# Towards a Cloud Architectural Decision Framework using Case-based Reasoning and Rule-based Reasoning

**Kent Ramchand**
School of Computer Science and Software Engineering,
Swinburne University of Technology, Melbourne, Australia
Email: kramchand@swin.edu.au

**Mohan Baruwal Chhetri**
School of Computer Science and Software Engineering,
Swinburne University of Technology, Melbourne, Australia
Email: mchhetri@swin.edu.au

**Ryszard Kowalczyk**
School of Computer Science and Software Engineering,
Swinburne University of Technology, Melbourne, Australia
Email: rkowalczyk@swin.edu.au
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
Email: rkowalcz@ibspan.waw.pl

## Abstract

Correct decision-making about the cloud platform architecture is crucial for the success of any cloud migration project; bad decisions can lead to undesirable consequences including project delays, budget overruns, application instability, below-par performance and creation of technical debt. Rule-Based Reasoning (RBR), a popular approach for solving clearly defined problems, can be used for cloud platform recommendation if a comprehensive set of requirements are available. However, the responsibility of decision-making is increasingly moving away from the hands of the technical subject matter experts, and into the hands of the business sponsors. Therefore, in this paper, we propose combining Case-Based Reasoning (CBR) with RBR to assist business sponsors in making strategic decisions between public, private and hybrid cloud with a high level of confidence even at the initial stages of the project.

**Keywords:** Cloud Migration, TCO Calculator, Decision Support Tool, Case-Based Reasoning, Rule-Based Reasoning.

# 1 Introduction

The Australian Government, through its National Cloud Computing Strategy (Australian Government, 2013) strongly encourages cloud uptake by both Government agencies and businesses alike, to boost productivity, innovation and business agility across the digital economy. Its Secure Cloud Strategy (Australian Government, 2017) states that government agencies should use the cloud as much as possible, and more importantly *use the public cloud services as default*, citing that it provides fast and competitive options. Yet, despite the push for cloud adoption, and its obvious benefits, there are several challenges associated with cloud migration (Gholami et. al., 2017) that need to be addressed. Any cloud migration decision process has to incorporate several aspects including requirements gathering, identifying and understanding constraints, tracing constraints to architectural building blocks, and identifying infrastructure components (Pahl, 2013). Failure to carefully consider these aspects and selecting the public cloud based on implied assumptions can lead to undesirable consequences including project delays, budget overrun, below-par performance, application instability, poor customer experience, and creation of technical debt.

We have previously developed a cloud decision framework (Ramchand et al. 2017; Ramchand et al. 2018) that uses Rule-Based Reasoning (RBR) to recommend cloud platform architectures based on an extensible set of functional, non-functional, compliance and regulatory requirements. It also provides a Total Cost of Ownership (TCO) calculator for the financial viability assessment of the technical recommendation, and supports iterative decision-making until a best *'fit for purpose'* cloud solution can be found that is both technically and financially viable. The framework requires a detailed set of decision criteria as input in order to make technical recommendations with a high degree of confidence, although it also supports a streamlined approach for decision-making with a reasonable level of confidence (Ramchand et. al, 2017). However, the growing prevalence of agile methodologies is making it important for organizations to identify a suitable cloud platform architecture, with a high level of confidence, early on in the project lifecycle (Younaset al. 2016); this responsibility of strategic decision-making is increasingly moving away from the hands of the technical subject matter experts, and into the hands of the business sponsors, who typically do not have access to enough information to provide a sufficiently complete set of requirements.

This issue can be addressed by using the Artificial Intelligence paradigm of Case-Based Reasoning (CBR) - a problem solving and knowledge reuse technique that seeks to solve new problems by reusing information and knowledge from similar situations in the past (Aamodt et al. 1995; Marling et al. 2002). CBR comprises the following steps: (1) analyse the new case (or problem) at hand; (2) based on the analysis, identify and retrieve relevant past cases from the case base; (3) rank the retrieved cases according to their relevance to the new case based on some "*similarity metric*"; (4) select one or more "*most similar*" cases to use for solving the new case; (5) create a solution to the new case based on the selected cases' solutions; (6) test and explore the created solution; and (7) if appropriate, add the new case and its solution to the case base for future use. There are three key benefits of using CBR for requirements elicitation: (a) it supports evidence-based decision-making, instead of relying on implied assumptions, (b) it supports explainable decision-making, unlike other black-box algorithms such as neural networks, and (c) it enables learning from experience. Learning occurs as a by-product of problem solving since the experience gained in successfully (or unsuccessfully) solving a problem can be used in the future to solve similar problems.

Therefore, in this paper, we propose an extension to our cloud decision framework, in which CBR is used in the requirements elicitation phase to assist business sponsors in identifying and prioritising a full set of requirements, following which RBR is used to recommend an appropriate cloud platform architecture.

The rest of the paper is organized as follows. Section 2 presents related work on cloud decision frameworks that use Case-Based Reasoning. Section 3 presents our proposed extension that combines CBR with RBR to enable better decision-making. Section 4 provides a simple illustrative scenario to illustrate applicability and coverage of concepts. The sample scenario underpins the motivation and need for technical decision support and illustrates how a combination of RBR and CBR can assist with this. Section 5 concludes the paper by providing a summary of the completed work and identifying areas of future work.

# 2 Related Research

Requirements elicitation is not straightforward and requirements engineering research has recognized elicitation incompleteness as an important issue (Burnay et al. 2015; Daramola, O et al. 2012).

Requirements incompleteness may occur when information remains implicit – either due to tacit knowledge, implicit requirements or implicit assumptions (Christel et al. 1992; Hickey et al. 1994), or if information remains unknown. It can also occur when the stakeholders cannot decide what it is to be built when interfacing with people or machines (Brooks et al, 1987). Existing tooling for requirements elicitation such as Requisite Pro[1] and Doors[2] do not assist in solving this problem (Daramola et al. 2012) and the implications of incomplete requirements are severe and several as highlighted in Section 1.

Case Based Reasoning (CBR) has been used previously in the Cloud platform decision process (Alhammadi et al. 2015; Soltani et al. 2016). In Soltani et al. (2016), CBR is used to recommend a cloud platform and automate the process from business requirements through to provisioning of resources in public cloud. The authors use a combination of application business and non-functional requirements to drive a process of comparison with the case history to measure a 'similarity' threshold score representative of precision, that is, a measurement of the usefulness of the case. It does not use 'closeness' as part of the comparison. If there is a match, it returns the most appropriate set of resources in the IaaS platform and identifies the most cost effective deployment back to the user. Similarly, in (Alhammadi et al. 2015), CBR is used with Analytical Hierarchical Process (AHP) to support decision making; CBR has five categories of cases (technical, organisational, security, economic and regularity) with different weighting to determine similarity of the case with those in the case history to assess its usefulness. AHP is used to calculate these weights, following which the result of the new case is compared with those of the retrieved cases.

Alternatively, in (MuBbacher 1992) the CBR system represents and organises requirements with the help of commonalities and variabilities, and retrieves requirements through similarity based retrieval options. The approach advocated is to use Requirements Traceability Matrix and an effort database as the model to compare requirements similarity followed by a requirements engineer to complete the requirements, which is not required in our framework. It then attempts to capture estimates around the SDLC in the effort database and assumes an organisation is at CMM Level 2 (meaning projects and plans are in place to use repeatable processes and work products in an enterprise) (CMMI Institute, 2018).

CBR has been widely used in other domains such as software development projects using CBR and Object Oriented design patterns for Service Oriented Architectures (Rodriguez et al. 2018), construction projects using CBR with a case history of family home project costs (Ji 2011), and for on-line course production (He 2014), utilising CBR and a Work Breakdown Structure (WBS) as a means of deriving estimates.

The main contribution of our research is to use CBR to address the problem of elicitation incompleteness. If the key decision-maker in the cloud migration project is unable to provide a comprehensive set of requirements, CBR can assist with requirements completion by utilising the knowledge gained from past completed cloud migration projects. CBR enables the automation of the measurement of the similarity and closeness of the new case with candidates from the case history subsequently improving the quality of decision-making. Productivity is gained through automating what would otherwise be a manual process for the comparison and determination of the closest cases. Furthermore, overlaying the actual results with the case history enables optimised decision making. We provide two approaches; one where we automate the process to populate the remaining criteria, provide a recommendation and conduct the feasibility assessment; and another, where a user can be provided suggested classifications for each of the remaining criteria, one at a time.

To the best of our knowledge, our approach is the first to use CBR to assist decision makers with incomplete requirements to obtain a comprehensive set of decision criteria based on which a cloud platform architectural decision can be made.

# 3 Cloud Decision Framework with Case Based Reasoning

We have previously developed a cloud decision framework (Ramchand et al. 2018) that uses RBR to make a cloud platform recommendation based on a detailed decision criteria set (see Figure 1 below). Under normal circumstances, the RBR based decision framework requires a business sponsor to identify a comprehensive criteria set with classifications to have a cloud platform recommendation provided.

[1] https://www.ibm.com/support/knowledgecenter/en/SSSHCT_7.1.0/com.ibm.reqpro.help/get_start/c_product_overview.html

[2] https://www.ibm.com/us-en/marketplace/rational-doors

The shortcoming of this approach is that business sponsors are increasingly having to provide the criteria for cloud platform selection and are unable to provide a full set of requirements early in the initiative.

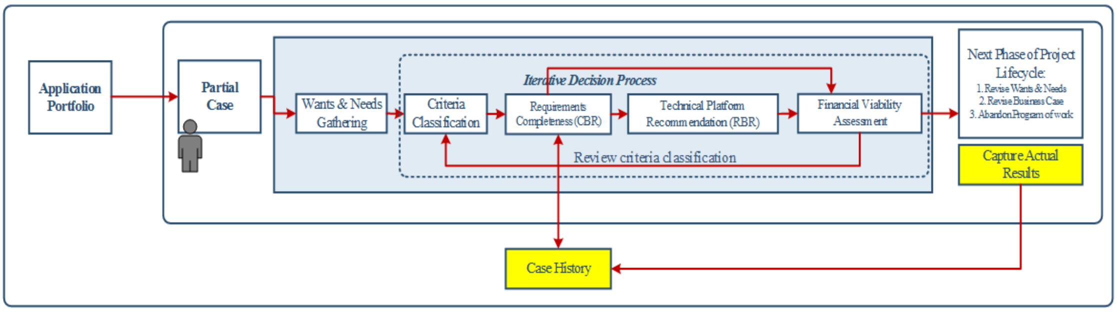
Application Portfolio
Partial Case
Iterative Decision Process
Wants & Needs Gathering
Criteria Classification
Requirements Completeness (CBR)
Technical Platform Recommendation (RBR)
Financial Viability Assessment
Review criteria classification
Next Phase of Project Lifecycle:
1. Revise Wants & Needs
2. Revise Business Case
3. Abandon Program of work
Capture Actual Results
Case History

**Figure 0 : Cloud Decision Framework using RBR and CBR**

To overcome this shortcoming, we extend the framework by introducing the CBR-based requirements elicitation step. The introduction of CBR provides business stakeholders with an avenue to obtain insights from prior technical recommendations and subsequent outcomes from the case history. The CBR approach is not necessary if all the criteria are known upfront by the user of the Decision Support Tool, as the tool will produce the same result every time. However, if the user is aware of a subset of criteria, the CBR approach assists with requirements elaboration as illustrated in Figure 2. CBR will assist with determining similarity and closeness through the strength of matching with cases.

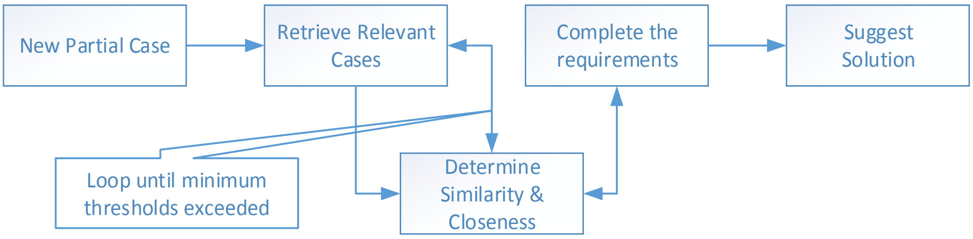
New Partial Case
Retrieve Relevant Cases
Complete the requirements
Suggest Solution
Loop until minimum thresholds exceeded
Determine Similarity & Closeness

**Figure 2: Requirements Completion using Case Based Reasoning**

## Building a case history

Access to a sufficiently large case base is necessary for leveraging CBR for requirements completion. In our framework, we assume that the case history is initially built from cases where RBR is used to make the technical cloud platform recommendation and both the recommended and actual solutions are recorded as cases using an appropriate representation. Figure 3 below shows an example case model representation using attribute values. It should be noted that our ongoing work is focussed on building an appropriate model for the variety of cases required for decision making. Once there are sufficient cases in the case history, the cases can be used for requirements completion. Each time an user of the system provides an incomplete set of requirements, CBR can be used to identify and retrieve the 'best fit' historical case/s for completing the remaining criteria classifications, following which RBR can be used to determine an appropriate cloud platform recommendation and conduct a financial viability assessment.

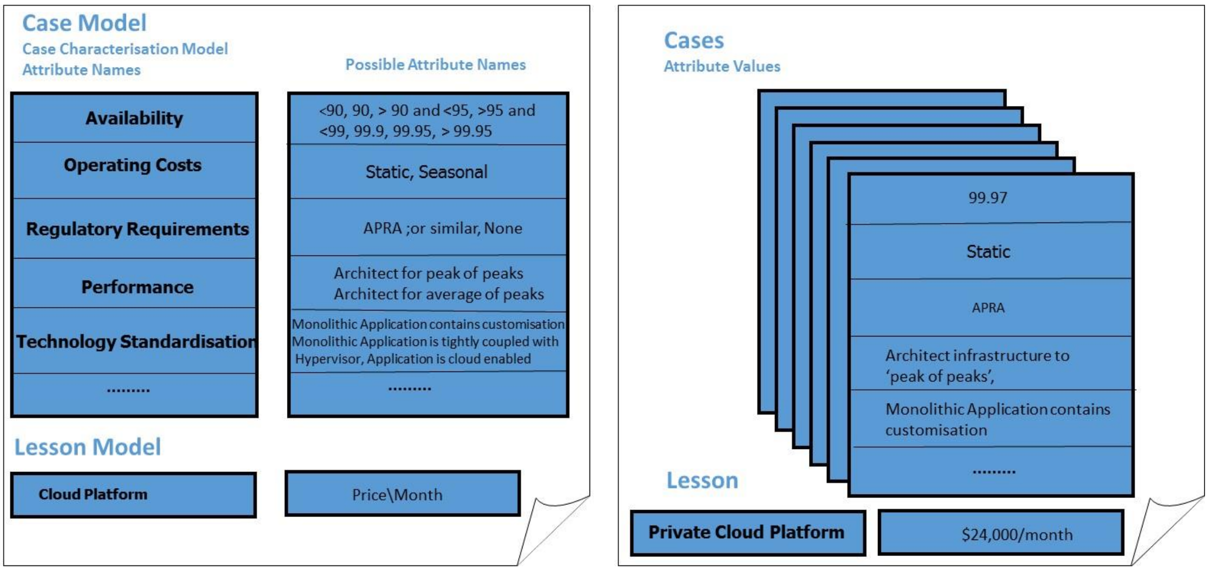
Case Model
Case Characterisation Model
Attribute Names
Possible Attribute Names
Availability
<90, 90, > 90 and <95, >95 and <99, 99.9, 99.95, > 99.95
Operating Costs
Static, Seasonal
Regulatory Requirements
APRA ;or similar, None
Performance
Architect for peak of peaks
Architect for average of peaks
Technology Standardisation
Monolithic Application contains customisation
Monolithic Application is tightly coupled with Hypervisor, Application is cloud enabled
.........
.........
Lesson Model
Cloud Platform
Price\Month
Cases
Attribute Values
99.97
Static
APRA
Architect infrastructure to 'peak of peaks',
Monolithic Application contains customisation
.........
Lesson
Private Cloud Platform
$24,000/month

**Figure 3 Example Case Model Representation**

There are a number of different techniques that can be used for similarity matching (Aamodt and Plaza 1995; MuBbacher 1992; Alhammadi et al. 2015). In our model, the number of criteria classification matches between the historical and the new case determines the degree of similarity. Similarly, the closeness is determined by the degree of alignment between the priority of criteria for the new and historical cases. A minimum threshold is set for each comparison based on the desired confidence level.

### Updating the case history

The technical recommendation of our cloud decision framework is not a binding decision. The business sponsors may choose to follow it as-is or make their own minor/major adjustments which can result in a final outcome that is different to that recommended by the system. Therefore, our framework will also support the capturing of the actual outcome and associated costs when the technical recommendation is actually implemented. Capturing this information optimises the decision making process for new cases. Essentially, the user will capture the actual costs and compare them with what was estimated. Building this case history for each business SME's own purposes will likely increase the accuracy of future decision making in their cloud environments.

## 4 An Illustrative Scenario

As a simple illustrative example we use the Contact Centre scenario to illustrate how CBR can be used for requirements elicitation. In this scenario, an enterprise reaches a point in time where its Contact Centre (application) infrastructure reaches its end of life and a need arises to consider alternative compute, storage and network infrastructure options. The scenario includes both the new case and historical case/s with associated priorities (p1 – new partial case, p2 – historical case) and classifications as shown in Table 1.

Prior to analysing the classifications from the new case with the case history, the user specifies a threshold for similarity matching. The threshold is a configurable parameter, whose value can be set based upon the level of confidence preferred by user. As an example, a threshold of 50% is used in our scenario. Having surpassed the 50% threshold of matches at a classification level (matches in italics), the 'closeness' is measured through analysing the priorities. The priorities match is also greater than 50% in this case, hence this historical case is used to elaborate the remaining requirements. The benefit of having this match is that the user is better placed than having no reference point at all to guide their decision making. A key benefit of using CBR for requirements completion is that the recommended decision is explainable as it is evidence based and traceable to attributes of the case.

| **Scenario 1** | *Avaya Contact Centre* |
|---|---|

| **Cloud Decision Criteria** | *Criteria* | *Classification* | *P(1)* | *P(2)* | *From Case History* |
|---|---|---|---|---|---|
| | Availability | Required | 1 | 1 | Required |
| | Business Service Availability | Required | | | Required |
| | Long running business process | Required | | | Optional |
| | Application Usage | Optional | | | Optional |
| | Regulatory requirements | Required | 2 | 2 | Optional |
| | Operating Costs | Optional | 3 | 3 | Optional |
| | Performance | Optional | 4 | 5 | Optional |
| | Application architecture | Required | | | Optional |
| | Application constraints | Required | | | Optional |
| | Security | Required | | | Optional |
| | Data Security Classification | Optional | | | Optional |
| | Network Global Load Balancing | Optional | | | Optional |
| | Connectivity to private MPLS network or internet VPN | Optional | | | Optional |
| | Hypervisor | Required | | | Required |
| | Enterprise Control | Required | | | Optional |
| | Data Classification | Required | | | Required |
| | Technology Standardisation | Required | 2 | 4 | Optional |

**Table 1 : Avaya Contact Centre Scenario**

# 5 Conclusion

In this paper, we have presented our work in progress on using CBR to address the problem of requirements incompleteness in the early stages of a cloud migration project. With the cloud platform decision increasingly moving into the hands of the business sponsors, it is likely that not all criteria that influence the decision are known upfront. To address this issue, in our approach, CBR is used to supplement partial requirements specified by the business sponsors with default information from similar past cases to generate a comprehensive set of requirements for a migration initiative. RBR is then used to recommend the most appropriate cloud platform architecture based on these requirements. Our ongoing research focus is on building an appropriate model for case representation followed by extensive evaluation of the proposed approach. As future work, we also intend to explore combining CBR with RBR for technical cloud platform recommendation.

# 6 References

Aamodt, A., & Plaza, E. 1994. “Case-based reasoning: Foundational issues, methodological variations, and system approaches”. *AI communications*, 7(1), pp 39-59.

Agarwal, R., & Karahanna, E. 2000. “Time flies when you're having fun: Cognitive absorption and beliefs about information technology usage”. *MIS quarterly*, pp 665-694.

Alhammadi, A., Stanler, C., Eardley, A. 2015. “A Knowledge based Decision Making Tool to Support Cloud Migration Decision Making”, 17th International Conference on Enterprise Information Systems (ICEIS-2015), pp 637-643

Australian Government. 2013. “Australian Cloud Computing Strategy”. [online] Available at: https://www.communications.gov.au/sites/g/files/net301/f/National_Cloud_Computing_Strategy.pdf [Accessed 20 Jul. 2018]

Australian Government. 2017. “Secure Cloud Strategy”. [online] Available at: https://www.dta.gov.au/what-we-do/policies-and-programs/secure-cloud

Brooks, Frederick P. Jr, “No Silver Bullet: Essence and Accidents of Software Engineering,” *Computer*, 20(4):10-19, 1987

C. Burnay, I. Jureta and S. Faulkner, "Towards a Model of Topic Relevance during requirements elicitation - Preliminary results," *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, Athens, 2015, pp. 151-158.

Champigneulle, P., Williams, J. and Echternach Jr., D. 2017. “The Hybrid Cloud Survival Kit”. [online] Available at: https://advisory.kpmg.us/content/dam/advisory/en/pdfs/hybrid-cloud-survival-kit.pdf [Accessed 20 Jul. 2018]

Christel, M.G., Kang, K. C. “Issues in requirements elicitation,” Technical Report CMU/SEI-92-TR-12 ESC-TR-92-012, 1992.

CMMI Institute. 2018. “CMMI Levels of Capability and Performance”. [online] Available at: https://cmmiinstitute.com/learning/appraisals/levels

Daramola, O,. Moser, T,. Sindre, G,. Biffl, S. 2012, “Managing Implicit Requirements Using Semantic Case-Based Reasoning”. Norwegian University of Science and Technology

Gholami, M. F., Daneshgar, F., Beydoun, G. & Rabhi, F. 2017. “Challenges in migrating legacy software systems to the cloud—an empirical study”. *Information Systems*, 67, pp 100-113.

He, W. 2014. “A framework of combining case - based reasoning with a work breakdown structure for estimating the cost of online course production projects”. British Journal of Educational Technology, 45(4), pp 595-605.

Ji, S. H., Park, M., & Lee, H. S. 2011. “Cost estimation model for building projects using case-based reasoning”. Canadian Journal of Civil Engineering, 38(5), pp 570-581.

Bareiss, E. R., Porter, B. W., & Wier, C. C. 1990. “Protos: An exemplar-based learning apprentice”. In Machine Learning, Volume III, pp. 112-127.

Marling, C., Sqalli, M., Rissland, E., Muñoz-Avila, H., & Aha, D. 2002. “Case-Based Reasoning Integrations”. AI Magazine, 23(1), 69.

MuBbacher, G. 1998. “Combining CBR and community analysis for software requirements reuse”, Simon Fraser University

Ramchand, K., Chhetri, M. B., & Kowalczyk, R. 2017. “Towards a Flexible Cloud Architectural Decision Framework for Diverse Application Architectures”. In The 28th Australasian Conference on Information Systems, (ACIS2017), WestPoint Tasmania, Hobart, Tasmania, 4-6 December 2017.

Ramchand, K., Chhetri, M. B., & Kowalczyk, R. 2018. “Towards A Comprehensive Cloud Decision Framework with Financial Viability Assessment”. In Proceedings of the 22nd Pacific Asia Conference on Information Systems (PACIS 2018), 26-30 June 2018, Yokohama, Japan (p. 2744).

Rodriguez, G., Andres Diaz-Pace, Alvaro, S. 2018. “A case-based reasoning approach to reuse quality-driven designs in service-oriented architectures”. Information Systems 77 pp 167-189

S. Soltani, K. Elgazzar and P. Martin, "QuARAM Service Recommender: A Platform for IaaS Service Selection," *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, Shanghai, 2016, pp. 422-425.

Younas, M., Ghani, I., Jawawi, D. N., and Khan, M. M. 2016. “A Framework for Agile Development in Cloud Computing Environment”. Journal of Internet Computing and Services, 17(5), 67-74.

## Copyright