizy, Michael M. Gorlick, Richard N. Taylor, Dennis Heimhigner, Gregory Johnson, Nenad Medvidovic, Alex Quilici, David S. Rosenblum, and Alexander L. Wolf. 1999. An architecture-based approach to self-adaptive software. *IEEE Intelligent Systems* 14, 3 (1999), 54–62.

[58] D. Perez-Palacin and R. Mirandola. 2014. Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation. In *International Conference on Performance Engineering,* 3–14.

[59] Andres J. Ramirez and Betty H. C. Cheng. 2011. Automatic derivation of utility functions for monitoring software requirements. In *14th International Conference on Model Driven Engineering Languages and Systems* (MODELS '11). Springer-Verlag, Berlin, 501–516.

[60] A. J. Ramirez, A. C. Jensen, and B. H. C. Cheng. 2012. A taxonomy of uncertainty for dynamically adaptive systems. In *7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems,* 99—108.

[61] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14 (Apr. 2009), 2, 131–164. DOI: https://doi.org/10.1007/s10664-008-9102-8

[62] Purandare S., N. Al Islam, and M. Cohen. 2023. Self-adaptive mechanisms for misconfigurations in small uncrewed aerial systems. In *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Press.

[63] Peter Seddon and Rens Scheepers. 2012. Towards the improved treatment of generalization of knowledge claims in IS research: Drawing general conclusions from samples. *European Journal of Information Systems* 21 (Apr. 2012). DOI: https://doi.org/10.1057/ejis.2011.9

[64] Laurens Sion, Koen Yskout, Dimitri Van Landuyt, and Wouter Joosen. 2018. Risk-based design security analysis. In *1st International Workshop on Security Awareness from Design to Deployment*. ACM, 11–18. DOI: https://doi.org/10.1145/3194707.3194710

[65] Javier Troya, Nathalie Moreno, Manuel F. Bertoa, and Antonio Vallecillo. 2021. Uncertainty representation in software models: A survey. *Software and Systems Modeling* 20 (2021), 1183–1213. DOI: https://doi.org/10.1007/s10270-020-00842-1

[66] Jeroen Van Der Donckt, Danny Weyns, M. Usman Iftikhar, and Sarpreet Singh Buttar. 2019. Effective decision making in self-adaptive systems using cost-benefit analysis at runtime and online learning of adaptation spaces. In *Evaluation of Novel Approaches to Software Engineering.* Ernesto Damiani, George Spanoudakis, and Leszek A. Maciaszek (Eds.), Springer International Publishing, Cham, 373–403.

[67] M. Jeroen Van Der Donckt, Danny Weyns, M. Usman Iftikhar, and Ritesh Kumar Singh. 2018. Cost-benefit analysis at runtime for self-adaptive systems applied to an Internet of things application. In *13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE '18).* SciTePress - Science and Technology Publications, Lda, Setubal, PRT, 478–490. DOI: https://doi.org/10.5220/0006815404780490

[68] Uwe van Heesch, Paris Avgeriou, and Rich Hilliard. 2012. Forces on architecture decisions—A viewpoint. In *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture,* 101–110. DOI: https://doi.org/10.1109/WICSA-ECSA.212.18

[69] Norha M. Villegas, Gabriel Tamura, Hausi A. Müller, Laurence Duchien, and Rubby Casallas. 2010. DYNAMICO: A reference model for governing control objectives and context relevance in self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II - International Seminar.* Rogério de Lemos, Holger Giese, Hausi A. Müller, and Mary Shaw (Eds.), Springer, 265–293. DOI: https://doi.org/10.1007/978-3-642-35813-5_11

[70] Danny Weyns. 2019. Software engineering of self-adaptive systems. In *Handbook of Software Engineering*. Springer, Cham, 399–443.

[71] Danny Weyns. 2020. Introduction to self-adaptive systems: A contemporary software engineering perspective. Wiley. ISBN 978-1-119-57494-1.

[72] D. Weyns and R. Calinescu. 2015. Tele assistance: A self-adaptive service-based system exemplar. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems,* 88–92.

[73] D. Weyns, I. Gerostathopoulos, N. Abbas, J. Andersson, S. Biffl, P. Brada, T. Bures, A. Di Salle, M. Galster, P. Lago, et al. 2023. Self-adaptation in industry: A survey. *ACM Transactions on Autonomous and Adaptive Systems* 18, 2, Article 5 (2023), 44 pages. DOI: https://doi.org/10.1145/3589227

[74] D. Weyns and U. Iftikhar. 2022. ActivFORMS: A formally founded model-based approach to engineer self-adaptive systems. *ACM Transactions on Software Engineering and Methodology* 32, 1 (2022), Article No. 12, 1–48 pages.

[75] D. Weyns, U. Iftikhar, D. Hughes, and N. Matthys. 2018. Applying architecture-based adaptation to automate the management of Internet-of-things. In *12th European Conference on Software Architecture*. Springer, 49–67.

[76] D. Weyns, U. Iftikhar, S. Malek, and J. Andersson. 2012. Claims and supporting evidence for self-adaptive systems: A literature study. In *7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems,* 89–98. DOI: https://doi.org/10.1109/SEAMS.2012.6224395

[77] Danny Weyns, Sam Malek, and Jesper Andersson. 2012. FORMS: Unifying reference model for formal specification of distributed self-adaptive systems. *ACM Transactions on Autonomous and Adaptive Systems* 7, 1 (2012), 8:1–8:61.

[78] D. Weyns, M. Usman Iftikhar, and J. Söderlund. 2013. Do external feedback loops improve the design of self-adaptive systems? A controlled experiment. In *2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS),* 3–12. DOI: https://doi.org/10.1109/SEAMS.2013.6595487

[79] R. Wieringa. 2010. Design science methodology: Principles and practice. In *32nd ACM/IEEE International Conference on Software Engineering (ICSE'10)*. ACM, New York, NY, 493–494. DOI: https://doi.org/10.1145/1810295.1810446

[80] R. J. Wieringa. 2014. *Design science Methodology for Information Systems and Software Engineering*. Springer.

[81] E. Yuan, S. Malek, B. Schmerl, D. Garlan, and J. Gennari. 2013. Architecture-based self-protecting software systems. In *9th International ACM Sigsoft Conference on Quality of Software Architectures.* ACM, 33–42. DOI: https://doi.org/10.1145/2465478.2465479

Appendices

A Summary Validation Methods Viewpoints in Existing Studies

Table A1. Validation Methods of the Viewpoints in Existing Studies

Evaluation Design	**Papers' ID**
Single case study	[31, 51, 55]
Multiple case study	[42, 52, 68]
Set Up	
Industry	[6, 31, 51, 55, 68]
Academia	[42, 52, 68]
Participants	
Students	[42, 68]
Professionals	[6, 31, 51, 55]
Number of Participants	
< 10	[51, 55]
> 10	[6, 68]
Evaluation Metrics	
Usefulness	[42, 51, 55, 68]
Applicability	[6, 31, 52, 55]
Understandability	[51, 55]
Reusability	[6, 42]
Effort	[51]
Feasibility	[52]

(Continued)

Table A1. Continued

Evaluation Design	**Papers' ID**
Data Collection Sources	
Interview	[6, 31, 42, 51, 55]
Focus group	[68]
Questionnaire/survey	[6, 55]
Participact observation	[55, 68]
Assignment artifacts	[42, 55, 68]
Metrics Measurement Method	
Likert Scale	[51, 55]
Correctness of solutions per model kind	[55]
Identification of categories from results and matching them to RQs	[68]
Use of template in design cases crossed checked with feedback and interviews	[42]

B Evaluation Case

Consider an IoT application that consists of battery-powered motes deployed in the environment. The motes that are strategically placed over a geographical area as schematically show in Figure B1. The motes are equipped with sensors to provide access control to certain buildings (RFID sensor), to monitor motion in buildings and other areas (passive infrared sensor), and to sense the temperature. The sensor data of all the motes is relayed via a wireless network to the fifteenth node, which is an IoT gateway that is deployed at a central monitoring facility. Security staff can monitor the status of buildings from the monitoring facility and take appropriate action whenever unusual behavior is detected.

The communication in the network is time synchronized, meaning that the data is sent in cycles, each cycle comprising a fixed number of communication slots where a sender mote and a receiver mote can communicate with one another. A first part of the cycle contains slots allocated to the motes to communicate their sensor date to the gateway (i.e., downstream); a second part is allocated to the gateway to communicate messages to the motes (upstream). A time synchronization schedule defines a particular ordering in which data can be communicated between motes in the network.

Motes generate data based on their sensors and the conditions in the environment. For instance, motes that measure temperature will periodically communicate data samples, while motes that measure the presence of people will only communicate data when new relevant data is available. To transmit the data, motes use particular power settings. Higher power settings produce stronger signals, reducing the change that messages get lost during communication. For motes that can communicate their data via multiple motes towards the gateway, the fraction of messages sent along each path can be set as well. This way, the flow of traffic through the network can be managed.

System Configurations. The settings of the IoT network can be dynamically adapted. To that end, the IoT network offers a management interface that allows changing the network settings. This interface is accessible via the gateway. We consider here a simplified version of the IoT network, where the management interface enables the following possible settings of the network:

- The current power settings of the motes. We consider three possible power settings for the whole network: low, medium, high;

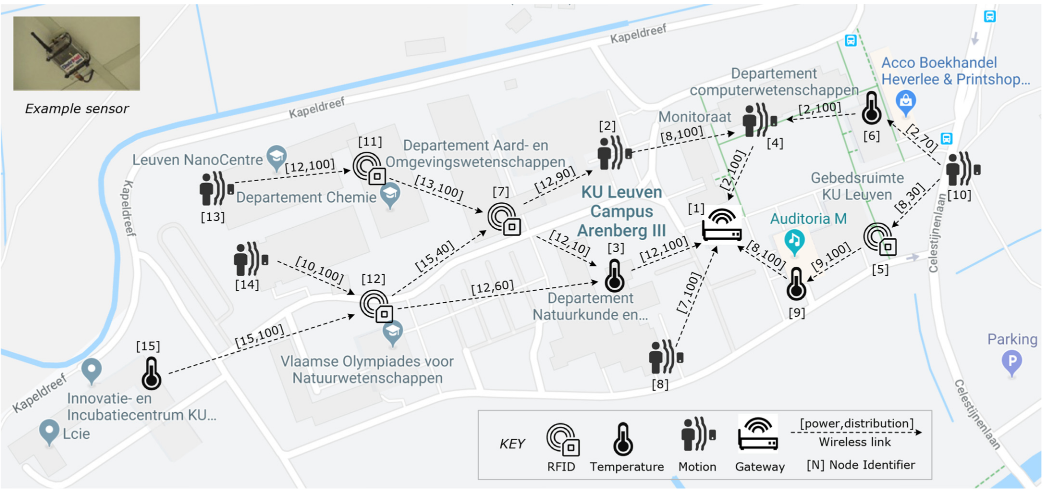


Fig. B1. Schematic overview of the IoT application.

Table B1. Possible Configurations of the IoT Network

Configuration	Power Setting Network	Schedule Network
C_1	Low	S_1
C_2	Medium	S_1
C_3	High	S_1
C_4	Low	S_2
C_5	Medium	S_2
C_6	High	S_2

- The current time synchronization schedule. We consider two possible schedules for the network: S_1 and S_2.

Consequently, in this simplified network, there are six possible configurations, as show in Table B1.

Uncertainties. The IoT application is expected to remain fully operational for a long time on a set of batteries (multiple years), while offering reliable communication. To guarantee these quality properties, the settings of the motes need to be optimally configured. Yet, guaranteeing the required qualities is complex as the system is subject to uncertainties. Here, we consider two uncertainties:

(1) *Noise:* Due to external factors, such as weather conditions, the quality of the communication between motes may be affected, which in turn may lead to packet loss;
(2) *Jamming attacks:* Jamming attacks are purposeful attempts to interfere the transmission of the regular data in the network, which may undermine the service provided by the system.

Network Data. The management interface enables to collect the following data of the network:

- The current configuration of the network, including the power settings of the motes and the current time synchronization schedule;
- The recent energy consumed by the motes of the network;

Table B2. Expected Energy Consumption and Packet Loss for Different Configurations

Configuration	Energy (mC)	Packet Loss Noise (%)			Packet Loss Jamming (%)		
		Low	Medium	High	Low	Medium	Jamming
C_1	40	2	4	6	3	6	9
C_2	80	1	2	3	2	4	6
C_3	120	0	0	1	1	2	3
C_4	30	3	6	8	4	8	12
C_5	60	2	3	4	3	6	8
C_6	90	1	1	2	2	3	4

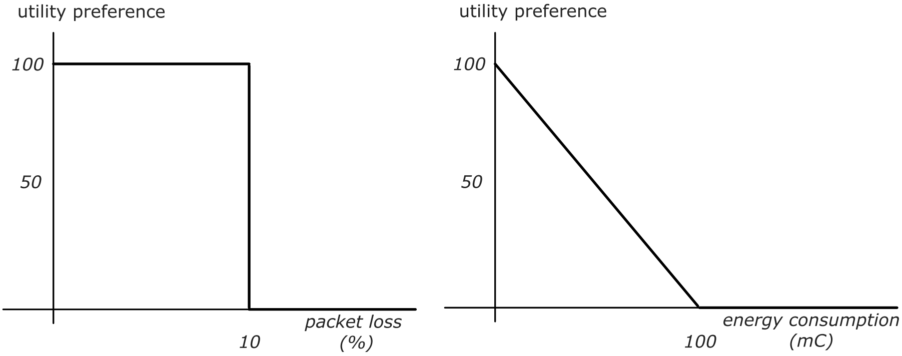


Fig. B2. Utility response curves for IoT system.

- —The recent packet loss of the network implied by noise;
- —The recent packet loss of the network implied by potential jamming attacks.

We assume here that the system is equipped with infrastructure that can distinguish packet loss generated by regular noise in the environment and packet loss generated by potential jamming.

Table B2 gives an overview of the expected energy consumption and packet loss for each configuration under different levels of uncertainty. Note that these are expected values based on stakeholder input; the actual values at runtime may differ from those in the table.

Stakeholder Requirements. The stakeholders want to optimize two quality properties of the network: packet loss and energy consumption. To that end, they defined utility response curves that are shown in Figure B2.

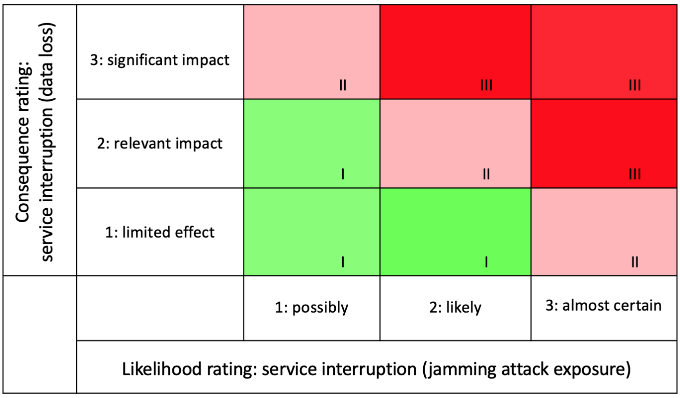
The graph on the left shows that the utility preference for configurations with packet loss below 10% is 100%, while any value above 10% has a preference of 0%. The graph on the right shows that the utility preference for configurations regarding energy consumption of the network decreases linearly from 100% to 0% for energy consumption up to 100mC. The stakeholders want to optimize the overall utility of the system at any time based on the utility response curves of the two qualities with a double weight for benefit compared to cost.

In addition, the stakeholders require that the cost for adapting the network should be taken into account such that the highest value for cost is achieved. Table B3 shows the costs for adapting the network settings.

Finally, the stakeholders want the system to take into account the risk of service interruption due to jamming attacks when adaptation is applied. Concretely, the stakeholders want the system to allocate 30% of the weight to risk compared to 70% for cost for value when adaptation is applied.

Table B3. Cost for Adapting the Network Settings

Change of Setting	Cost (mC)
Change power with schedule S_1	5
Change power with schedule S_2	10
Switch schedule $S_1 \longrightarrow S_2$	30
Switch schedule $S_2 \longrightarrow S_1$	15



Consequence rating: service interruption (data loss)

	1: possibly	2: likely	3: almost certain
3: significant impact	II	III	III
2: relevant impact	I	II	III
1: limited effect	I	I	II

Likelihood rating: service interruption (jamming attack exposure)

Fig. B3. Consequence/likelihood matrix for service interruption.

Table B4. 3-Point Scale of Likelihood Rating for Service Interruption

Jamming Attack Exposure	Jamming Attack Level
1. Possibly	Low
2. Likely	Medium
3. Almost certain	High

Table B5. 3-Point Scale of Consequence Rating for Service Interruption

Data Loss	% Packet Loss due to Jamming Attack
1. Limited effect	$0 \leq 3$
2. Relevant impact	$4 \leq 6$
3. Significant impact	> 6

To that end, the stakeholders defined a consequence/likelihood matrix shown in Figure B3. that maps configurations to risk levels (I to III). Table B4 shows how the 3-point scale for likelihood rating is defined, while Table B5 defines the 3-point scale for consequence rating.

Each cell in the matrix expresses the estimated risk at one of three possible levels, with level I being the lowest risk and level III the highest. As an example, configuration C_2 with a power setting *medium* and time synchronization schedule S_1 (see Table B1) that is potentially subject to a jamming attack at medium level creating a 4% packet loss caused by a potential jamming attack (see Table B2) will have a jamming attack exposure of “2. likely” and a data loss with “2. relevant impact,” corresponding to an estimated risk of level II (the cell in the center of Figure B3).

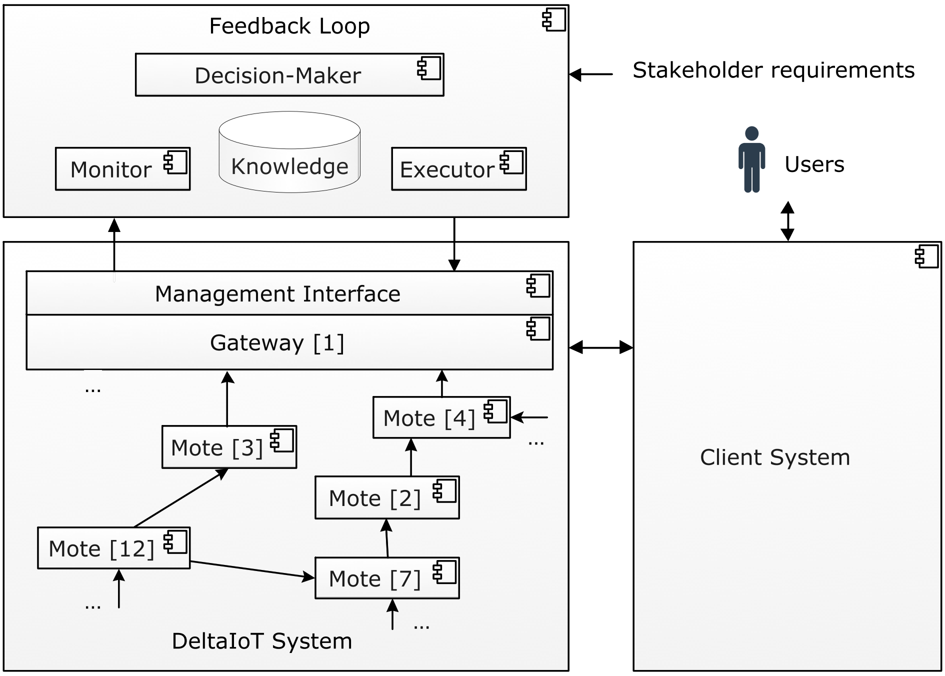


Fig. B4. Architecture of the IoT system.

Self-Adaptation. To realize the different stakeholder requirements, system operators can use the management interface to track the network parameters and adjust the network settings. However, such manual intervention is a costly, slow, and error-prone activity. By enhancing the IoT network with self-adaptation capabilities, the system will automatically track the uncertainties at runtime and use up-to-date information to find and adapt the settings of the motes such that it satisfies the stakeholder requirements.

Task. Figure B4 shows the architecture of the IoT system with a feedback loop that realizes self-adaptation. The client system provides the front end of the application to the users. Our focus is here on the design of the *Decision-Maker* component.

The task consists of two parts:

Part I Design a decision-making component that takes into account the utility for packet loss and energy consumption, the value for cost, and the risk due to potential jamming attacks, when adapting the network.

Part II Assume that the current configuration of the system is C_2 and consider two possible adaptation options C_1 and C_5 with estimated data provided above. Apply now your solution for the decision-making and determine whether adaptation would be useful, and if so which of the two adaptation options would be selected for adaptation.

For Part I, we expect a design at the level of sub-components of the decision-maker. Examples of possible data components are a representation of the current configuration, a representation of an adaption option, a representation of a consequence/likelihood matrix, an estimated risk, etc. Examples of functional components are a module that estimates the cost, a module that performs an analysis. Interface definitions can be added if needed to support the explanation of your solution for developers. The internals of the components should be described such that the different aspects of decision-making are defined and can be understood by a developer; the implementation of concrete algorithms can be left to the developers.

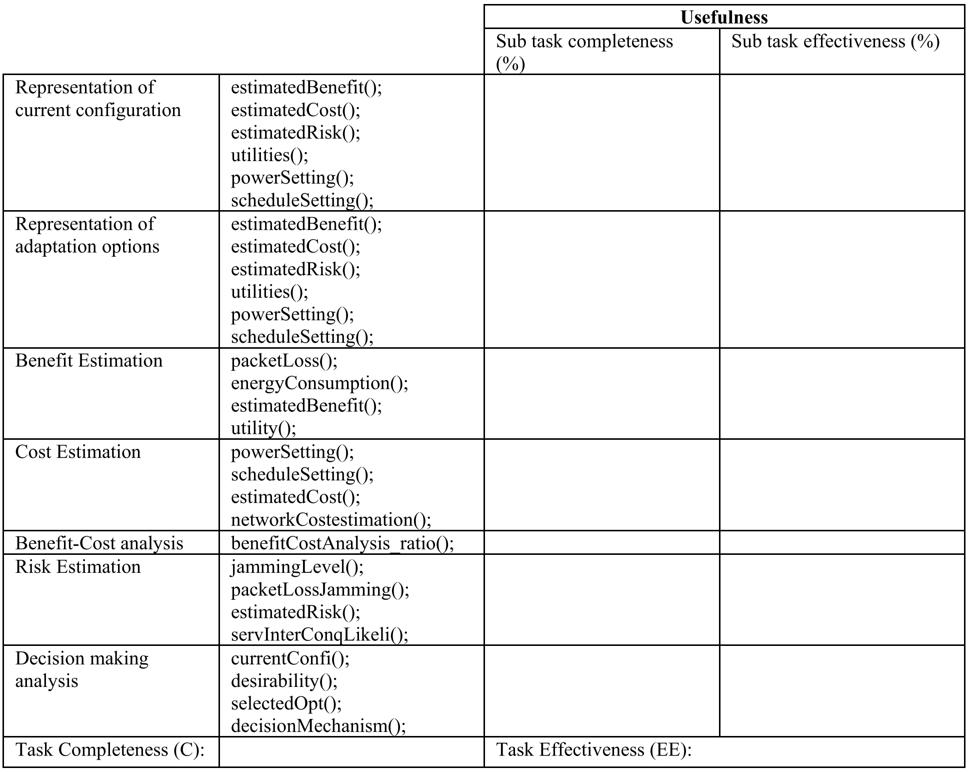
For Part II, besides the current configuration, only two adaptation options need to be considered to determine the need for adaptation. The selection of the adaptation option should comply with the specification of the mechanism for decision-making (Part I) that takes into account the stakeholder requirements.

You can use pen and paper to specify your design. Please make sure that drawings and text are clear and understandable.

C Detailed Metrics for Data Analysis

Figure C1 shows a detailed version of the schema we used to perform the evaluation of solutions submitted for task one. The rows in the table represent possible break-downs of the solution for task one, while columns show metrics of usefulness.

A *representation* refers to a data diagram or textual representation of a concept such as current configuration. It may include actual values for different elements of the concept (e.g., cost of current configuration, power setting, schedule, benefit, etc.) or simply refer to a list of possible elements (e.g., attributes, functions, links and dataflow).



		Usefulness	
		Sub task completeness (%)	Sub task effectiveness (%)

Representation of current configuration	estimatedBenefit();
	estimatedCost();
	estimatedRisk();
	utilities();
	powerSetting();
	scheduleSetting();		

Representation of adaptation options	estimatedBenefit();
	estimatedCost();
	estimatedRisk();
	utilities();
	powerSetting();
	scheduleSetting();		

Benefit Estimation	packetLoss();
	energyConsumption();
	estimatedBenefit();
	utility();		

Cost Estimation	powerSetting();
	scheduleSetting();
	estimatedCost();
	networkCostestimation();		

Benefit-Cost analysis	benefitCostAnalysis_ratio();		

Risk Estimation	jammingLevel();
	packetLossJamming();
	estimatedRisk();
	servInterConqLikeli();		

Decision making analysis	currentConfi();
	desirability();
	selectedOpt();
	decisionMechanism();		

Task Completeness (C):		Task Effectiveness (EE):	

Fig. C1. Usefulness metrics for evaluation of participants' solutions to task one.

Benefit/cost/risk estimation refers to a data diagram or textual description specifying constituting components required for the relevant estimation analysis and their relationships. This may or may not include actual values for the example.

Benefit-cost and decision-making refer to a textual or pseudo mathematical analysis of cost-benefit and decision-making, indicating how the final outcome may be concluded.

Sub-Task Completeness.

(1) Low values (≤ 35%): the main components and/or explanation are presented, however most of the details (i.e., attributes, functions, relations, values) are missing.
(2) Medium values (> 35% and < 70%): the main components and/or explanation are presented; however, half of the details (i.e., attributes, functions, relations, values) are missing.
(3) High values (≥ 70%): the main components and/or explanation are presented and more than half of the details (i.e., attributes, functions, relations, values) are fully provided.

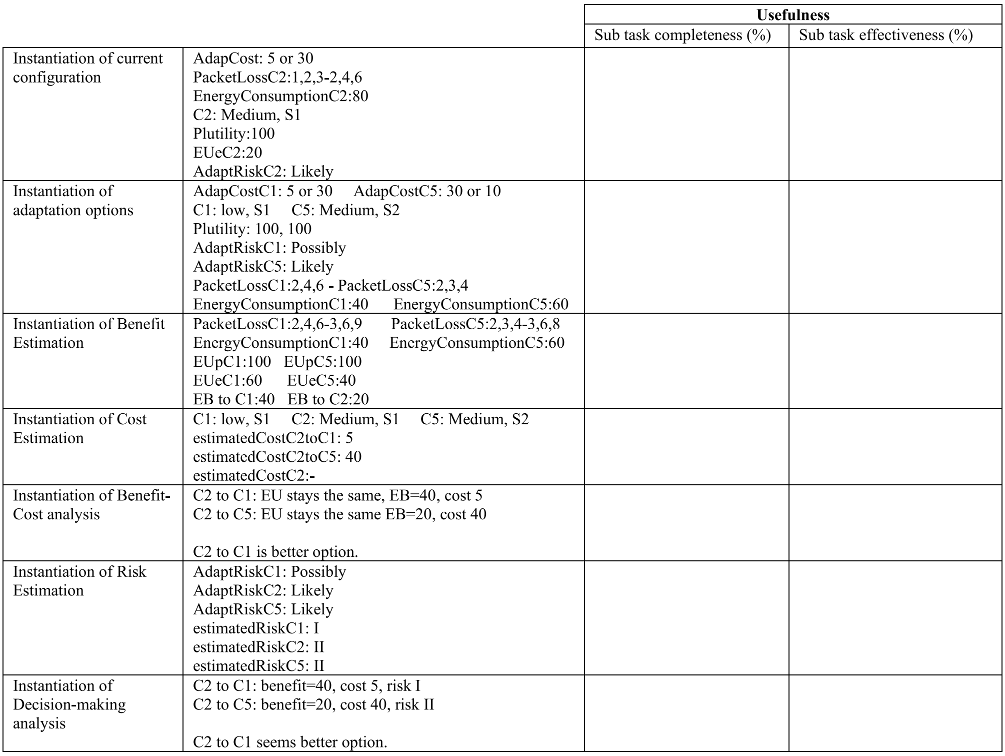
Sub-Task effectiveness.

(1) Low values (≤ 35%): the main components and/or explanation are presented, however most of the details (i.e., attributes, functions, relations, values) are incorrect.
(2) Medium values (> 35% and < 70%): the main components and/or explanation are presented; however, half of the details (i.e., attributes, functions, relations, values) are incorrect.
(3) High values (≥ 70%): the main components and/or explanation are presented and more than half of the details (i.e., attributes, functions, relations, values) are correct.

Task Completeness. Average of parts of the subtasks of a task completed (percentage).

Task Effectiveness. Average of parts of the subtasks of a task correctly completed (percentage).

Figure C2 shows a detailed version of schema we used to analyze submissions of task two.



		Usefulness	
		Sub task completeness (%)	Sub task effectiveness (%)
Instantiation of current configuration	AdapCost: 5 or 30 PacketLossC2:1,2,3-2,4,6 EnergyConsumptionC2:80 C2: Medium, S1 Plutility:100 EUeC2:20 AdaptRiskC2: Likely		
Instantiation of adaptation options	AdapCostC1: 5 or 30 AdapCostC5: 30 or 10 C1: low, S1 C5: Medium, S2 Plutility: 100, 100 AdaptRiskC1: Possibly AdaptRiskC5: Likely PacketLossC1:2,4,6 - PacketLossC5:2,3,4 EnergyConsumptionC1:40 EnergyConsumptionC5:60		
Instantiation of Benefit Estimation	PacketLossC1:2,4,6-3,6,9 PacketLossC5:2,3,4-3,6,8 EnergyConsumptionC1:40 EnergyConsumptionC5:60 EUpC1:100 EUpC5:100 EUeC1:60 EUeC5:40 EB to C1:40 EB to C2:20		
Instantiation of Cost Estimation	C1: low, S1 C2: Medium, S1 C5: Medium, S2 estimatedCostC2toC1: 5 estimatedCostC2toC5: 40 estimatedCostC2:-		
Instantiation of Benefit-Cost analysis	C2 to C1: EU stays the same, EB=40, cost 5 C2 to C5: EU stays the same EB=20, cost 40 C2 to C1 is better option.		
Instantiation of Risk Estimation	AdaptRiskC1: Possibly AdaptRiskC2: Likely AdaptRiskC5: Likely estimatedRiskC1: I estimatedRiskC2: II estimatedRiskC5: II		
Instantiation of Decision-making analysis	C2 to C1: benefit=40, cost 5, risk I C2 to C5: benefit=20, cost 40, risk II C2 to C1 seems better option.		

Fig. C2. Usefulness metrics for evaluation of participants' solutions to task two.

Instantiation refers to a data diagram or textual representation of a concept such as current configuration representing the actual values for different elements of the concept (e.g., cost of current configuration, power setting, schedule, benefit, etc.) in the example scenario.

Instantiation of benefit/cost/risk estimation refers to a data diagram or textual description specifying constituting components required for the relevant estimation analysis and their relationships including actual values for the example scenario.

Benefit/cost/risk estimation refers to textual or pseudo mathematical analysis of cost-benefit and decision-making, indicating the final outcomes for the example scenario.

Benefit-cost and decision-making refers to textual or pseudo mathematical results indicating the outcomes for the relevant estimations in the example scenario.

Sub-Task Completeness.

(1) Low values (≤ 35%): the main components, explanation and/or logic are presented, however most of the details (i.e., the values and instantiation elements) are missing.
(2) Medium values (> 35% and < 70%): the components, explanation, and/or logic are presented, however half of the details (i.e., the values and instantiation elements) are missing.
(3) High values (≥ 70%): the main components, explanation, and/or logic are presented and more than half of the details (i.e., the values and instantiation elements) are fully provided.

Sub-Task Effectiveness.

(1) Low values (≤ 35%): the components, explanation, and/or logic are presented, however most of the details (i.e., the values and instantiation elements) are incorrect.
(2) Medium values (> 35% and < 70%): the components, explanation, and/or logic are presented, however half of the details (i.e., the values and instantiation elements) are incorrect.
(3) High values (≥ 70%): the main component, explanation, and/or logic are presented and more than half of the details (i.e., the values and instantiation elements) are correct.

Task Completeness. Average of parts of the subtasks of a task completed (percentage).

Task Effectiveness. Average of parts of the subtasks of a task correctly completed (percentage).

Data for *satisfaction* is collected through the survey and refers to the level of satisfaction from interacting with model kinds. The values can vary from "1 = not satisfied" to "5 = exceptionally satisfied."

Figure C3 shows the detailed schema used to analyze effectiveness and efficiency of viewpoint and tasks submissions of task two.

Completeness. Average of parts of the subtasks of a task completed (percentage).

Effectiveness. Average of parts of the subtasks of a task correctly completed (percentage).

Time (T). Represents the time spent per task as recorded by the participant.

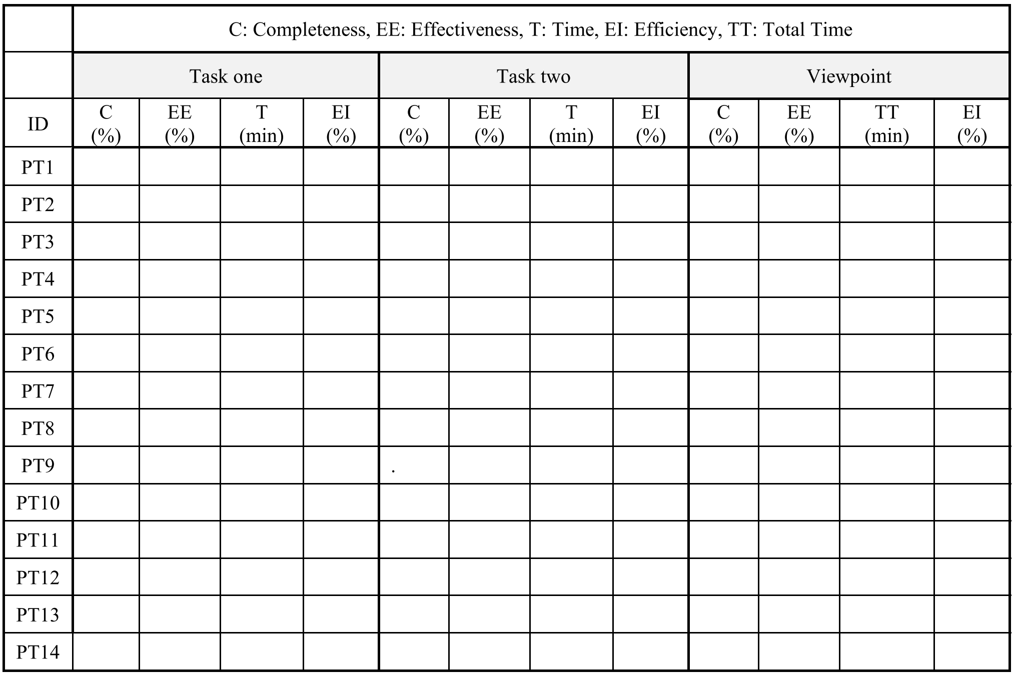
Efficiency. Relative percentage of task effectiveness divided by the time used for a task.

Total Time (TT). Represents total time spent on design tasks (i.e., time spent on task one plus time spent on task two).

Viewpoint Effectiveness. The average of the task effectiveness of all tasks.

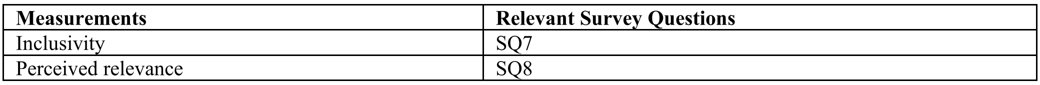
Viewpoint Completeness. The average of the task completeness of all tasks.

Relevant survey questions and task solutions used for analysis of applicability and understandability are detailed in Figures C4 and C5.



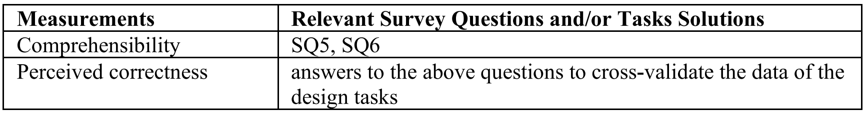
	C: Completeness, EE: Effectiveness, T: Time, EI: Efficiency, TT: Total Time											
	Task one				Task two				Viewpoint			
ID	C (%)	EE (%)	T (min)	EI (%)	C (%)	EE (%)	T (min)	EI (%)	C (%)	EE (%)	TT (min)	EI (%)
PT1												
PT2												
PT3												
PT4												
PT5												
PT6												
PT7												
PT8												
PT9					.							
PT10												
PT11												
PT12												
PT13												
PT14												

Fig. C3. Viewpoint and tasks effectiveness and efficiency analysis schema.



Measurements	Relevant Survey Questions
Inclusivity	SQ7
Perceived relevance	SQ8

Fig. C4. Relevant survey questions for applicability analysis.



Measurements	Relevant Survey Questions and/or Tasks Solutions
Comprehensibility	SQ5, SQ6
Perceived correctness	answers to the above questions to cross-validate the data of the design tasks

Fig. C5. Relevant survey questions or tasks solutions for understandability analysis.

D Statistics

Figures D1 and D2 provide an overview of the use of model kinds while performing tasks as indicated by the participants in the survey, overall and per individual participants, respectively.

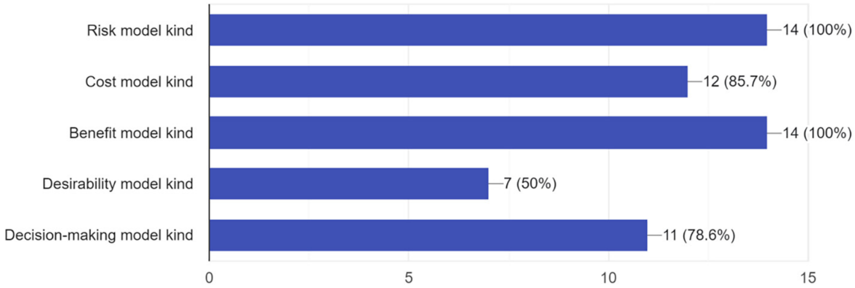
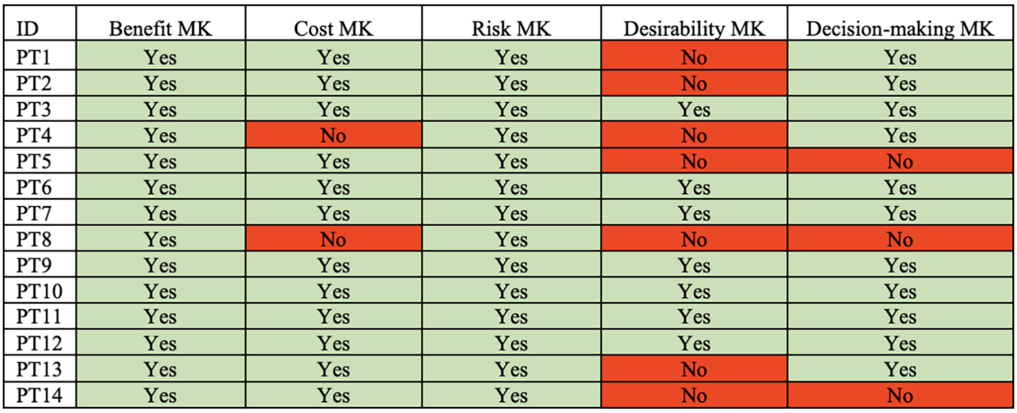


Fig. D1. Use of model kinds by participants.



ID	Benefit MK	Cost MK	Risk MK	Desirability MK	Decision-making MK
PT1	Yes	Yes	Yes	No	Yes
PT2	Yes	Yes	Yes	No	Yes
PT3	Yes	Yes	Yes	Yes	Yes
PT4	Yes	No	Yes	No	Yes
PT5	Yes	Yes	Yes	No	No
PT6	Yes	Yes	Yes	Yes	Yes
PT7	Yes	Yes	Yes	Yes	Yes
PT8	Yes	No	Yes	No	No
PT9	Yes	Yes	Yes	Yes	Yes
PT10	Yes	Yes	Yes	Yes	Yes
PT11	Yes	Yes	Yes	Yes	Yes
PT12	Yes	Yes	Yes	Yes	Yes
PT13	Yes	Yes	Yes	No	Yes
PT14	Yes	Yes	Yes	No	No

Fig. D2. Use of model kinds per participants' IDs.

E Abbreviations

Table E1 lists the main abbreviations used in the article.

Table E1. Main Abbreviations Used in the Article

Abbreviation	**Full Term**
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
MAPE	Monitor-Analyze-Plan-Execute
CBAM	Cost-Benefit-Analysis Method
MK_i	Model Kind i
C_i	Configuration/Adaptation option i
Ci	Concern i
EB_{C_i}	Estimated Benefit for adaptation option C_i
$U_{C_i}^{Ci}$	Utility of adaptation option C_i for concern Ci
S_{SP}^{ST}	Service of type ST provided by provider SP
EC_{C_i}	Estimated Cost for adaptation option C_i
VFC_{C_i}	Value-For-Cost for adaptation option C_i
s_b	Scaling function estimated benefit
s_c	Scaling function estimated cost
LC_{C_i}	Likelihood of adaptation option C_i for concern Ci
CC_{C_i}	Consequence of adaptation option C_i for concern Ci
ER_{C_i}	Estimated Risk for adaptation option C_i
$EBCR_{C_i}$	Estimated Benefit - Cost - Risk for adaptation option C_i
T	Threshold
$\times$	Multiplier

Received 5 December 2023; revised 3 August 2024; accepted 6 November 2024