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

DANNY WEYNS, Katholieke Universiteit Leuven, Belgium and Linnaeus University, Sweden
PARIS AVGERIOU, University of Groningen, The Netherlands
RADU CALINESCU, University of York, UK
SARA M HEZAVEHI, University of Groningen, The Netherlands and Linnaeus University, Sweden
RAFFAELA MIRANDOLA, Politecnico di Milano, Italy
DIEGO PEREZ-PALACIN, Linnaeus University, Sweden

Abstract. Over the past two decades, researchers and engineers have extensively studied the problem of how to enable a software system to deal with uncertain operating conditions. One prominent solution to this problem is *self-adaptation*, which 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 developed so far use decision-making mechanisms that focus on achieving a set of goals, i.e., that select for execution the adaptation option with the best *estimated benefit*. A few approaches have also considered 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 factors when evaluating the options for adaptation when mitigating uncertainty is essential, not only for satisfying the concerns of the stakeholders, but also to ensure safety and public acceptance of self-adaptive systems. In this paper, we present an ISO/IEC/IEEE 42010 compatible architectural viewpoint that considers the estimated benefit, cost, and risk as core factors of each adaptation option considered in self-adaptation. The viewpoint aims to support software architects responsible for designing robust decision-making mechanisms for self-adaptive systems.

1 INTRODUCTION

Modern software systems are expected to deal with changing operating conditions. Examples are dynamic workloads, fluctuating availability of services, and changes in the environment of the system. For designers, these changes create uncertainties that may be difficult to anticipate before deployment [9, 15, 27, 29, 30]. Yet, without mitigation, such uncertainties may jeopardise the concerns of the stakeholders of the system. One approach to mitigate uncertainties is self-adaptation [12, 33]. Self-adaptation extends a system with a feedback loop that tracks the system and its environment to resolve the uncertainties and adapt the system to deal with the changing conditions, or gracefully degraded if necessary. Over the past two decades, researchers and engineers have extensively studied the problem of how to realise self-adaptation using feedback loops [33].

A common approach is to equip the feedback loop with a decision-making mechanism that evaluates the relevant options for adaptation (i.e., the possible configurations of the system that are considered to adapt the system) and selects the option with the best expected outcome in terms of achieving the system goals. We refer to this as the *estimated benefit* that can be achieved by self-adaptation when a particular adaptation option is selected for execution. Consider for instance an e-health system that remotely monitors vital parameters of elderly people and takes action when needed; for instance, change drugs, visit and support the elderly, or send an emergency team. To perform the analysis of vital parameters or dispatch health carers when needed, the system uses third-party services. The load of the system (i.e., service requests) may dynamically change in ways

Authors' addresses: Danny Weyns, Katholieke Universiteit Leuven, Belgium , Linnaeus University, Sweden, danny.weyns@kuleuven.be; Paris Avgeriou, University of Groningen, The Netherlands; Radu Calinescu, University of York, UK; Sara M Hezavehi, University of Groningen, The Netherlands , Linnaeus University, Sweden; Raffaela Mirandola, Politecnico di Milano, Italy; Diego Perez-Palacin, Linnaeus University, Sweden.

that may be difficult to predict; for instance, the health conditions of elderly may be affected by changing weather. To deal with such changes, the system may dynamically adapt the allocation of service requests to third-party services such that the response time of the service remains below a given threshold, while the reliability of the system is maximised. To that end, the system will estimate the benefit in terms of the values of the required quality attributes (i.e., response time and reliability in our example) for different compositions of services provided by the service providers, and then adapts the system such that it selects the best composition of services [35].

A few approaches also take into account the *estimated cost* for adapting the system when evaluating the options for adaptation, see for instance [23, 32]. In this context, we use 'cost' to refer to the one-off cost to perform an adaptation of the running system.[1] As an example, switching services in the e-health system may require the system to test the reliability of newly selected services before using them. Such tests require extra resources and time. The level of testing that is required may depend on the service-level agreement of the service providers of the services under test. When assessing a composition of services provided by third-parties, the system will calculate the estimated cost for testing the different newly integrated services of the composition. Hence, when making adaptation decisions, the system should not only consider the estimated benefit of the each of the adaptation options, but should also take into account the estimated cost of the different service compositions that can be selected.

Besides estimated benefit and cost, we argue that adaptation decisions should also take into account the *estimated risk* the system would be exposed to if an adaption option were selected and used to adapt the running system. Considering that 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 jeopardise ethical or legal concerns [16]. With the advancing digitisation of society and industry, these aspects apply to a wide variety of computing systems that are becoming vital for our society. Compared to other areas where risks are taken into account in decision-making, see for instance [2, 3, 6, 20], the decision-making of self-adaptive systems is lagging behind. Yet, compared to other areas, where risk analysis is done by humans supported by tools, risk analysis in self-adaptive systems needs to be done automatically by the self-adaptive system itself within the time window that is available to take adaptation decisions. Clearly, this calls for solid preparation during system design, such that adaptation decisions that take risk into account can be made efficiently by the system during operation. This paper aims at making a step forward towards dealing with risk in self-adaptive systems. We illustrate the importance of considering estimated risk in the e-health system, which is an example of a valuable computing system for society. Selecting a new composition of services from different service providers may affect the privacy of the elderly people that use the services. In particular, different service providers may apply different rules regarding the collection and treatment of data from the e-health system. Hence, the selection of a particular set of services implies a particular data privacy risk for the system users. When selecting services, the system should estimate this risk, e.g., based on the trustworthiness of the service providers. Hence, decision-making in self-adaptation should in general take into account the estimated benefit, cost, and risk of the different options considered for adaptation.

This paper presents an architectural viewpoint for decision-making in self-adaptive systems that takes into account estimated benefit, cost, and risk as first-class citizens. The primary users that will benefit from the viewpoint are the software architects of self-adaptive systems. The viewpoint is structured using the template recommended by the ISO/IEC/IEEE 42010 standard [22]. This

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

standard defines an architecture viewpoint 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 an architect 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, modelling templates, analytical methods and possibly other operations useful on models of the model kind. A viewpoint can be instantiated for a domain at hand, resulting in a view that comprises architecture models that address the concerns framed by its governing viewpoint. As such, the viewpoint presented in this document 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 is grounded in a study of the literature and supported with extensive experiences in engineering self-adaptive systems across a wide range of domains.

2 SCOPE OF THE VIEWPOINT

Given the wide range of approaches used to realise self-adaptation in general, and decision-making in self-adaptive systems in particular, it is essential that we clarify the definition 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 centred on *architecture-based adaptation* [17, 26, 28, 37], which is an established approach to engineering self-adaptive systems. Architecture-based adaptation has a dual focus [34]: on the one hand the use of software architecture as an abstraction to *define* a self-adaptive system at design time, and on the other hand the use of architectural models to *reason* about change and make adaptation decisions at runtime. The viewpoint presented in this paper focusses on the second aspect. Architecture-based adaptation makes a distinction between domain concerns that are handled by the *managed system*, i.e., what the system should provide to the user in terms of functions or services, and adaptation concerns that are handled by the *managing system*, i.e., how the domain concerns are achieved in terms of benefits, cost, and risk. A reference approach to realise the managing system is by means of a so-called MAPE feedback loop [24, 38]. MAPE refers to the basic functions realised by the feedback loop: Monitor the system and its environment, Analyse the situation and the options for adaptation, Plan the adaptation of the managed system for the best 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 [37] (MAPE is therefore sometimes also referred to as MAPE-K). It is important to note that MAPE provides a reference model that describes the essential functions of a managing system and the interactions between them. A concrete architecture maps the functions to corresponding components, which can be a one-to-one mapping or any other mapping, such as a mapping of the analysis and planning functions to one integrated decision-making component.

The focus of the viewpoint is on the adaptation concerns, in particular the decision-making process to select 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 analysed, 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 [18] (for instance the frequency of changes or their intensity). 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 Cost Benefit Analysis Method (CBAM) [11] and the IEC 31010:2019 standard on risk management and risk assessment techniques [7]. CBAM is an established method for analysing 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 under uncertainty.

3 RUNNING EXAMPLE

We illustrate the different parts of the architectural viewpoint using a classic example from the literature [35]: the e-health system that we already used in the introduction. Consider a simple service-based system as shown in Figure 1. This system offers a remote health-assistance service to patients, and relies on data collected via wearable devices. The health-assistance service is realised by a set of specific services that are executed in a workflow as shown in the figure. The core of the application exploits 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 analyses the data, and depending on the analysis results nothing needs to be done or action is required. In the latter case, the medical service 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 medical staff to visit the patient. 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 to 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 data is managed.

At runtime, it is possible to pick any combination of the available services offered by the service providers. The adaptation goals that express the benefits of adaptation are to keep the average failure low, while minimising the resources required to run the e-health service. Switching services in the system may imply a cost associated with 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 on the confidentially of the data of patients within legal constraints; e.g., kept strictly local, stored with partners, shared with partners, non-specified. Finally, medical analysis services perform their analysis based on the measurements of a limited period using bounded computational resources, there is a risk that the diagnoses derived from the analysis results may not not be 100% accurate. This may indirectly affect the health conditions of the patients.

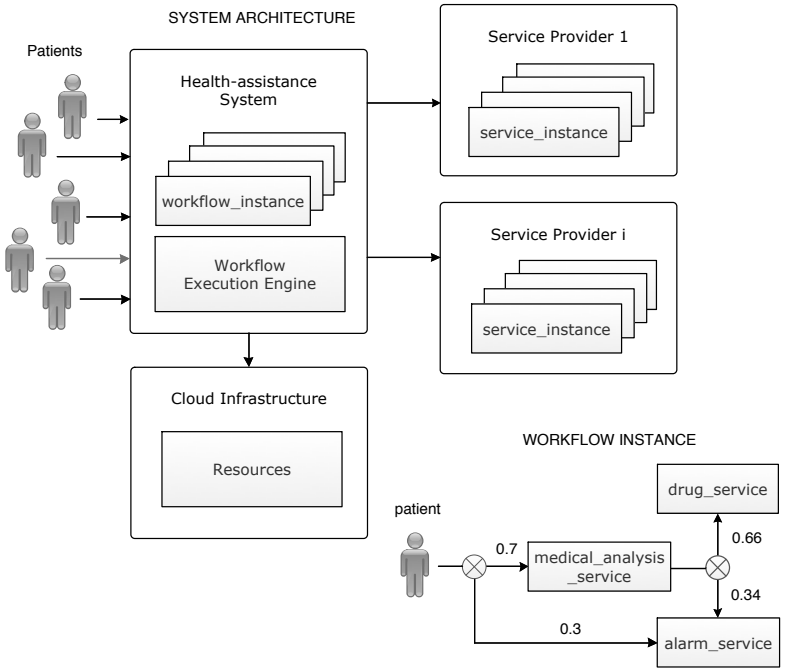


Fig. 1. Service-based system: system architecture and workflow instance

4 SPECIFICATION OF THE VIEWPOINT

Based on a thorough analysis of the literature [27, 29], a number of surveys of the community [9, 18, 19], and our own experiences with developing self-adaptive systems in a variety of domains, e.g., [10, 21, 32, 36], we defined the architecture 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 realising a decision-making module for self-adaptation taking into account the estimated benefit, cost, and risk of the relevant adaptation options.

It is important to put the viewpoint in a broader context of the design of the feedback loop of a self-adaptive system. 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 (in the MAPE workflow) [34]. 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 analysed 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 realising 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.

4.1 Stakeholders and Concerns

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

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 environmemt, 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 risks 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 correspond 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 cost 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 benefit against cost of an adaption option in order to express its desirability? How to compare and rank desirability results?
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 desirability of adaptation options with estimated risk when making adaptation decisions?

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 behaviour 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 benefits of the self-adaptive system, the costs that may be induced by adaptation, and the risks 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

[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.

a high desirability has a high benefit and a low cost. Users are primarily interested in the benefits provided by the self-adaptive system as well as the risks they may be exposed to. Others are those people that 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: *benefits* of adaptation (architects, owners, and users), *costs* of adaptation (architects and owners), *desirability* of adaptation (architects, owners), *risks* 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 realisation 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, optimise the resources to operate the system, keeping the cost required for adaptation low, and minimising the exposed risks. The users are the patients that use the service via a wearable device either to analyse vital parameters or to direct 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 the data privacy and decisions made by the system. Others can be relatives and friends of patients, care professionals, among other people.

4.2 Viewpoint Model Kinds

The viewpoint comprises five model kinds. Table 2 presents the first three model kinds: *benefit estimation* (MK1), *cost estimation* (MK2), and *benefit-cost analysis* (MK3). Table 3 presents the last two model kinds: *risk estimation* (MK4) and *decision-making* (MK5).

Figure 2 gives a high-level overview of the process to use the model kinds. The process starts with the design of the benefit estimation model using the benefit estimation model kind. This model deals with concern C1 (benefit of adaptation options). Next (or in parallel), the cost estimation model can be designed using cost estimation model kind. This model deals with concern C2 (cost of adaptation options). The next step in the process is the design of the benefit-cost estimation model. This model takes as input the benefit-cost estimation model kind, the benefit estimation model and the cost estimation model. The benefit-cost estimation model deals with concern C3 (desirability of adaptation options). Then (or in parallel), the risk estimation model is designed using the risk estimation model kind. This model deals with concern C4 (risk of adaptation options). 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 concern C5 (decision-making, i.e., selecting the best adaptation option).

We explain now each model kind in detail and illustrate it with an example of the health-assistance service system. More elaborated descriptions of the core elements used on the model kinds are provided in the Appendix.

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

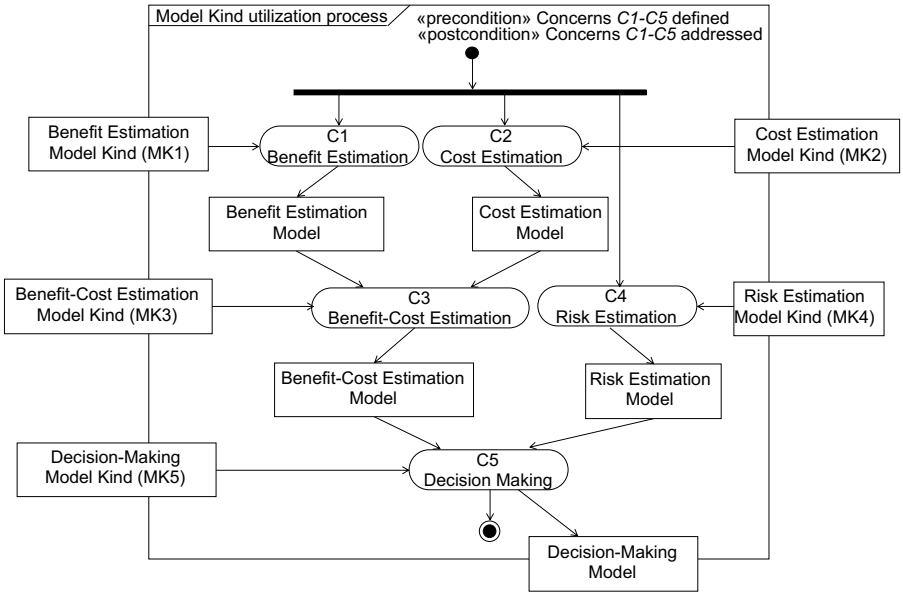


Fig. 2. Process to use model kinds.

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 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 4 shows an instance of the benefit estimation model kind for the health assistance system. The current configuration 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 at utility responses. Figure 3 shows the utility response curves for both goals.

As shown in the left part, 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 part of the figure shows the utility preference for resource usage, defined in a similar way. As an example, configuration C_1 has a 100% utility preference for a failure rate of 0.5% and 50% for a resource usage of 18 units. Adaptation option C_2 has a utility preference of 85% for a failure rate of 1.3% and 100% for a resource usage of 3 units. Current configuration C_c on the other hand has a utility preference of 65% for a failure rate of 1.5% and 50% for a resource usage of 15 units. The utility estimator determines the estimated utilities for each relevant adaptation option. To that

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

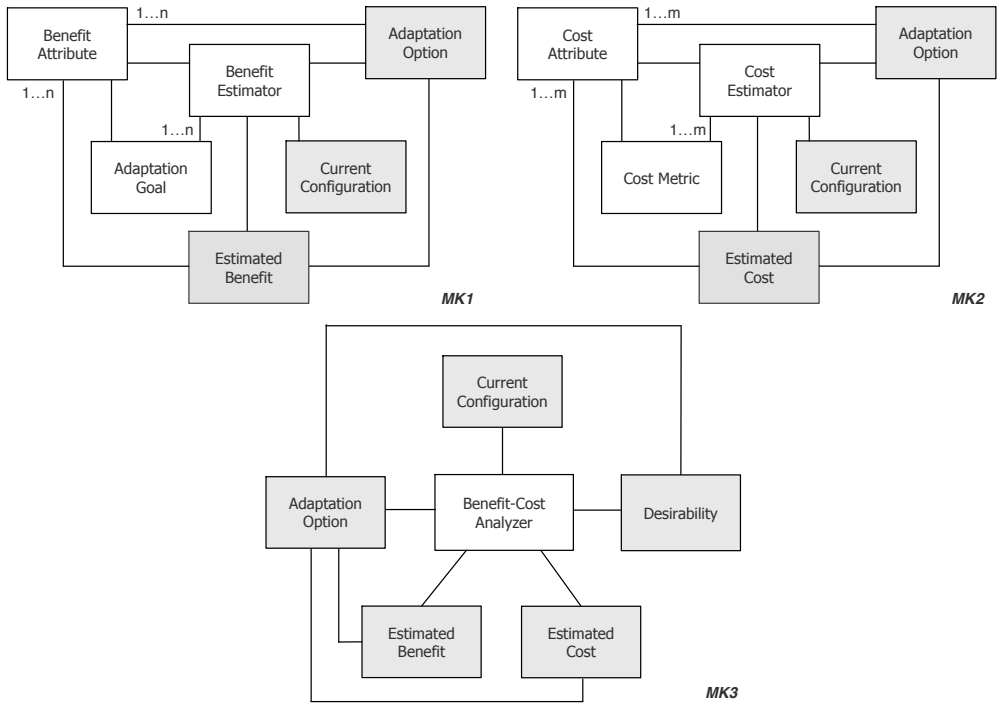
Model Kinds (description):

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 costs metrics.

MK3 - Benefit-Cost Estimation (deals with concern C3): describes per adaptation option how the desirability is determined based a benefit-cost analysis using the estimated benefit and estimated cost of that adaptation option.

Model Kinds (meta-models):



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

end, the estimator configures a runtime model of the workflow for each combination of concrete services (adaptation options) together with the actual probabilities that paths are selected. This model is then analysed by a statistical model checker that runs a number of simulations for each adaptation option. The result of the analysis is an estimate for failure rate and resource usage for each adaptation option with a 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

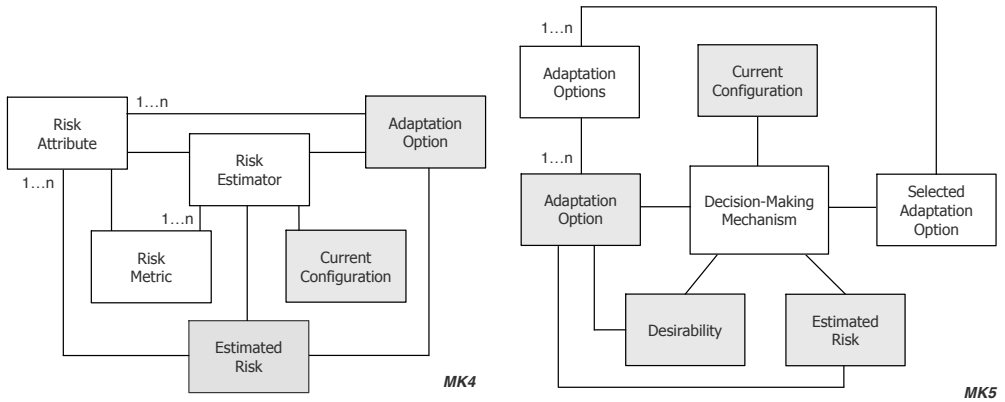
Table 3. Viewpoint – Model Kinds MK4 and MK5

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 (gray boxes represent model elements shared among model kinds)

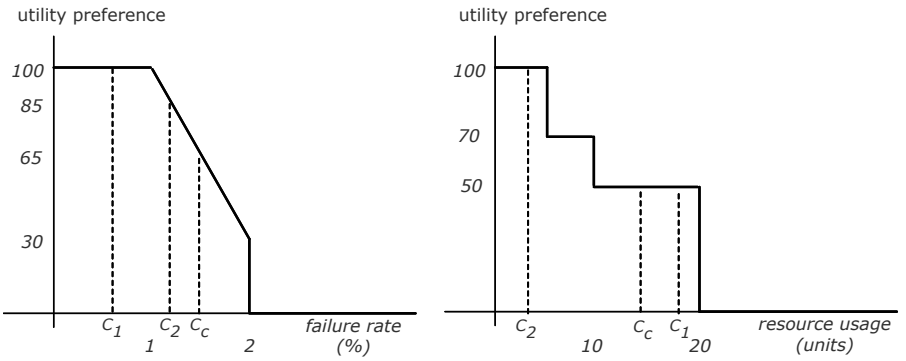


Fig. 3. Example utility response curves

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

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

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 the weight for failure rate (in the example 0.7); the second term with a similar structure refers to resource usage R (with weight 0.3). Applied to adaptation option C_1 in Figure 3 the estimated benefit is:

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

Similarly, the benefit of adaptation option C_2 in Figure 3 is:

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

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

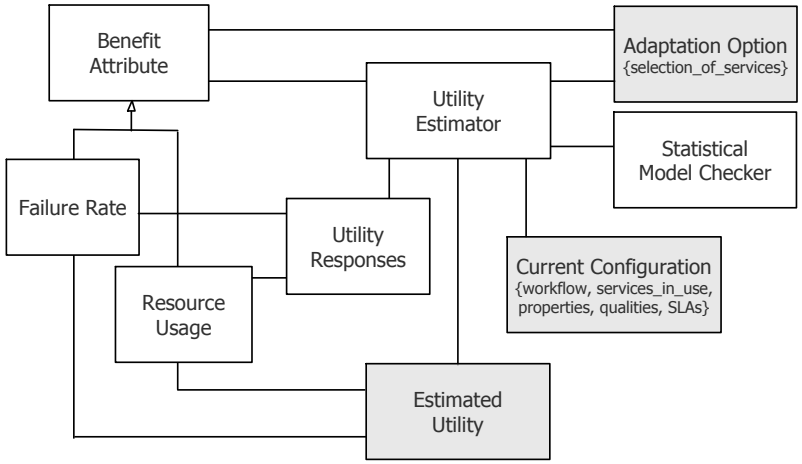


Fig. 4. Example instance of Benefit Estimation Model Kind

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 an 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 to quantify 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 list of cost attributes with associate cost metrics, see Table 5. This table is based on examples extracted from the literature and our own experiences, see for instance [23, 25, 32]. We distinguish three groups of adaptation costs. The first group refers to resources that are required to realise adaptation, such as bandwidth, processor time, memory, and power. The second group relates to overhead of

the system in terms of quality properties that are affected by adaptation, which can be reduced availability, a performance penalty, or costs to deal additional security vulnerabilities implied by adaptation. The third group refers to a monetary price that is implied by the realisation of adaptation. Table 5 is not meant to be exhaustive and can be easily refined or extended.

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

Example. Figure 5 shows an instance of the cost estimation model kind for the health assistance system. We consider one cost attribute: the tests of new services that are considered by the adaptation options [8]. Testing overhead requires extra resources, which corresponds to computation overhead in Table 5. The amount of testing that is needed may depend on trustworthiness of service providers. Table 5 illustrates a possible cost model.

Table 5. Example cost model for e-health system with required resources for services.

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

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:

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

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 cost only applies to *DS* and *AS*, i.e., the services that need to be tested before they can be adapted. The cost for testing the different services can be found in Table 5. Similarly, t]he 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 to test only one new services, the estimated cost of C_1 is smaller as 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 service level agreement, 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.

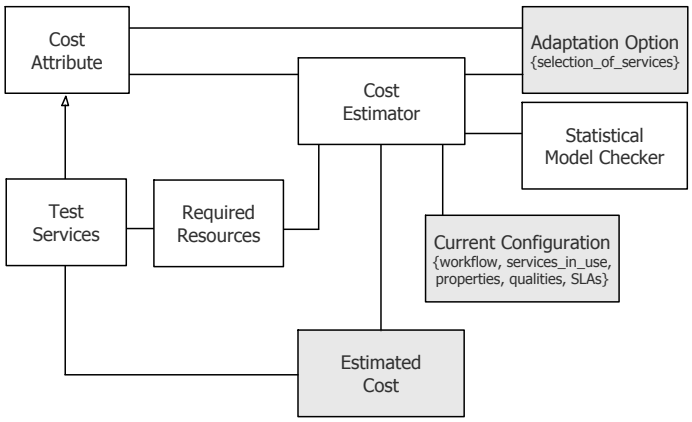


Fig. 5. Example instance of Cost Estimation Model Kind

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 that stakeholders prefer the selection of an adaptation option C_i over other options by comparing total estimated benefit with total estimated cost. The desirability of adaptation options increases with higher estimated benefit and lower estimated cost. The *benefit-cost analyser* computes the desirability of an adaptation option. The computation of desirability 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 so called "Value-For-Cost" that defines desirability as the ratio of estimated overall benefit to estimated overall cost (both scaled). 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 6 shows the benefit-cost model kind instantiated for the health assistance system. We use value-for-cost (VFC) to express the desirability of adaptation options. The value-for-cost 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 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 that we already used to illustrate estimated benefit and estimated cost, we obtain the following values for VFC.

$$VFC_{C_1} = \frac{24.5}{4} = 6.13; \quad VFC_{C_2} = \frac{29.0}{6} = 4.83 \tag{9}$$

Although adaptation option C_2 has a higher estimated benefit as adaptation option C_1, the desirability of adaptation option C_1 in terms of value-for-cost is significantly better as for 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 value-for-cost based on estimated benefit and cost of only adaptation options C_1 and C_2, adaptation option C_1 would be selected for adaptation.

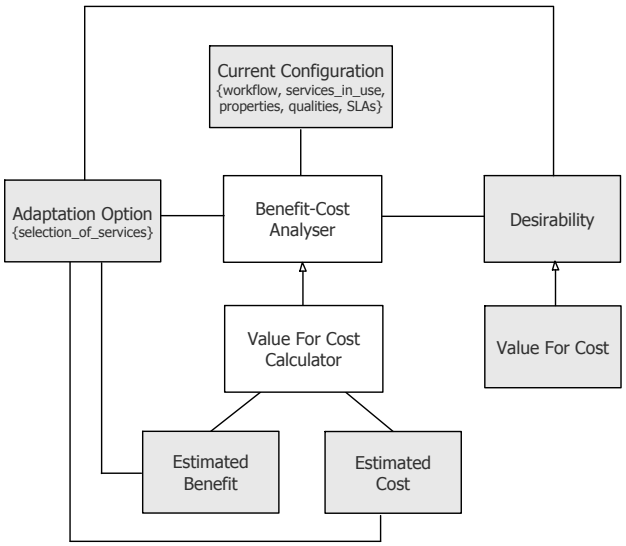


Fig. 6. Example instance of Desirability Estimation Model Kind

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 potential effects of uncertainties on system objectives in terms of their likelihoods and consequences (positive or negative or both) [7]. A *risk attribute* represents a particular type of risk that is implied by adapting the managed system with a given adaptation option. Different risk attributes can be associated with applying an adaptation option, such as safety, environment, finances, etc. *Risk metrics* define measures to quantify 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 risks. Essential to these techniques is that they capture the stakeholders their concerns and intent [16]. Example mechanisms include consequence/likelihood matrix, cause-consequence analysis, and decision-tree analysis [7]. These methods differ in the purpose of the assessment, the information that is requited/available, the importance of the decision, the time available to make a decision, among other criteria. Hence, the choice for 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 list of high-level risk attributes with associate risk metrics, see Table 6. This table is extracted from literature on risks, including [4, 7, 13, 14, 16, 31].

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

Example. Figure 7 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 not 100% accurate analysis results. We illustrate risk estimation for the health assistance system using a *consequence/likelihood matrix*.

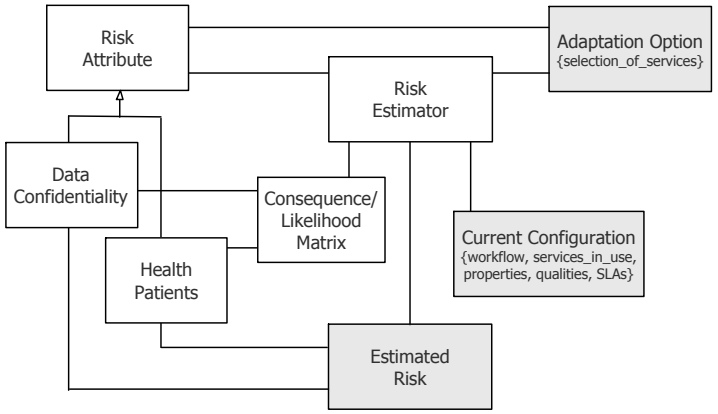


Fig. 7. Example instance of Risk Estimation Model Kind

A consequence/likelihood matrix (or risk matrix) enables specifying an estimated risk according to its consequence and likelihood. The matrix requires customised 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.[3] For instance, a service provider with label gold will store patient data locally. Exposure of data is expected to happen rarely and if it would happen it will have a negligible effect. On the other hand, a service provider with bronze label will share patient data with partners. Consequently, exposure of data is likely and if it would happen 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 to 4). We apply the following rules in the example:

[3]The metrics used in this example are inspired by examples provided in the Risk Standard IEC 31010:2019 [7]. Yet, this is just an example for illustration purposes, a wide variety of scales and values may apply depending on the situation at hand.

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)

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

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

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 8 shows a consequence/likelihood matrix for the risk attribute confidentially of the data. In the example, the scale for consequences for confidentiality of data in terms of "data exposure" have 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 (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 all provided by provider 1) is mapped as follows:

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

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

A value of $LC_{C_2} = 2$ corresponds to likelihood of data exposure "possibly" and a 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., $ERL_{C_2}^{Data} = 2$ (see Figure 8). Likewise, adaptation option C_1 (with a medical analysis service provided by service provider 1, an a drug and alarm service provided by provider 2) can be mapped to likelihood "rarely" and consequence "limited impact," 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 on 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 [7]. 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 very low risks of the other attributes when combined. To that end, adaptation options with estimated risks above certain thresholds may be ruled out before composing risk attributes.

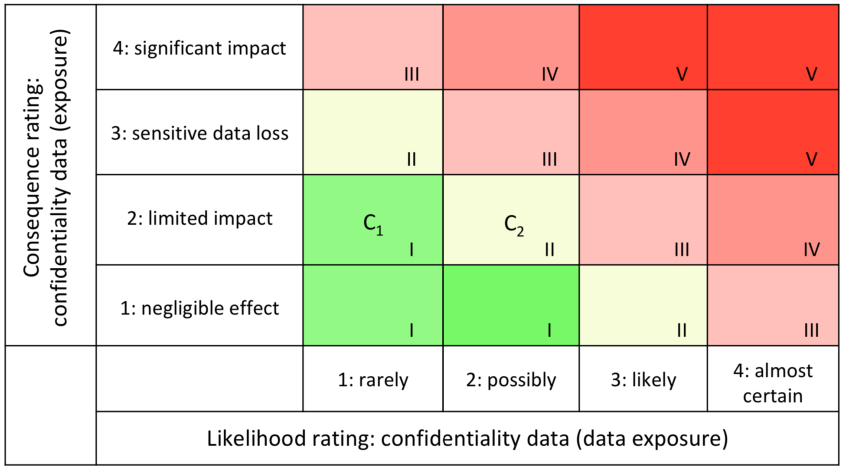


Fig. 8. Example of a consequence/likelihood matrix for confidentially of the data.

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 x 4 consequence/likelihood matrix for health of patients with risk levels I to V similar to the consequence/likelihood matrix for data confidentiality. Further, assume that the estimated risk for 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 estimate risk would matter.

Decision-Making Model Kind (MK5). The fifth and last model kind describes how adaptation decisions are made (see Table 3 right). A *decision-making mechanism* provides the means to select an adaptation option from the set of available options taking into account the estimated desirability (based on the estimated desirability analysis in terms of estimated benefit and cost), and estimated risk (based on estimated risk analysis). The *selected adaption option* C_s represents the new configuration that is selected to be applied to the managed system in order to adapt it. In general decision-making realises 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 [5] that may require additional or more detailed data.

Example. Figure 9 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 value-for-cost and risk.

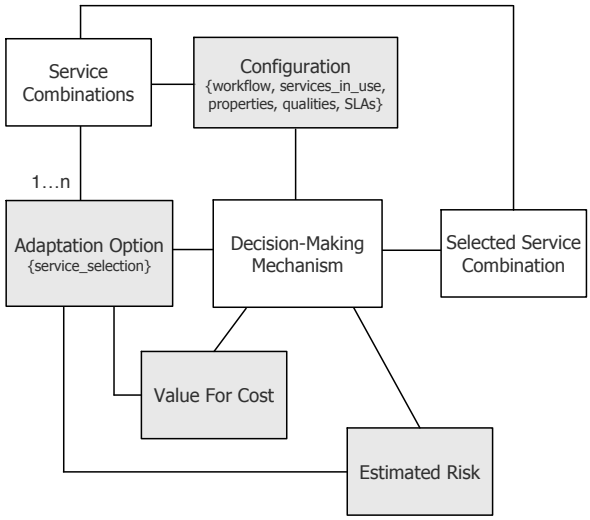


Fig. 9. Example instance of Decision-Making Model Kind

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

$$EBCR_{C_1} = ED_{C_1} * W_{VFC} - ER_{C_1} * W_R = 6.13 * 0.5 - 1.8 * 0.5 = 2.17 \tag{18}$$

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

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

4.3 Viewpoint Analysis

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

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.

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 balancing the tradeoffs between estimated benefit and estimated cost when designing the benefit-cost analyser. This analysis is usually performed at design time, but may also 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 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 $s_c(EC_{C_i}) = EB_{C_i}$ and we applied that to determine estimated VFC of two adaptation options (C_1 and C_2).

We look now 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 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 making the sum of the threshold and the fraction of the estimate above the threshold multiplied with a factor x. Figure 11 shows how estimated value-for-cost (VFC_{C_i}) is determined for two adaptation options C_1 and C_2.

In this particular setting, threshold T is set to 25 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 desirability model kind ($VFC_{C_1} = 6.13$ and $VFC_{C_2} = 4.83$). As we can see, for values of x equal or smaller than 3.9 the value-for-cost 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 $x = 3.9$ the higher contribution of estimated benefit of C_2 above the threshold (namely 29.0 compared to 24.5 for C_1),

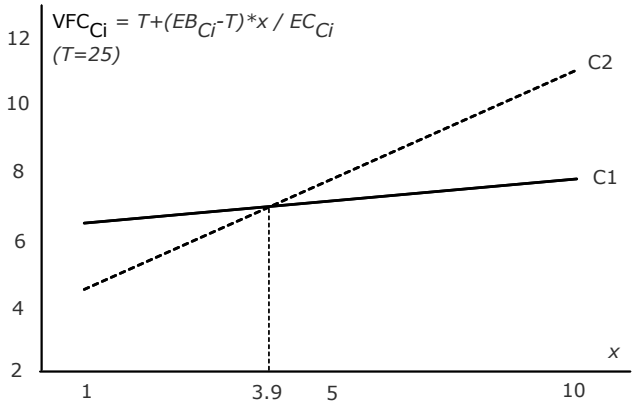


Fig. 10. Change of desirability (VFC) based on weights for estimated benefit and cost.

would compensate the higher cost (6 for C_2 versus 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 balancing estimated desirability (based on estimated benefit and cost) and estimated risk in the decision-making process. The results will help determining 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 how a change of the weights have an effect on the decision-making. Figure 11 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., 0.31 and hence, values of W_R approximately equal or larger than 0.69) adaptation option C_2 would be preferred over C_1.This means, the stakeholders would give much more attention to risk as to desirability in terms of value-for-cost. 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.

5 CONCLUSION

In this paper, we presented a specification of an architecture viewpoint for benefit-cost-risk-aware decision-making in self-adaptive systems, aligned with ISO/IEC/IEEE 42010. The viewpoint is intended to address the concerns of architects, system owners, users and other stakeholders of self-adaptive systems. The novelty of the viewpoint lays in the combination of estimated benefit, cost, and risk as first-class citizens in the decision-making process to select configurations to adapt the system. We devised the viewpoint to be flexible in the mechanisms used to make different decisions. The viewpoint is also intended to be compatible with other architectural approaches that

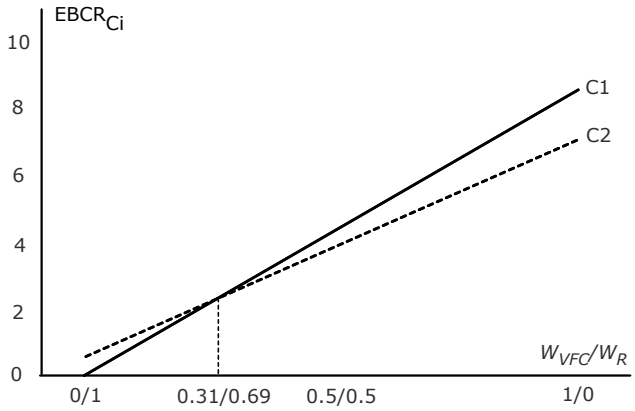


Fig. 11. Changing decision based on weights for estimated desirability (VFC) and risk.

are required to design other aspects of a self-adaptive system, such as monitoring of the system and its environment, the generation of plans, and the execution of these plans to enact adaptations.

REFERENCES

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

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

[3] 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. https://doi.org/10.1109/ICSE.2013.6606612

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

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

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

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

[8] Radu Calinescu, Naif Alasmari, Colin Paterson, , and Raffaela Mirandola. 2021. Quantitative Verification with Adaptive Uncertainty Reduction. *Submitted* (2021).

[9] R. Calinescu, R. Mirandola, D. Perez-Palacin, and D. Weyns. 2020. Understanding Uncertainty in Self-adaptive Systems. In *International Conference on Autonomic Computing and Self-Organising Systems*.

[10] R. Calinescu, D. Weyns, S. Gerasimou, M. 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. https://doi.org/10.1109/TSE.2017.2738640

[11] CBAM. 2018. Cost Benefit Analysis Method, SEI, CMU https://www.sei.cmu.edu/architecture/tools/evaluate/cbam.cfm. (2018).

[12] Betty HC Cheng et al. 2009. Software engineering for self-adaptive systems: A research roadmap. In *Software engineering for self-adaptive systems*. Springer.

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

[14] Mustafa Elmontsri. 2014. Review of the Strengths and Weaknesses of Risk Matrices. *Journal of Risk Analysis and Crisis Response* 4 (2014), 49–57. Issue 1. https://doi.org/10.2991/jrarc.2014.4.1.6

[15] 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.

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

[17] 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.

[18] Sara M. Hezavehi, Danny Weyns, Paris Avgeriou, Radu Calinescu, Raffaela Mirandola, and Diego 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. https://doi.org/10.1145/3487921

[19] Sara M. Hezavehi, Danny Weyns, Paris Avgeriou, Radu Calinescu, Raffaela Mirandola, and Diego Perez-Palacin. 2021. Uncertainty in Self-Adaptive Systems: A Research Community Perspective. arXiv:2103.02717 [cs.SE]

[20] Tobias Holstein, Gordana Dodig-Crnkovic, and Patrizio Pelliccione. 2018. Ethical and Social Aspects of Self-Driving Cars. *arXiv, 1802.04103, cs.CY* (2018). arXiv:1802.04103 [cs.CY]

[21] U. Iftikhar and D. Weyns. 2014. ActivFORMS: Active Formal Models for Self-Adaptation. *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (2014), 125–134. https://doi.org/10.1145/2593929.2593944

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

[23] 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. https://doi.org/10.1109/ICDCS.2010.88

[24] Jeffrey O. Kephart and David M. Chess. 2003. The Vision of Autonomic Computing. *Computer* 36, 1 (2003), 41–50.

[25] 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*. IEEE Press, 114–125. https://doi.org/10.1109/SEAMS.2019.00023

[26] Jeff Kramer and Jeff Magee. 2007. Self-Managed Systems: An Architectural Challenge. In *Future of Software Engineering (FOSE '07)*. IEEE Computer Society, 259–268.

[27] 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*. Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-802855-1.00003-4

[28] 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.

[29] 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.

[30] 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*.

[31] Laurens Sion, Koen Yskout, Dimitri Van Landuyt, and Wouter Joosen. 2018. Risk-Based Design Security Analysis. In *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment* (Gothenburg, Sweden). ACM, 11–18. https://doi.org/10.1145/3194707.3194710

[32] 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*. SCITEPRESS - Science and Technology Publications, Lda, 478–490. https://doi.org/10.5220/0006815404780490

[33] Danny Weyns. 2019. Software Engineering of Self-adaptive Systems. In *Handbook of Software Engineering*. Springer, Cham, 399–443.

[34] Danny Weyns. 2020. Introduction to Self-Adaptive Systems: A Contemporary Software Engineering Perspective. Wiley. ISBN 978-1-119-57494-1.

[35] 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.

[36] Danny Weyns, M. Usman Iftikhar, Danny Hughes, and Nelson Matthys. 2018. Applying Architecture-Based Adaptation to Automate the Management of Internet-of-Things. In *12th European Conference on Software Architecture (Lecture Notes in Computer Science, Vol. 11048)*. Springer, 49–67.

[37] 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.

[38] 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. https://doi.org/10.1109/SEAMS.2013.6595487

APPENDIX A: TERMINOLOGY

Current configuration represents the actual aspects of the managed system and the environment that are relevant to adaptation. This includes the current configuration of components of the managed system, the current values of the properties of interest, and the current values of uncertainties that are relevant to adaptation. In the service-based system example, the current configuration consists of the current set of services used in the workflow, the actual values associated with the different paths in the workflow, and the current values of properties such as reliability, and service level agreements.

Adaptation options are the possible configurations that can be reached by adapting the current configuration of the system. In the service-based system, the adaptation options are determined by the combination of all concrete services provided by the service providers that can be composed to in the workflow.

Adaptation goals represent the quality requirements that need to be achieved by the managing system. In the service-based system example, the adaptation goals are failure rate and resources usage service invocations.

Benefit attributes of an adaptation option are the estimated values for each property that corresponds with an adaptation goal. These properties are usually quality properties of the system. In the service-based system example, the quality attributes are the estimated failure rate and estimated resource usage of an adaptation option that is expected to be achieved when the system is adapted according to that adaptation option.

Benefit estimator is a mechanism that is used to estimate the benefit of adaptation options. In the service-based system example, we may use an estimator that evaluates the utility of adaptation options by simulating the workflow of different adaptation options using the actual values associated with the different paths in the workflow.

Estimated benefit represents the estimated overall benefit of an adaptation option based on the estimated benefit attributes and the associated adaptation goals. In the service-based system example, the benefit may be represented as an utility that is determined as a sum of the weighted values of the estimated quality attributes: failure rate and resource usage.

Cost dimensions represent the different costs that are implied by adaptation of the managed system. In the service-based system example, we may consider the overhead that is implied by testing new concrete services that are included in the service workflow.

Cost metrics define measures to quantify the cost dimensions. For instance, testing new services requires resources that may depend on the service level agreements made with the service providers.

Cost estimator is a mechanism that is used to estimate the cost attributes of the adaptation options. In the service-based system example, the cost estimator may determine the expected resources required to test new services when selecting different adaptation options.

Estimated cost represents the estimated overall cost of an adaptation option based on the estimates for the different cost attributes and their relevance. In the service-based system, the estimated cost corresponds to the resources that are expected to be required to test new services of the selected adaptation option.

Benefit-cost analyser determines the desirability of an adaptation option. Computing desirability requires that the estimated benefit and estimated cost are expressed in a common metric and are scaled to be comparable. For the service-based system the desirability of adaptation options can be computed as value-for-cost that determines desirability as the scaled estimated benefit over the scaled estimated cost.

Estimated desirability expresses the degree that stakeholders prefer one adaptation option over the other options by comparing overall estimated benefit with overall estimated cost. For the

service-based system the value-for-cost can be used to represent the estimated desirability of adaptation options.

Risk attributes represent the different types of risk that are implied by adapting a managed system with a given adaptation option. Risk attributes can related to safety, environment, finances, etc. In the service-based system, the confidentiality of data and the health of patients are two important risk attributes.

Risk metrics define measures to quantify risk attributes. 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.

Risk estimator is a mechanism that is used to estimate the risk attributes of the adaptation options. In the service-based system example, the estimated risk may be determined based on the service level agreements with the providers of the selected services of the adaptation options, based on the likelihood/consequence matrices of the different risk attributes.

Estimated risk of an adaptation option represents the estimated overall risk of an adaptation option based on the estimates for the different risk attributes and the weights associated with them. In the service-based system example, the risk may be represented as a weighted sum of the two risk attributes related to data exposure and health of patients.

Decision-making mechanism is a mechanism that selects an adaptation option from the set of available options taking into account the estimated desirability and estimated risk of all available adaptation options. In the service-based system example, the decision-making mechanism may select the adaptation option that maximises the difference between weighted desirability and risk.

Selected adaption 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 service that the workflow needs to invoke; some of the current services may remain in use, others may need to be replaced by new selected services.