# Towards a Method for Decision Support in Multi-cloud Environments

Aida Omerovic
SINTEF ICT
Norway
Email: aida.omerovic@sintef.no

Victor Muntés-Mulero and Peter Matthews
CA Technologies
CA Labs Europe
Email: {victor.muntes,peter.matthews}@ca.com

Alexander Gunka
BOC Information Systems
Austria
Email: alexander.gunka@boc-eu.com

***Abstract*—Providers of cloud services as well as the cloud services themselves differ in the business models, functionality, quality of service, cost, value, etc. which makes the choice of a provider and a service difficult. Beyond that the complexity and lack of transparency with respect to cost and quality render the run-time adaptation and replacement of services almost impossible. This position paper presents main results of our recent efforts towards development of a decision support method (DSM) in multi-clouds. The DSM aims at taking into account risk, quality and cost aspects in order to assist a decision maker in choosing providers and services in a multi-cloud environment. We characterize the needs for the DSM in the multi-cloud context and propose an initial version of the process for the DSM. Based on the method proposed and the needs identified, we elaborate to what degree the current state of the art can be leveraged and what further multi-clouds-specific extensions are needed.**

***Keywords*—multi-cloud; decision support; risk assessment; quality prediction; cost prediction; architectural design; trade-off analysis; cloud service selection; cloud provider selection.**

## I. Introduction

The rapidly increasing number of cloud services and cloud service providers opens for new opportunities [1] in designing application and enterprise architectures. It also enables new business models and investments [2] [3] [4], new quality levels [5], as well as new capabilities. The services can be orchestrated and their compositions adapted even more dynamically than earlier. Availability of similar services from several providers opens for replaceability between services, or redundancy of services. As a result, the quality may improve and the risk of vendor lock-in will normally be reduced. However, there are also significant challenges [6] involved in realizing collaborations between clouds. One of the major challenges regarding cloud services and their providers is that they differ in the business models, functionality, quality of service, cost, value, etc. Another challenge is complexity and lack of transparency with respect to cost and quality. This makes the choice of a provider and a service difficult and the run-time adaptation and replacement of services almost impossible. When selecting the cloud services and the cloud providers, systematic support for identifying the candidate services and understanding the implications of choosing the different alternatives, is needed.

Decision support [7] for multi-cloud environments imposes several challenges compared to the traditional model-based decision support. Most notably, the dynamics of multi-cloud require light-weight processes and tools, the decision makers depend on easy-to-understand representations of the impacts of the decisions, the notion of cost is to a lower degree established in the existing approaches supporting the trade-off analysis of enterprise and software architectures, and a merge of the aspects of risk, cost and quality in a consolidated view imposes a new complexity as well as methodological challenges.

The specific objective of this paper is to establish the necessary baseline for a tool-supported decision support method (DSM) aimed at facilitating selection of cloud services and providers in a multi-cloud environment. In particular, we argue that risk, quality and cost are among the main three factors in such a selection process. To that end, we aim at providing a decision support which analyses the impacts of the possible decision alternatives in a multi-cloud environment with respect to those three factors. We believe that a trade-off analysis between risk, cost and quality based on a consolidated view of the three will provide a useful basis for a decision maker in assessing the possible choices through a cost-benefit analysis.

This position paper presents the main results of the recent efforts towards development of a DSM for multi-cloud environments. We characterize the needs for the DSM in the multi-cloud context and propose an initial version of the process for the DSM. Based on the method proposed, we elaborate on the suitability of both the method proposed and the state of the art for analyzing risks as well as for predicting quality and cost in the multi-cloud context.

The paper is organized as follows. Section 2 summarizes the state of the art regarding risk analysis, quality prediction, and cost analysis. Section 3 characterizes the needs for the DSM in the multi-cloud context. Section 4 proposes an initial process for the DSM. Section 5 discusses to what degree the state of the art can be leveraged within the DSM process proposed. Main conclusions are provided in Section 6.

## II. State of the art

The ISO 31000 standard for risk management comes with no specific techniques, modeling languages or recommended tools for how to conduct risk assessment in practice. However, most established risk management methods [8] [9] [10] [11] follow the ISO 31000 process, and provide such additional support. Common for these approaches is that they are designed to support risk management and risk documentation from the perspective of an organization and its policies. There is lack of support in the state of the art for extracting the risk picture that is relevant for specific external stakeholders, such as services consumers, and to present this picture in an intuitive and easily understandable way. There is also lack of an approach which combines cloud modeling and risk modeling. There exist many different approaches to service modeling [12] [13] [14] [15], focusing on expressing relevant elements and aspects of services, such as actors and components, roles, activities, interfaces and contracts. However, none of these have a risk-oriented view where stakeholders are represented as risk owners, and where the assets at stake are made explicit.

In a model-based decision making, the decisions are made based on a number of factors. The major ones include functional and non-functional properties, as well as cost and the added value. A trade-off between such factors is the basis for decision making. This trade-off is particularly complex between the non-functional factors, the variable parts of the architecture, and the cost of the selected solutions. The variability, as well as incomplete information or knowledge, are also sources of risk. Since functional requirements normally are less flexible and specified rather early, and since the added value is strongly related to the functional properties, the factors that are tunable and highly interrelated are risk, quality and cost. Therefore, in a model-based decision making, the decisions are based on a trade-off assessment between risk, quality and cost. The risk assessment, in turn, is based on information that is gathered about assets, entities, actors, etc. that are involved in the service event or action in question.

As a basis for the elicitation of the adequate quality characteristics, we may use the software product quality standard ISO/IEC 9126 [5]. The ISO 9126 defines quality as “the totality of features and characteristics of a software product that bear on its ability to satisfy stated and implied needs”. The ISO 9126 standard provides an established specification of decomposed quality notions with their qualitative and quantitative definitions. The standard defines a quality model for external and internal quality, and for quality in use. External quality is the totality of the characteristics of the software product from an external view when the software is executed. Internal quality is the totality of characteristics from an internal view and is used to specify properties of interim products. The characteristics of the internal and external quality model are functionality, reliability, usability, efficiency, maintainability and portability. These are in turn decomposed into a total of 34 sub-characteristics. Quality in use is the user’s view of the quality of the software product when it is used in a specific environment and a specific context of use. The quality in use characteristics are effectiveness, productivity, safety and satisfaction. There is also a further decomposition of all characteristics into the related metrics.

SMI [16] is a standardization effort from the Cloud Services Measurement Index Consortium (CSMIC) consisting of academic and industry organizations. The Service Measurement Index (SMI) uses a series of characteristics and measures to create an common means to compare different services from different suppliers. The characteristics are categorized as Usability, Performance, Agility, Security and Privacy, Financial, Assurance and Usability. Each of these characteristics has a number of measures that can be used to evaluate the risk in using a service. For example in the accountability category one of the measured attributes is Compliance and another is SLA verification both of which can be used to create a risk measure for the service and the provider. CSMIC is in negotiation with a number of large standardization organizations to develop a joint working group and specification.

According to Fenton and Neil [17], most prediction models use size and complexity metrics to predict defects. Others are based on testing data, the quality of the development process, or take a multivariate approach. The goal/question/-metric paradigm [18] [19] is a significant contribution to quality control and can be used for development of quality models and for the design of a measurement plan [20] [21]. To enable explicit risk and quality assessment, we make use of monitoring and measurement. Risk monitoring is a means to facilitate continuous risk assessment by the monitoring of relevant key indicators or metrics. An indicator can be defined as “something that provides a clue to a matter of larger significance or makes perceptible a trend or phenomenon that is not immediately detectable” [22]. To enable explicit risk and quality assessment, we make use of monitoring and measurement.

PREDIQT [23] is a tool supported method for model-based prediction of impacts of architectural design changes on system quality characteristics (performance, scalability, security, etc.). PREDIQT facilitates specification of quality characteristics and their indicators, aggregation of the indicators into functions for overall quality characteristic levels, and dependency analysis. The main objective of a PREDIQT-based analysis is prediction of system quality by identifying different quality aspects, evaluating each of these, and composing the results into an overall quality evaluation. This is useful, for example, for elicitation of quality requirements, evaluation of the quality characteristics of a system, run-time monitoring of quality relevant indicators, as well as verification of the overall quality characteristic fulfillment levels. PREDIQT makes use of models that capture the system design, the system quality notions, as well as the relations between them. An important aim of PREDIQT is to enable the right balance between practical usability of the models and the soundness of the predictions. The method is compatible with the ISO/IEC 9126 software quality standard, and has been successfully applied in real-life industrial settings [24] [25].

CORAS [8] is a tool-supported and model-driven approach to risk analysis that is based on the ISO 31000 risk management standard. Whereas alternative state-of-the-art approaches such as CRAMM [26] and OCTAVE [27] rely on text and tables, CORAS uses diagrams as an important means for communication, evaluation and assessment. Risk modeling is a technique for risk identification and assessment, and the state-of-the-art offers several tree-based and graph-based notations. Fault tree analysis [28] (FTA), event tree analysis [29] (ETA) and attack trees [30] are examples of the former and provide support for reasoning about the sources and consequences of unwanted incidents, as well as their likelihoods. Cause-consequence analysis [31] (CCA), Bayesian network [32] and Markov analysis [33] are examples of graph-based notations. CCA employs diagrams that combine the features of both fault trees and event trees, whereas the latter two serve as mathematical models for probabilistic and statistical calculations, respectively.

Approaches to quality assessment, risk analysis and security management provide support for decision making so as to ensure a required quality level while managing risks. However, while identifying and suggesting options and solutions, such as security mechanisms, the methods often lack techniques and tools for analyzing the associated cost and the return of investment in the identified solutions. Franqueira et al. [2] address this problem by proposing a method for handling security investment decisions achieved by so-called Real Option thinking. The method is partly based on Real Option Analysis [3] (ROA), which is a decision support technique in the area of capital investment by means of mathematical models to evaluate financial options. The method is supported by a security trade-off tool called SecInvest, which is implemented as a Bayesian network topology and supports decision makers in evaluating investment options and identifying the most suitable and cost-efficient ones. Other approaches to cost estimation in the setting of security investments are Net Present Value (NPV) [4], Return on Security Investment (ROSI) [34], Architecture Trade-Off Analysis Method (ATAM) [35], the Cost Benefit Analysis Method (CBAM) [7] and the Security Solution Design Trade-Off Analysis [36]. These and similar approaches can be understood as methods and techniques to facilitate so-called security economics.

## III. Characterization of needs

As a part of context establishment, we elicited quality aspects and risks which are specific to a multi-cloud environments. The elicitation was based on a comprehensive model of migration process. The model was used as a baseline and a checklist for understanding and decomposing the risk, quality and cost aspects. The exercise resulted in a high-level overview of main risks, as well as a model of decomposed quality characteristics which are specific to multi-clouds. The three overall characteristics identified are: interoperability, intercloud replaceability and security. In addition, cost of migration between multi-clouds was classified into cost of personnel, cost of time with two coexisting services, cost of compensation for uncertainty, and cost of hardware and other resources. Through these models, a common understanding of the main risk, quality and cost aspects in our context, was established. The initial experiences and results of the quality, cost and risk classification indicate that:

- Before eliciting the quality characteristics and risks of a multi-cloud based architecture, the context has to be thoroughly defined. Moreover, the architecture models of the target need to be established. This provides a common understanding of the scope and objectives, as well as the necessary frames for further modeling and decision making. For example, during the context establishment, a process model for migration was used as the foundation for eliciting the aspects and indicators related to quality, cost and risk.
- The decision support models should, once available, be able to take the proposed alternatives for architecture design (measures and treatments considered) and, based on each alternative, provide the resulting risk picture, predicted levels of fulfillment of the relevant quality characteristics, as well as the estimated costs. Thus, risk, quality characteristics and cost should be treated as separate concerns.
- Ideally, in order to accommodate for a cost-benefit analysis, the method should consider added value (or profit) in addition to cost. Minimizing cost and risks and maximizing quality levels is not necessarily a realistic goal. In fact, the benefits may arise from e.g. process improvement through the new architecture, improved or extended functionality, or similar. Thus the trade-offs between quality, risk and cost may vary significantly depending on the utility function and the risk attitude of the decision maker. In addition, the trade-off (or “selection criteria”) should take into account the need for balancing the cost with the added value beyond achieving the quality and risk relevant objectives.
- The method should be tool supported, and the tool should at least provide a diagram editor as well as an easy-to-understand presentation of the impacts of the decision alternatives on quality, risk and cost. The tool should also offer the interfaces needed for acquisition of the data needed for evaluation of the indicators, as well as the interfaces for the needed trace-link information.

## IV. Method for decision support for multi-cloud environments – a preliminary specification

The DSM for multi-cloud applications is a model-driven method consisting of three main artifacts: a process, a language and a tool. This section provides the initial specification of the DSM process and the actors involved. The DSM process consists of three overall phases, and each phase is decomposed into a set of sub-phases. The DSM process is undergone while developing, verifying and applying the comprehensive decision support models which include the aspects of architecture, risk, quality and cost. We assume the following four types of actors involved in the DSM process:

- Analyst: the analyst is an expert in the DSM and has the responsibility for leading and facilitating a DSM-based analysis. That is, the analyst coordinates the overall actors, collects the input for developing the decision support models, interacts with the overall actors during the model development and usage, makes sure that the necessary steps have been conducted within the resources allocated, and validates that the models have the needed quality and contents.
- Decision maker: the decision maker defines the scope and the objective of a DSM-based analysis. He/she will provide the instructions as to what parts of the architecture should be encompassed in the models, the expected validity of the models, the scope and kinds of the perspective changes/revisions of the architecture, etc. The decision maker will also be the main user of the decision models once they have been developed. He will therefore specify the decision alternatives in the decision models, and use the resulting impact estimates with respect to risk, cost and quality as an aid in the decision making. This actor is aware of the business model and strategy of the company. Hence, a decision maker may be a business expert as well, capable of making decisions based on his knowledge

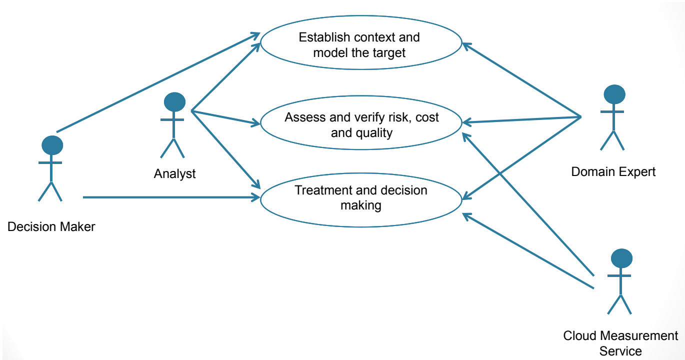


Fig. 1. The top level three phases and the actors involved in the DSM process

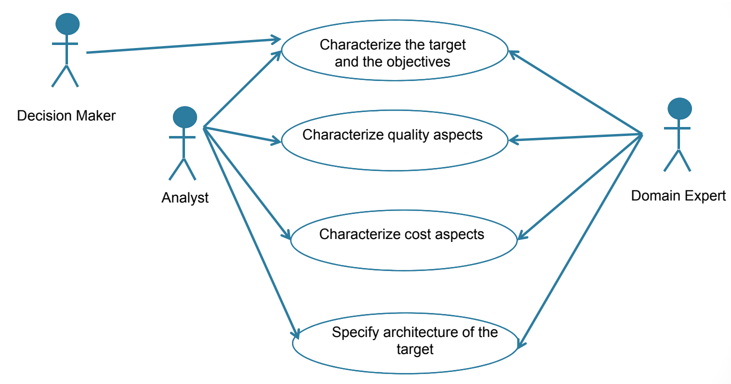


Fig. 2. Establish context and model the target phase decomposed

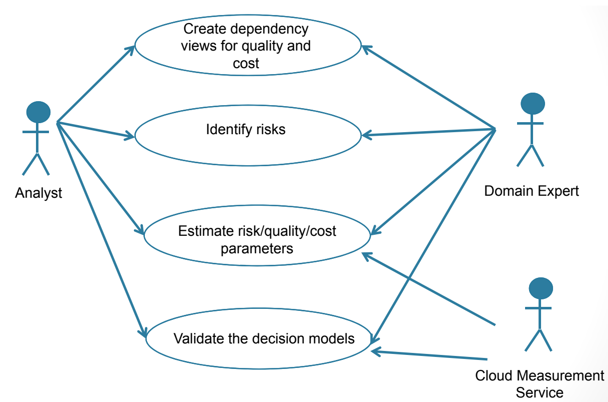


Fig. 3. Assess and verify risk, cost and quality phase decomposed

of the project budgets, allowable risks and the business processes being supported by the applications. Larger organizations may distinguish between a business expert who builds the requirements specification and a decision maker who selects services based on the specification. For simplicity, these two roles are in our case represented by the decision maker who has all the knowledge sufficient to take decisions.

- Domain expert: normally, a group of domain experts will be involved in a DSM-based analysis in relation to the development, validation and revision of the decision models. The domain experts will contribute by providing the thorough input regarding the current architecture, quality levels, dependencies and processes. The analyst will actively interact with the domain experts during all the three phases of the DSM process.
- Cloud measurement service: this is a (partially) automatized service for retrieval of the empirical data needed for estimating the parameters of the decision models. We assume that the parameters are estimated either based on the feeds from the cloud measurement service or based on expert judgments. A parameter may be estimated or measured either directly, or through estimation of a measurable indicator which then is aggregated and mapped to the decision model through a function. The dynamics of the indicators and the parameters as well as their relevance and uncertainty will be among the factors for determining whether the data acquisition should be automatic (e.g. real-time retrieval based on a monitoring environment) or manual, and how frequent it should be.

Figure 1 shows the overall three phases of the DSM process, as well as the actors involved. In the first phase, the context of the analysis is established. As a part of this, the scope is defined, the relevant risk, cost and quality notions are defined, and the architecture is modeled. In addition, the expected validity as well as perspective business models and architecture alternatives should be anticipated in order to cover the needed scope and level of detail in the target models. During the second phase, the decision models covering the risk, quality and cost aspects are instantiated with respect to target. As a part of this, the dependencies are modeled and the parameters (with the related indicators) are estimated. In addition, the models are validated through various kinds of triangulation, mainly based on the empirical input, logs, domain expert judgments, experience factories, etc. In the last phase, the decision models are applied by first specifying the decision alternatives, applying the alternatives on the models, and finally obtaining the resulting impact of the respective decisions on quality, risk and cost. The result is a consolidated view of the quality, risk and cost picture, provided each decision alternative.

Figure 2 shows the stages of the "establish context and model the target" - phase. First, the target and the objectives are characterized. Based on the initial input, the stakeholders involved deduce a high level characterization of the target architecture, its scope and the objectives of the DSM-based analysis, by formulating the system boundaries, system context (including the usage profile), system lifetime and the extent (nature and rate) of design changes expected. In the second stage, the quality aspects are characterized by specifying which quality characteristics are relevant for the target, and thereafter decomposing them down to indicators. A quantitative and a qualitative definition should be provided for all elements. Thirdly, a corresponding decomposition should be done for the cost aspects. In the last stage, the architecture is modeled with the detail level and within the frames specified during the characterization stage.

Figure 3 shows the stages of the "assess and verify risk, cost and quality" - phase. Firstly, the dependency views for

 ISBN: 978-1-61208-271-4 

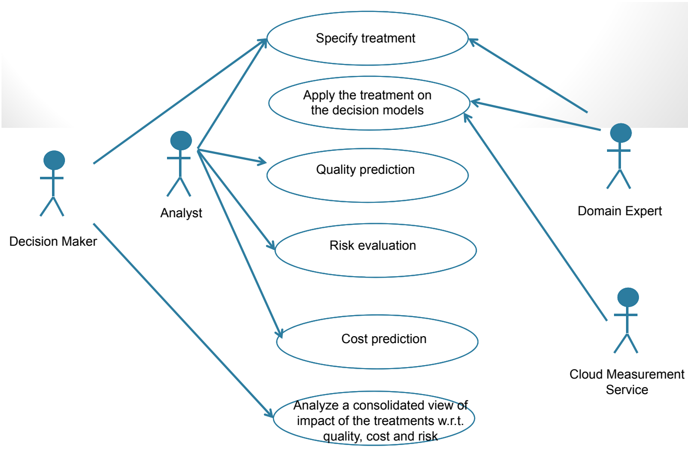


Fig. 4. Treatment and decision making phase decomposed

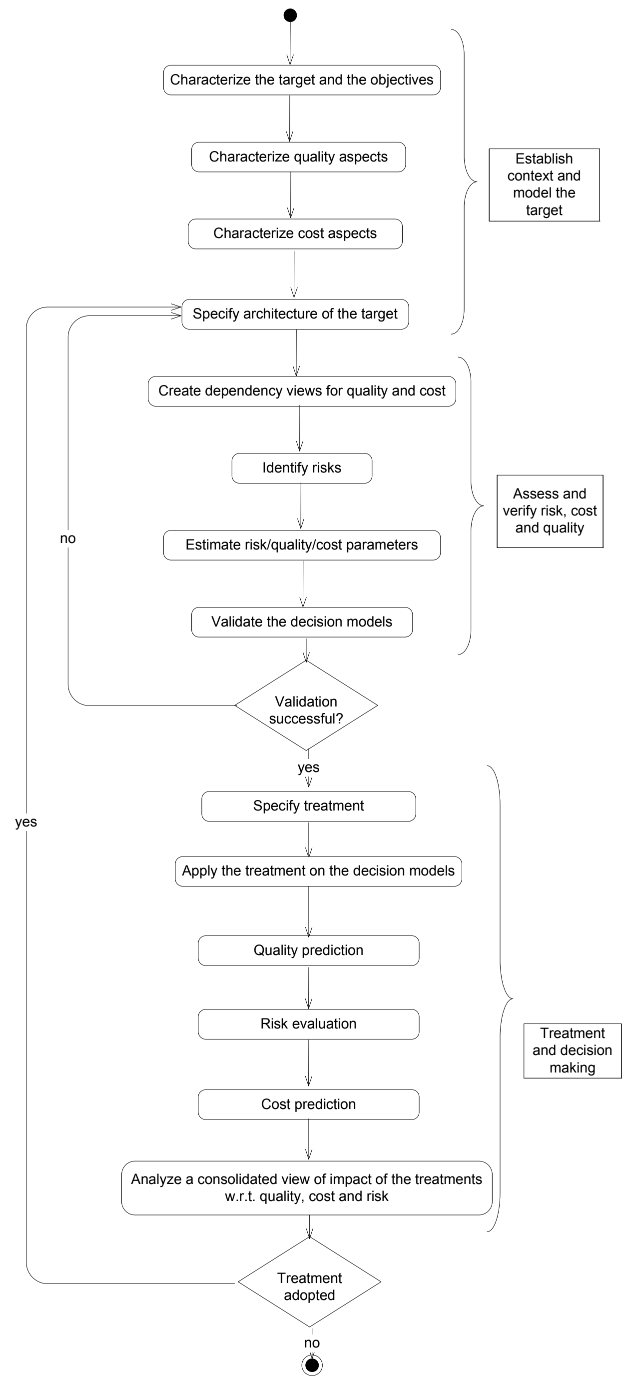


Fig. 5. The DSM process diagram with feedback loops

respectively quality and cost are developed. Secondly, assets and risks are identified in separate decision models ("threat diagrams"). The three types of the decision models (i.e. quality dependency views, cost dependency views and threat diagrams) are then annotated by the parameter values through evaluation of indicators or direct expert judgments on the prior parameters. Finally, triangulation is performed in order to validate the decision models. The models are approved once an acceptable level of uncertainty has been reached.

Figure 4 shows the stages of the "treatment and decision making" - phase. First, the respective decision alternatives are specified separately. Then, each alternative is applied on the decision models. The models and the respective calculus is used to propagate the impacts of each decision alternative on risk, quality and cost. Finally, a consolidated view of the impacts of the decision alternatives is presented to the decision maker.

Figure 5 shows an activity diagram with the entire DSM process, including the feedback loops. The right hand side of the figure indicates the phases presented in Figure 1. The activities are equivalent to the ones presented in relation to Figure 2, Figure 3 and Figure 4.

## V. Discussion

This section elaborates to what degree the existing PREDIQT and CORAS methods for for quality prediction and risk analysis, respectively, can serve as a baseline for our DSM in multi-clouds. The objective is to leverage the state of the art decision support, while extending it and adjusting to the special needs of the multi-clouds. Thus, the established methods, languages and tools can be reused with the well known properties and resources, while the efforts can be concentrated on the multi-cloud-specific extensions.

PREDIQT is a method (process, language, and tool support) for model-based prediction of system quality. The PREDIQT method produces and applies a multi-layer model structure, called prediction models, which represent system relevant quality concepts (through "Quality Model"), architectural design (through "Design Model"), and the dependencies between architectural design and quality (through "Dependency Views"). The Design Model diagrams are used to specify the architectural design of the target system and the changes whose effects on quality are to be predicted. The Quality Model diagrams are used to formalize the quality notions and define their interpretations. The values and the dependencies modeled through the Dependency Views (DVs) are based on the definitions provided by the Quality Model. The DVs express the interplay between the system architectural design and the quality characteristics. Once a change is specified on the Design Model diagrams, the affected parts of the DVs are

 ISBN: 978-1-61208-271-4 

identified, and the effects of the change on the quality values are automatically propagated at the appropriate parts of the DV.

CORAS is a method (process, language, and tool support) for conducting model-based security risk analysis. CORAS provides a customized language for threat and risk modeling, and comes with detailed guidelines explaining how the language should be used to capture and model relevant information during the various stages of the security analysis. The Unified Modeling Language (UML) is typically used to model the target of the analysis. For documenting intermediate results, and for presenting the overall conclusions we use special CORAS diagrams which are inspired by UML. The CORAS tool supports documenting, maintaining and reporting analysis results through risk modeling.

The DSM process is based on an attempt to merge the processes of CORAS and PREDIQT for a consolidated analysis of risk, quality and cost. Most of the stages of the DSM process can be found in CORAS and PREDIQT. The actors/stakeholders defined in the DSM are fully compliant with the ones defined by CORAS and PREDIQT. The types of the decision models proposed in the DSM are heavily based on the modeling notations, languages and tools of PREDIQT and CORAS, respectively. The approach to modeling of quality and cost aspects based on the DVs is a part of the PREDIQT method, while a language for risk modeling is provided by CORAS. The respective approaches to modeling in PREDIQT and CORAS are based on graphical modeling languages with defined propagation models. Both modeling approaches are developed with special focus on comprehensibility and expressiveness. In that manner, the models are accommodated for fulfilling real-life needs in terms of covering the representations needed while being rather intuitive so that non-experts should be able to relate to them in an industrial setting. The characterization of quality proposed in DSM is by PREDIQT addressed through the so called Quality Model. Both the Quality Model and the intended quality characterization in DSM are similar to the elicitation we have performed, which is briefly presented in Section 3.

The DSM process is to a high degree a superset of the processes of PREDIQT and CORAS. Moreover, the modeling approaches of PREDIQT and CORAS cover the concerns of quality and risk, as well as partially the concern of cost. Furthermore, the existing tools of CORAS and PREDIQT may be useful in the DSM context. Provided this baseline, we believe that utilization of the CORAS and PREDIQT methods including the processes, the languages and the tools, is worth a further evaluation in the DSM context. In particular, this means that case studies in multi-cloud environments should be performed in order to evaluate the feasibility of DSM, as well as the suitability of the relevant parts of PREDIQT and CORAS in a multi-cloud context.

## VI. Conclusion and Future Work

This position paper aims at establishing the necessary baseline for a DSM. The intended purpose of the DSM is to facilitate the selection of cloud services and providers in a multi-cloud environment. In particular, we argue that risk, quality and cost are among the main factors in such a selection process. We believe that a trade-off analysis between risk, cost and quality based on a consolidated view of the three will provide a useful basis for a decision maker in assessing the possible choices through a cost-benefit analysis.

Decision support for multi-cloud environments imposes however several challenges compared to the traditional model-based decision support. Most notably, the dynamics of multi-cloud require light-weight processes and tools, the decision makers depend on easy-to-understand representations of the impacts of the decisions, the notion of cost is to a lower degree established in the trade-off analysis of enterprise and software architectures, and a merge of the aspects of risk, cost and quality in a consolidated view imposes a new complexity as well as methodological challenges.

This paper presents the main results of our recent efforts towards the development of a DSM for multi-cloud environments. We characterize the needs for the DSM in the multi-cloud context and propose an initial version of the process for the DSM. Based on the experiences from CORAS and PREDIQT based analyses, and relying on the existing process descriptions and modeling approaches from CORAS and PREDIQT, we propose a comprehensive process for a DSM-based analysis, and present the roles of the actors/stakeholders involved. The DSM process consolidates the steps necessary towards development, verification and application of the decision support models. Based on the method proposed, we elaborate on the suitability of both the method proposed and the state of the art for analyzing risks as well as for predicting quality and cost in the multi-cloud context. We argue that many aspects of CORAS and PREDIQT, including the approaches to modeling (the modeling languages), the processes, and the respective tool support, should be well suited in the DSM context, i.e. in an analysis which merges the aspects of risk, quality and cost. However, in order to evaluate the feasibility of both the proposed DSM in general as well as the CORAS and PREDIQT methods in particular, in the multi-cloud context, realistic case studies should be performed and the proposed method adapted based on the experiences obtained.

Hence, the next steps in the development of decision support for multi-clouds should include case studies, evaluation and development of approaches to modeling (the modeling languages) for a consolidated model-based risk analysis, quality prediction and cost analysis. Moreover, the method should offer an easy-to-understand visualization of the impacts of the decision alternatives on quality, cost and risk. We also aim at refining the method and the tool requirements for DSM, as well as providing a prototype tool which will facilitate a DSM-based analysis.

## Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 318484 (MODAClouds).

## References

[1] R. Buyya, "Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility," in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid*. IEEE Computer Society, 2009.

[2] V. N. L. Franqueira, S. H. Houmb, and M. Daneva, "Using Real Option Thinking to Improve Decision Making in Security Investment," in *5th International Symposium on Information Security, LNCS 6426*. Springer, 2010, pp. 619–638.

[3] M. Amram and N. Kulatilaka, *Real Options: Managing Strategic Investment in an Uncertain World*. Harvard Business School Press, Cambridge, Massachusetts, 1999.

[4] M. Daneva, "Applying Real Options Thinking to Information Security in Networked Organizations. CTIT Report TR-CTIT-06-11," University of Twente, Tech. Rep., 2006.

[5] *ISO/IEC 9126 – Software engineering – Product quality – Part 1-4*, International Organization for Standardization/International Electrotechnical Commission, 2001-2004.

[6] M. Singhal, S. Chandrasekhar, G. Tingjian, R. Sandhu, R. Krishnan, A. Gail-Joon, and E. Bertino, "Collaboration in Multicloud computing Environments: Framework and Security Issues," *Computer*, vol. 46, no. 2, pp. 76–84, 2013.

[7] R. Kazman, J. Asundi, and M. Klein, "Making Architecture Design Decisions: An Economic Approach. Technical report CMU/SEI-2002-TR-035," Carnegie Mellon, Tech. Rep., 2002.

[8] M. S. Lund, B. Solhaug, and K. Stølen, *Model-Driven Risk Analysis - The CORAS Approach*. Springer, 2011.

[9] Siemens, "CRAMM - The Total Information Security Toolkit," March 2004, accessed: January 30, 2013. [Online]. Available: http://www.cramm.com

[10] C. J. Alberts and A. J. Dorofee, "OCTAVE Criteria. Technical Report CMU/SEI-2001-TR-016," CERT, Tech. Rep., 2001.

[11] T. Peltier, *Information Security Risk Analysis, 3rd edn.* Auerbach Publications, 2010.

[12] R. Chinnici, J. J. Moreau, A. Ryman, and S. Weerawarana, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation," June 2007, accessed: January, 2013. [Online]. Available: http://www.w3.org/TR/wsdl20

[13] J. Farrell and H. Lausen, "Semantic Annotations for WSDL and XML Schema. W3C Recommendation," August 2007, accessed: January, 2013. [Online]. Available: http://www.w3.org/TR/sawsdl

[14] "Service Oriented Architecture Modeling Language (SoaML) Specification, Version 1.0," Object Management Group, Tech. Rep., 2012.

[15] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, and R. Metz, "Reference Model for Service Oriented Architecture 1.0. ," OASIS, Tech. Rep., 2006.

[16] Cloud Services Measurement Index Consortium, "CSMIC," accessed: January 2013. [Online]. Available: http://csmic.org

[17] N. Fenton and M. Neil, "A Critique of Software Defect Prediction Models," *IEEE Transactions on Software Engineering*, vol. 25, pp. 675–689, 1999.

[18] V. R. Basili, "Software Modeling and Measurement: The Goal/Question/Metric Paradigm, Technical Report TR-92-96," University of Maryland, Tech. Rep., 1992.

[19] V. Basili, G. Caldiera, and H. Rombach, *The Goal Question Metric Approach*. Encyclopedia of Software Engineering, 1994.

[20] N. E. Fenton and S. L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*. PWS Publishing Co., 1998.

[21] C. Ebert, R. Dumke, M. Bundschuh, A. Schmietendorf, and R. Dumke, *Best Practices in Software Measurement*. Springer Verlag, 2004.

[22] A. Hammond, A. Adriaanse, E. Rodenburg, D. Bryant, and R. Woodward, "Environmental Indicators: A Systematic Approach to Measuring and Reporting on Environmental Policy Performance in the Context of Sustainable Development," World Resources Institute, Tech. Rep., 1995.

[23] A. Omerovic, *PREDIQT: A Method for Model-based Prediction of Impacts of Architectural Design Changes on System Quality. PhD thesis*. University of Oslo, 2012.

[24] A. Omerovic, A. Andresen, H. Grindheim, P. Myrseth, A. Refsdal, K. Stølen, and J. Ølnes, "A Feasibility Study in Model-based Prediction of Changes on System Quality. Technical report A13339," SINTEF ICT, Tech. Rep., 2010.

[25] A. Omerovic, B. Solhaug, and K. Stølen, "Assessing Practical Usefulness and Performance of the PREDIQT Method: An industrial case study," *Information and Software Technology*, vol. 54, no. 12, pp. 1377–1395, 2012.

[26] B. Barber and J. Davey, "The Use of the CCTA Risk Analysis and Management Methodology CRAMM in Health Information Systems," in *7th International Congress on Medical Informatics*, 1992, pp. 1589–1593.

[27] C. J. Alberts and J. Davey, "OCTAVE Criteria Version 2.0. Technical report CMU/SEI-2001-TR-016," Carnegie Mellon University, Tech. Rep., 2004.

[28] "IEC 61025 Fault Tree Analysis (FTA)," International Electrotechnical Commission, Tech. Rep., 1997.

[29] "IEC 60300-3-9 Dependability Management - Part 3: Application guide - Section 9: Risk analysis of technological systems - Event Tree Analysis (ETA)," International Electrotechnical Commission, Tech. Rep., 1995.

[30] B. Schneier, "Attack Trees: Modeling security threats," *Dr. Dobb's Journal*, vol. 24, no. 12, pp. 21–29, 1999.

[31] D. S. Nielsen, "The Cause/Consequence Diagram Method as Basis for Quantitative Accident Analysis. Technical report RISO-M-1374," Danish Atomic Energy Commission, Tech. Rep., 1971.

[32] I. Ben-Gal, *Bayesian Networks. In F. Ruggeri, R. S. Kenett, F. W. Faltin (eds.): Encyclopedia of Statistics in Quality and Reliability*. John Wiley & Sons, 2007.

[33] R. A. Howard, *Dynamic Probabilistic Systems. Volume I: Markov Models*. John Wiley & Sons, 1971.

[34] W. Sonnenreich, J. Albanese, and B. Stout, "Return on Security Investment (ROSI)-A Practical Quantitative Model," *Journal of Research and Practice in Information Technology*, vol. 38, no. 1, pp. 45–56, 2006.

[35] R. Kazman, M. Klein, and P. Clements, "ATAM: Method for Architecture Evaluation. Technical report CMU/SEI-2000-TR-004," Carnegie Mellon, Tech. Rep., 2000.

[36] S. H. Houmb, G. Georg, R. France, J. Bieman, and J. Jürjens, "Cost-benefit Trade-off Analysis Using BBN for Aspect-oriented Risk-driven Development," in *10th International Conference on Engineering of Complex Computer Systems*. IEEE Computer Society, 2005, pp. 195–204.