

Cloud Adoption: Prioritizing Obstacles and Obstacles Resolution Tactics Using AHP

Shehnila Zardari
School of Computer Science
University of Birmingham
Birmingham B15 2TT, United
Kingdom
sxz845@cs.bham.ac.uk

Rami Bahsoon
School of Computer Science
University of Birmingham
Birmingham B15 2TT, United
Kingdom
r.bahsoon@cs.bham.ac.uk

Aniko Ekárt
School of Engineering and
Applied Science
Aston University, B4 7ET
Birmingham, United Kingdom
a.ekart@aston.ac.uk

ABSTRACT

The enormous potential of cloud computing for improved and cost-effective service has generated unprecedented interest in its adoption. However, a potential cloud user faces numerous risks regarding service requirements, cost implications of failure and uncertainty about cloud providers' ability to meet service level agreements. These risks hinder the adoption of cloud. We extend the work on goal-oriented requirements engineering (GORE) and obstacles for informing the adoption process. We argue that obstacles prioritisation and their resolution is core to mitigating risks in the adoption process. We propose a novel systematic method for prioritising obstacles and their resolution tactics using Analytical Hierarchy Process (AHP). We provide an example to demonstrate the applicability and effectiveness of the approach. To assess the AHP choice of the resolution tactics we support the method by stability and sensitivity analysis.

Categories and Subject Descriptors

D.2 [Software Engineering]: Requirements/Specifications—*Obstacle Resolution, Obstacle Prioritization, Cloud Computing*

General Terms

Management, Legal Aspects

Keywords

Goal-Oriented Requirements Engineering for Cloud, AHP for Obstacle Prioritization, Obstacle Resolution in Cloud Adoption

1. INTRODUCTION

Buyya et al have defined cloud as:

“A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC '13 March 24-28, Gyeongju, Korea
Copyright 2014 ACM 978-1-4503-2469-4/14/03 ...\$15.00.

or more unified computing resources based on service level agreements established through negotiation between the service provider and consumer” [8].

In a cloud, hardware/software are shared and utilized as services over the Internet at potentially lower cost. Many services in the realm of cloud computing are offered as either Infrastructure as a Service, Platform as a Service, and Software as a Service. The ever increasing need for data processing, storage, elastic and unbounded scale of computing infrastructure at lower cost has provided great thrust for shifting the data and computing operations to the cloud. IBM advocates cloud computing as a cost efficient model for service provision [16]. The adoption of cloud computing is gaining momentum because most of the services provided by the cloud are low cost and readily available. The pay-as-you-go structure of the cloud is particularly suited to Small and Medium Enterprise (SME) who have little or no resources for IT services [6]. The growing trend of cloud computing has led many organisations and even individuals to move their computing operations, data, and/or commissioning their e-services to the cloud. Moving to the cloud has reduced the cost of computing and operations due to resource sharing, virtualization, less maintenance cost, lower IT infrastructure cost, lower software cost, expertise utilization and sharing etc [27]. For example, the New York Times managed to convert 4TB of scanned images containing 11 million articles into PDF files, which took 24 hours for conversion and used 100 Amazon EC2 Instances [14]. Such relatively quick conversion would be very expensive if done in-house.

Despite the rapid growth of cloud use, there has been a general lack of systematic methodologies aiming at engineering requirements and early mitigation of risks in the cloud adoption. Decisions regarding the selection of cloud service providers are made on ad hoc basis, based on recommendations or on the reputation of the service provider. Such gap exposes businesses considering the cloud to unpredictable risks. It would be expensive to get “locked in” with a wrong cloud or to suffer from the ill compliance of Service level Agreements (SLAs). Evaluating pre adoption choices at early stages is cost-effective strategy to mitigate risks for probable losses due to wrong or uninformed selection decisions.

In [38], the authors have contributed to a novel Goal-Oriented Requirements Engineering method for cloud adoption, where the work was the first attempt ever to propose GORE for cloud adoption, evidenced by citations. The

method aims at assisting businesses in screening, negotiating service provision (functional and qualities) and consequently informing the selection decisions of cloud service providers. Due to the dynamic evolution and continuous change of cloud provision, mismatches may occur between what the user requires and what the cloud provider provide. The method aims at managing the tradeoffs associated with managing mismatches of users' requirements against cloud's Service Level Agreement (SLA) to reduce risks for probable violations. A cloud adoption lifecycle building on GORE was proposed in [38]. [7] refined the lifecycle by incorporating obstacles analysis and resolution tactics in the adoption process. Some concrete and abstract obstacle resolution tactics were also proposed, leveraging on the seminal work of [22] but tailored to the case of cloud. The expected beneficiaries of the work are small to large businesses and multi-tenant users, who wish to exploit the cloud. Such work bridges an important gap in making the process of cloud adoption transparent, systematic, user oriented and risk-aware. We argue that obstacles prioritization and their resolution are important for mitigating risks in the adoption process. The novel contribution of this paper is a new systematic method for prioritizing cloud adoption obstacles and their resolution tactics using Analytical Hierarchy Process (AHP). We demonstrate the applicability and usefulness of the approach through an example. We do the sensitivity and stability analysis to assess the AHP choice of the resolution tactics.

2. THE ANALYTICAL HIERARCHY PROCESS FOR HANDLING OBSTACLES IN CLOUD ADOPTION USING EXAMPLE

Since its early applications in the seventies, the Analytical Hierarchy Process has found a wide interest both between decision theorists and appliers [33] [34]. The analytical hierarchy process (AHP) is a decision-making method and is well described in [34]. An AHP hierarchy is a vehicle for structuring a decision problem. The classical AHP hierarchy consists of an overall goal, a group of options or alternatives for achieving the goal, and a set of criteria, which relate the alternatives to the goal. The criteria can be further decomposed into subcriteria, sub-subcriteria, and so forth. In a nutshell, the AHP hierarchy can be then analyzed through a sequence of pairwise comparisons using numerical scales of measurement; these could be subjective or objective metrics. The criteria are pairwise compared against the goal for importance. The alternatives are pairwise compared against each of the criteria for preference. The comparisons are computed and priorities are derived for each node in the hierarchy. The prioritization can then provide insights for the decision makers and analysts, which can inform the selection decision. The process is structured and systematic; it overcomes the limitations of ad hoc decision making in complex decision making problems with various alternatives and criteria. AHP is sufficiently described in many text books on the subject e.g. see [34]

2.1 Motivation for Obstacle Resolution in Cloud Adoption

AHP is intuitively appealing to the obstacle resolution problem. Its hierarchy can provide intuitive and simple ready means for structuring, tracing and analyzing goals, obstacles, and their resolution tactics. Complementing the

AHP structure with its analysis and prioritization process, the combination does provide a new systematic method for prioritizing obstacles and selecting suitable tactics for resolving them for the goals to be achieved. In this context, an obstacle resolution tactic corresponds to an alternative. Resolution tactics are pairwise compared against each of the identified obstacles for preference to the extent to which its resolution can satisfy the goal. Obstacles correspond to criteria, which can be further decomposed and refined into sub-obstacles; its negation has relative importance for reaching the goal (these are pairwise compared against the goal for importance).

There is a general lack of systematic methods for prioritizing obstacles and identifying best strategies for mitigating the risks. We systematically model requirements for cloud adoption and their corresponding obstacles. Obstacles can be deemed as risks; they can have significant impact on adoption process if not properly elicited, modeled and mitigated. We propose a two-phase risk aware method for cloud adoption benefiting from Analytical Hierarchy Process. The first phase elicits and prioritizes obstacles by understanding their potential impact on adoption. The second phase searches for good-enough resolution tactics for the prioritized obstacles so they can be mitigated for risks. Stability and sensitivity analysis is proposed to assess the choice of the resolution tactics.

We argue that there are significant benefits for using AHP to manage the prioritization of obstacles and their resolution tactics. Firstly, it is beneficial to quantify obstacles' importance through understanding their consequences on the adoption process, if not properly managed and mitigated for risks. Secondly, the adoption process is often constrained by limited resources (e.g., budget, man-months, schedule), exhaustive treatments for all obstacles is often resource demanding and may be difficult. A "selective" strategy, through obstacles prioritization is a sensible and wise option to tackle the complexity of the space and to focus efforts on the critical obstacles, which cannot wait. The strategy can elevate the consideration of ones, which are likely to be critical and downgrade and eliminate others, which can be tolerated or have little/no risk impact. As obstacles can be resolved using several alternative resolution tactics, each may provide different added value. Thirdly, the importance of obstacles can vary by the orders of magnitude. Quantification of their relative importance and significance for risks can provide an objective assessment for their management and resolution. In contrary to the criticism, we argue that AHP pair-wise comparison technique can be a cost-effective strategy to probe for better testing and understanding of the presence and the importance of obstacles. The pair-wise comparison can be risk revealing. The pair-wise comparison comprises much redundancy and is therefore less sensitive to judgmental errors. AHP points to inconsistencies by calculating a consistency ratio of judgmental errors. The smaller the consistency ratio the lesser the inconsistency, hence the more reliable results. Fourthly, we acknowledge that it may be problematic to scale up for larger projects using AHP. Karlsson et al [18] recommend grouping the requirements, where grouping requirements can reduce the number of comparisons in an efficient manner. We believe that our use of GORE for cloud adoption and our handling of obstacles using GORE do provide an elegant grouping benefiting from the goal refinements hierarchy, promoting potential scalabil-

ity. Fifthly, Karlsson et al [18] evaluated six different prioritization techniques based on pair-wise comparisons and concluded that AHP was the most promising approach because it is based on a ratio scale and includes a consistency check. Last but not least, the major disadvantage of AHP is that it is time consuming for large problems. Research efforts have been made to decrease the number of comparisons and hence the time needed [17][26]. Results of these have been that the number of comparisons can be decreased by as much as 75% [17].

2.2 Obstacles and Risks in Requirements Engineering for Cloud Adoption

Obstacles were first proposed by Potts in [31]. Potts probes the elicitation of likely obstacles for a particular goal by asking questions like: “Can this goal be obstructed, if so, how?” KAOS provides well-developed and widely used systematic methods for identifying, handling and resolving obstacles [23]. In brief, van Lamsweerde et al recommend that obstacles be identified from leaf-level goals. Once identified, obstacles can be refined like goals (by AND/OR decomposition) and consequently handled, managed and resolved using appropriate tactics. According to Anton [4] while an obstacle denotes the reason why a goal failed, scenarios determine concrete circumstances under which the goal may fail. In [39], authors have demonstrated that by analyzing Service Level Agreements (SLA) of cloud providers and matching them against users’ requirements in the goal refinements process, potential SLA violations, conflicts and likely risks can be revealed. In the context of GORE, the potential violations of the SLAs can be best modeled, analyzed, linked to consumers’ goals and mitigated for risks using obstacles analysis [39].

Using obstacles analysis to mitigate likely risks of cloud SLAs violations starts by examining the consumers’ goals, where obstacles are identified from the goal graph and terminal goals, which need to be assigned to cloud agents. Cloud agents can be a software-, platform- and/or infrastructure-as-service. The obstacles can be recursively refined; it is essential that the set of identified obstacles is complete for every goal (i.e. at least for high-priority goals). To resolve obstacles we would like to identify as many obstacles as possible for the goals, particularly for high-priority goals. A set of obstacles O_1, \dots, O_n is complete for goal G if the following condition is true: $\neg O_1 \wedge \dots \wedge \neg O_n \models G$, which means that if none of the identified obstacles occur then the goal is satisfied [23]. In [39], the interlink between risks and obstacles for cloud adoption problem had been systematically modeled. A sample of obstacle resolution tactics is depicted in Table 1. User can select among these tactics to resolve the identified obstacles and mitigate risks in the adoption process. Interested readers can refer to [39] for details.

Figure 1 describes the high-level steps involved in resolving the obstacles in the cloud adoption process. Given the likely large space of possible obstacles and their resolution tactics, we use AHP for their prioritization. In first phase obstacles are prioritized based on their relative importance and significance to the goal satisfaction if been resolved. In the second phase, the application of AHP aims at informing the decision of reaching the optimal set of tactic(s) for resolving obstacles given numerous candidate alternatives. This is done by understanding their relative importance and significance upon resolving the tactics. A distinctive feature of our

method is the use of sensitivity analysis to assess the stability of the solution before we decide on a tactic to resolve the obstacle. As the assignment of weights and alternative values with respect to subjective criteria are approximate, it is preferable to get a stable solution. In other words, the order of the solution should not change with small perturbations in the values of the alternatives. If the solution is stable, we will proceed to resolve the obstacles. The method and its working steps are sufficiently described in section 3.

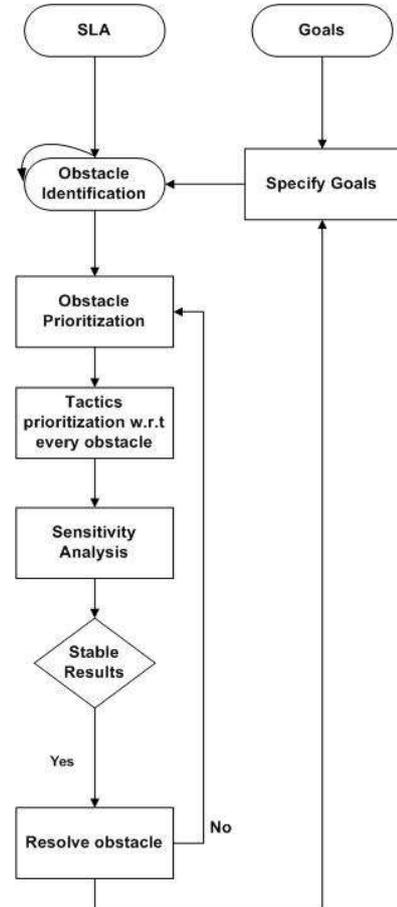


Figure 1: Steps for resolving obstacles

3. THE PROCESS OF OBSTACLE RESOLUTION FOR CLOUD ADOPTION THROUGH AN EXAMPLE

We use an Industrial case study reported in [13] by Faniyi et al referred to as Smart Bank as an example to show how AHP can be applied for mitigating risks in the process of cloud adoption. Smart Bank was interested in assessing the architectural design decisions for security upon moving to the cloud. In [13], the architecture design decisions were evaluated using Architectural Tradeoffs Analysis Method (ATAM) incorporating dynamic analysis via implied scenario generation and were named as ATMIS. Looking at the case, we have identified the goals and the obstacles that

Table 1: Obstacle resolution tactics for cloud adoption

Category	Short Description	Obstacle Resolution Tactic	Example
Obstacle Prevention	Avoid The Obstacle	Tactic1: Add another goal to prevent obstacle Tactic2: Improve/Negotiate SLA.	Tactic 1: Encrypt data (add goal) before sending it to the cloud to achieve security of the data Tactic 2: Negotiate SLA for storing the data in the desired location
Cloud Service Substitution	Assign the responsibility of obstructed goal to another cloud	Tactic3: Transfer the operationalization of the goal to another cloud Tactic4: Split goal handling among multiple clouds	Federated cloud can be used for different applications to match business needs
Goal Weakening	Weaken the obstructed goal specification	Tactic 5: Weaken goal objective function Relax requirements Relax the degree of satisfaction	Response time required for an email service can be increased to few seconds if the cloud provider is unable to guarantee the desired response time

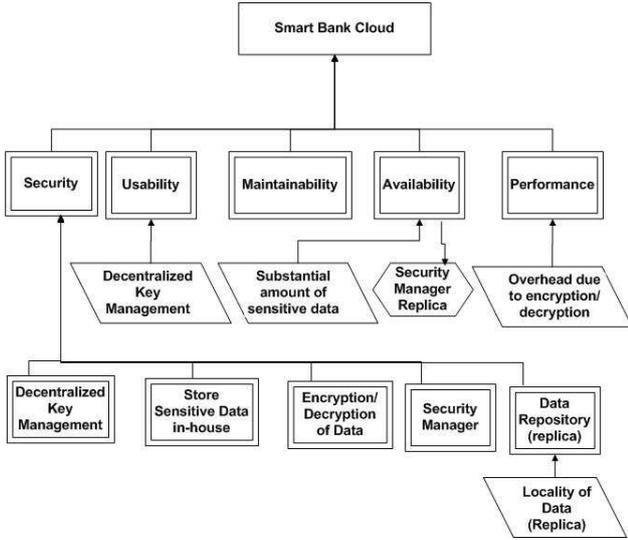


Figure 2: Portion of the goal tree for Smart Bank

may be encountered for achieving those goals using GORE, as described in Figure 2. We have adopted the following conventions. Double edged rectangles show the goals, parallelograms are the obstacles in achieving some of the goals and hexagons represent a cloud agent operationalising the goal. This can be a cloud software-, infrastructure-, or platform-as a service or a cloud service provider. An AND-refinement is represented by an arrow with a small circle connecting the subgoals contributing to the parent goal. An OR-refinement is graphically represented by multiple arrows pointing to the same goal.

3.1 Building a Hierarchy for Smart Bank Obstacles

We use AHP for prioritizing the obstacles and obstacle resolution tactics. As the decision in the obstacles resolution process revolved around goals, their likely obstacles and their candidate tactics for resolving them, it is perfectly justified to be modeled using AHP primitives. Figure 3 shows the hierarchy of goals, obstacles and the obstacle resolution tactics. The overall goal in our case is to resolve obstacles in smart bank example. We build the hierarchy from the top

(depicting goals) through the intermediate levels (depicting obstacles) and moving to the lowest level, which comprises the list of alternative resolution tactics. The major decision criteria occupy the second level of the hierarchy (i.e. which obstacles should be resolved and their relative importance for the goal’s satisfaction), and the sub-criteria occupy the third level of hierarchy. The last level of hierarchy has all the alternative options. Four obstacles were identified for achieving different goals of the smart bank. For simplicity we are only making a hierarchy of the main goal, their obstacles and resolution tactics. We have not included the subgoals in the hierarchy for the simplicity of exposition.

3.2 Scale for Pairwise Comparisons

Prioritization of both obstacles and obstacle resolution tactics is done through pair-wise comparison. Thomas L. Saaty proposed a 9-grade value scale to compare different choices. Scale ranges from 1 (equal value) to 9 (where an option is extremely more valuable than the other). In first phase obstacles are prioritized based on their relative importance and significance to the goal satisfaction if been resolved. In the second phase, the application of AHP aims at informing the decision of reaching the optimal set of tactic(s) for resolving obstacles given numerous candidate alternatives. Understanding their relative importance and significance upon resolving the tactics does this. By using AHP user can prioritize the obstacles that should be resolved first and can select the best obstacle resolution tactic for a particular obstacle. We use Table 2 for valuing the pairwise comparisons. We refer obstacle Substantial Amount of sensitive data as O1, obstacle Locality of data as O2, obstacle Overhead due to encryption/decryption as O3 and obstacle Decentralized key management as O4.

3.3 Prioritize Obstacles Using AHP

Prioritize the obstacles by pairwise comparisons using Table 2. The values are inserted in a matrix. Since there are 4 obstacles, the matrix is 4x4 order. An element is equally important when compared to itself, so where the row of O1 and column of O1 meet in position (O1, O1) insert 1. The diagonal of the matrix therefore must consist of 1’s. For each pair of obstacles (starting with O1 and O2, for example) insert their determined relative priority in the position (O1, O2) where the row of O1 meets the column of O2. In position (O2, O1) insert the reciprocal value. Continue to perform pairwise comparisons of O1-O3, O1-O4, and so on.

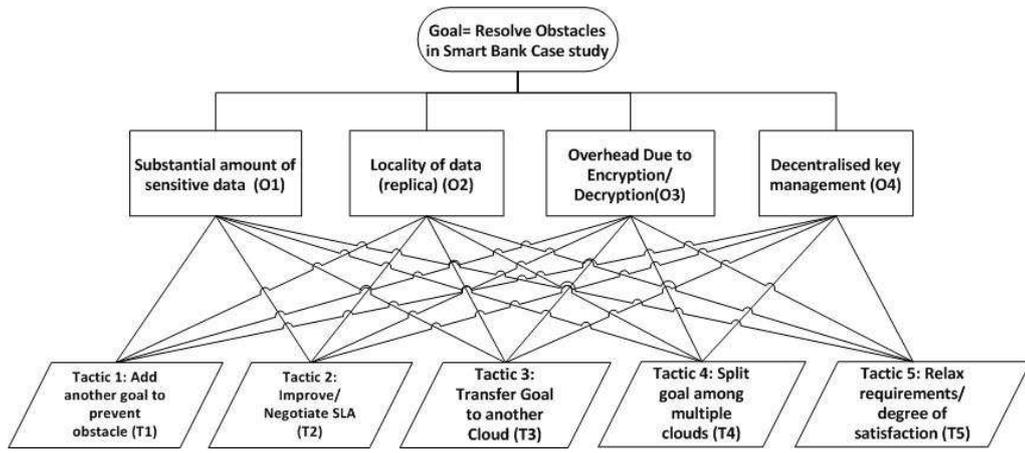


Figure 3: Hierarchy for Smart Bank

Table 2: Scale for pairwise comparisons

Relative Importance	Short Definition	Explanation
1	Equally important	Two obstacles are equally critical
3	Weakly more important	Slightly favor resolving one obstacle over another
5	Strongly more important	Strongly favor obstacle for resolving over another
7	Very strongly more important	An obstacle resolution is strongly favored and its dominance is demonstrated in practice
9	Extremely important	The evidence favoring one over another is of the highest possible order of affirmation
2, 4, 6, 8	Intermediate values between two adjacent judgments	Used to facilitate compromise between two slightly differing judgments
Reciprocals	If obstacle A has one of the above numbers assigned to it when compared with obstacle B, then B has a reciprocal value when compared with A.	

Thus for prioritizing obstacles, six pairwise comparisons are required. The values that are inserted in the matrix can be obtained by a voting procedure where each stakeholder is allowed to allocate a number from Table 2 to the obstacles. Stakeholders who should be involved in the prioritization of the obstacles can be identified using the already existing tools like StakeNet [25].

Pairwise comparison of obstacles according to their criti-

cality on the system gives the following matrix

$$\begin{matrix}
 & O1 & O2 & O3 & O4 \\
 \begin{matrix} O1 \\ O2 \\ O3 \\ O4 \end{matrix} & \begin{pmatrix} 1 & 5 & 3 & 7 \\ 0.20 & 1 & 0.33 & 5 \\ 0.33 & 3 & 1 & 5 \\ 0.14 & 0.20 & 0.20 & 1 \end{pmatrix}
 \end{matrix}$$

After comparisons normalize each column (add its components and divide each component by the sum of that column). We obtain the following matrix.

$$\begin{pmatrix} 0.5966 & 0.5435 & 0.6618 & 0.3889 \\ 0.1193 & 0.1087 & 0.0735 & 0.2778 \\ 0.1989 & 0.3261 & 0.2206 & 0.2778 \\ 0.0852 & 0.0217 & 0.0441 & 0.0556 \end{pmatrix}$$

The sum of the rows of normalized matrix is a column vector which when averaged by the size of 4 columns yields the vector of priorities. After obtaining the priority vector multiply the vector with matrix of comparisons, which will result in a new product vector. Divide the first component of product vector with the first component of priority vector, the second component of the product vector with second component of the priority vector and so on, we obtain another vector called ratio vector. The calculations made for prioritizing the obstacles reveal that O1 has the highest priority followed by O3, O2 and O4 respectively.

$$\begin{pmatrix} Priority & Product & Ratio \\ 0.5477 & 2.400939 & 4.383829 \\ 0.1448 & 0.597942 & 4.128573 \\ 0.2558 & 1.13118 & 4.421621 \\ 0.0517 & 0.210032 & 4.065666 \end{pmatrix}$$

Take the sum of the components of the ratio vector and divide it by the number of components we have an approximation to a number λ (called the principal eigenvalue). The closer the λ is to n (the number of activities in a matrix) the more consistent is the result. The deviation from consistency is represented by $(\lambda-n)/(n-1)$ which is the consistency index (CI). The ratio of consistency index to random index (RI) is called the consistency ratio (CR). The values of random index for matrix of different orders are defined in [34].

The consistency ratio of 0.10 or less is considered acceptable. For obstacles the consistency ratio is acceptable. CI=0.083307 CI/RI= 0.09

3.4 Selecting an Appropriate Obstacle Resolution Tactic

We compare all the five obstacle resolution tactics with respect to every obstacle for selecting the best resolution tactic. We calculated the priority of obstacle resolution tactics for the four major obstacles using AHP. Section 4 shows the results of calculations in the form of a matrix.

For decentralized key management the best tactic is Tactic 5. For the obstacle overhead due to encryption/decryption the best tactic is Tactic 4 where the bank can keep its data on two different providers' servers. Obstacle substantial amount of sensitive data can be resolved using Tactic 1 which has the highest priority. For the Locality of data the best obstacle resolution tactic is Tactic 4 i.e. smart bank should engage into negotiation before signing up the contract for cloud to make clear where the data of the bank will be stored.

4. STABILITY AND SENSITIVITY ANALYSIS

The selection of obstacle resolution tactics can be considered as a multicriteria decision problem. When solving a multicriteria decision problem, it is desirable to choose a decision function that leads to as stable a solution as possible. Since the assignment of weights to the tactics and obstacles is based on human judgment, it would be preferable to obtain the same solution (i.e. to have a stable solution) when slightly modifying the weights. We consider the decision functions based on α power means [12]:

$$\left(\frac{\sum_{i=1}^n w_i a_i^\alpha}{\sum_{i=1}^n w_i} \right)^{1/\alpha} \quad (1)$$

Where the weights w_i in our case are the priority vector for the obstacles and the alternative values a_i are the different tactics.

We calculate the stability index of our multicriteria decision functions for perturbations of alternative values ϵ as:

$$S(\epsilon) = (\prod_{j=1}^{n-1} \delta_j(\epsilon))^{\frac{1}{n-1}} \quad (2)$$

where δ_j is the stability value of the order between alternative j and $j+1$, if the alternative values are allowed to vary by at most ϵ . A value of $S(\epsilon)=1$ is equivalent to a stable solution whereas a value of $S(\epsilon)=0$ corresponds to an unstable solution. In order to calculate the stability of our solution, we put the priority vectors in a matrix for every obstacle:

Weight	T1	T2	T3	T4	T5	
O1	0.5477	0.5083	0.2306	0.1357	0.0735	0.0519
O2	0.1448	0.0459	0.5095	0.2197	0.1418	0.0832
O3	0.2558	0.0439	0.0749	0.1573	0.5186	0.2052
O4	0.0517	0.2261	0.1316	0.0782	0.0428	0.5213

The α power mean aggregated values for the different tactics for values of α of 2, 3 and 5 are shown in Table 3.

We consider two situations: potential changes of up to $\epsilon=100\%$ of the alternative values and potential changes of up to $\epsilon=10\%$ of the alternative values. We calculate the stability values for the order for each pair of consecutive

Table 3: α power mean aggregated values for different tactics

α	T1	T2	T3	T4	T5
2	0.3807	0.2627	0.1540	0.2734	0.1652
3	0.4170	0.2966	0.1577	0.3311	0.2132
5	0.4507	0.3510	0.1654	0.3949	0.2909

Table 4: Stability index for resolution tactics

α	Stability Index S(0.1)	Stability Index S(1)	$\delta(1) \setminus \text{Priority}(\text{tactic})$			
			1 (T1)	2 (T4)	3 (T2)	4 (T5)
2	0.4886	0.0669	0.16	0.019	0.22	0.03
3	0.8612	0.1079	0.11	0.055	0.16	0.14
5	0.7632	0.0978	0.065	0.058	0.09	0.27

alternatives $\delta_j(\epsilon)$ and then the stability index $S(\epsilon)$. Table 4 shows the calculated stability index.

The stability index in both cases is highest for $\alpha=3$, suggesting that as overall solution this is the best. For variations of up to 10% of the original value the 3-power mean decision function is close to perfect stability. However, the use of the 2-power mean is recommended if only the order of the first two tactics is of interest, given that the best tactic will be selected and the largest δ value for the order between tactic T1 and T4 is achieved when using the 2-power mean.

5. RELATED WORK

A comprehensive use of Goal-oriented requirements engineering can be found in [37]. Requirements are represented in the form of goals in GORE. The advantage of using this approach is that a goal graph provides vertical traceability from high-level strategic concerns to low-level technical details; it allows evolving versions of the system under consideration to be integrated as alternatives into one single framework. Goal-oriented approaches have received significant attention over the years, where goals were used for modelling functional requirements [24], non-functional requirements, and Agent-oriented systems [7]. For example, Mylopoulos et al used goal-oriented approach for eliciting, specifying and refining the non-functional requirements [28][7] demonstrates another use of Goal-oriented approaches in Agent Oriented Programming (AOP) for open architecture that need to change and evolve due to changing requirements. GORE was also used to model the system architecture to meet changing business goals and for evolving systems [5][15]. One interesting application of GORE, which has inspired our work is that of [3][10], where GORE was used to inform the process of selecting Commercial off the Shelf (COTS) products matching user's requirements. Though the fundamental use of GORE exhibits resemblance with that of [3][10], the problem of cloud adoption is by far more challenging as we are dealing with "open loop" systems, with dynamic, unbounded and elastic scale where continuous service evolution is the norm. Due to continuous evolution of the cloud there may be numerous obstacles for adopting cloud services. [23] presents a detailed view of handling obstacles in goal-oriented requirements engineering. The work of [39] is the earliest attempt for linking obstacle analysis and goal elaboration to risks. The work of [9] has followed a similar argument. [9] proposed a probabilistic framework for risk assessment in the requirements analysis phase.

The AHP has been applied to wide variety of decisions.

In computer science it has been used for selecting software project management tools [1], software requirements prioritization [18][32] where requirements are ranked in two dimensions: according to their value to the customer and according to their estimated cost of implementation, software selection [21], COTS selection [20], prioritizing quality characteristics for component-based development [35], vendor selection [36], software architecture evaluation [40], and selection of websites [30]. AHP has also been used for architecting distributed software application in conjunction with integer programming where it has been used to determine the optimal combination of design alternatives [2]. AHP has so far not been used to mitigate risks in a cloud. In case of the cloud where large number of users is concerned and little is known about the cloud service provider it is quite difficult to devise a risk management plan. So far, we are not aware of any GORE-AHP strategy related approaches for mitigating risks using obstacles analysis.

Research efforts over the years have looked at the problem of service discovery with runtime mechanisms to inform and optimize the selection e.g. self-managed applications in the cloud [29] and self-optimizing architecture [29]. There is a recent research on cloud migration [19]; this research does not involve user requirements for cloud adoption. [11] has proposed a goal-oriented simulation approach for cloud-based system design. Up to the authors' knowledge, there has been no research on cloud procurement and adoption from requirements engineering perspective using the notion of obstacles. The need for such research is timely as there is complete lack of systematic methodologies, which could help stakeholders screen, match, negotiate their requirements against cloud services' provision and manage the tradeoffs associated with matches/mismatches of users' requirements and mitigating risks.

6. CONCLUSION AND FUTURE WORK

This paper presents a systematic risk-aware method for prioritizing obstacles encountered in the cloud adoption process and their resolution tactics using Analytical Hierarchy Process (AHP). The method leverages on goal-oriented requirements engineering, obstacles handling and AHP to systematically model risks and their mitigation strategies. We have exemplified the steps of the method using a non-trivial case to demonstrate its applicability. To evaluate the accuracy of the approach, we have used stability and sensitivity analysis. The analysis aims to assess the AHP choice of resolution tactics, which have the potentials to mitigate risks. We have observed that for variations of up to 10% of the original value, the 3-power mean decision function tend to be close to perfect stability. This means that the method can accommodate up to 10% "discrepancy" upon the allocation of weight by various stakeholders. Ongoing work includes an automated tool support for the cloud adoption and obstacles resolution processes using AHP.

7. ACKNOWLEDGMENTS

The research leading to these results has received funding from NED University of Engineering and Technology, Pakistan. Ms. Zardari would like to thank her son Emaad Navid Shaikh for his patience and support during her research.

8. REFERENCES

- [1] N. Ahmad and P. A. Laplante. Software project management tools: Making a practical decision using AHP. In *The 30th IEEE/NASA Software Engineering Workshop*, 2006.
- [2] T. Al-Naeem, I. Gorton, M. A. Babar, F. Rabhi, and B. Benatallah. A quality-driven systematic approach for architecting distributed software applications. In *The ACM 27th International Conference on Software Engineering*, 2005.
- [3] C. Alves and A. Finkelstein. Challenges in COTS decision-making: A goal-driven requirements engineering perspective. In *The 14th International Conference on Software Engineering and Knowledge Engineering*, 2002.
- [4] A. I. Anton. Goal-based requirements analysis. In *The Second IEEE International Conference on Requirements Engineering, Colorado, USA*, 1996.
- [5] A. I. Anton and C. Potts. The use of goals to surface requirements for evolving systems. In *The 20th International Conference on Software Engineering*, 1998.
- [6] S. Biggs and S. Vidalis. Cloud computing: The impact on digital forensic investigations. In *the International Conference for Internet Technology and Secured Transactions*, 2009.
- [7] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8:203D1d7236, 2004.
- [8] R. Buyya, C. S. Yeo, and S. Venugopal. Market oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In *The 10th IEEE International Conference on High Performance Computing and Communication*, 2008.
- [9] A. Cailliau and A. van Lamsweerde. Assessing requirements-related risks through probabilistic goals and obstacles. *Requirements Engineering, Springer*, 18, Issue 2:129–146, 2013.
- [10] L. Chung and K. Cooper. A knowledge-based COTS-aware requirements engineering approach. In *The 14th International Conference on Software Engineering and Knowledge Engineering*, 2002.
- [11] L. Chung, T. Hill, O. Legunsen, Z. Sun, A. Dsouza, and S. Supakkul. A goal-oriented simulation approach for obtaining good private cloud-based system architectures. *Journal of Systems and Software*, 86(9):2242 – 2262, 2013.
- [12] A. Ekart and S.Z. Nemeth. Stability analysis of tree structured decision functions. *European Journal of Operational Research*, 160:676–695, 2005.
- [13] F. Faniyi, R. Bahsoon, A. Evans, and R. Kazman. Evaluating security of architectures in unpredictable environments: A case for cloud. In *the 9th Working IEEE/IFIP Conference on Software Architecture*, 2011.
- [14] D. Gottfrid. Protrated supercomputing fun! The New York Times, November 2007.
- [15] D. Gross and E. Yu. Evolving system architecture to meet changing business goals: an agent and goal-oriented approach. In *The Fifth IEEE*

International Symposium on Requirements Engineering, 2001.

- [16] IBM. IBM perspective on cloud computing, 2008.
- [17] J. Karlsson, S. Oisson, and K. Ryan. Improved practical support for large-scale requirements prioritising. *Requirements Engineering, Springer-Verlag London*, 2:51–60, 1997.
- [18] J. Karlsson, C. Wohlin, and B. Regnell. An evaluation of methods for prioritizing software requirements. *Information and Software Technology, Elsevier Science*, 39:939–947, 1998.
- [19] A. Khajeh-Hosseini, D. Greenwood, and I. Sommerville. Cloud migration: A case study of migrating an enterprise it system to IaaS. In *the 3rd International Conference on Cloud Computing*, 2010.
- [20] J. Kontio, S.-F. Chen, K. Limperos, R. Tesoriero, G. Caldiera, and M. Deutsch. A COTS selection method and experiences of its use. In *The 20th Annual Software Engineering Workshop at NASA Goddard Space Flight Center, Greenbelt, Maryland*, 1995.
- [21] V. S. Lai, B. K. Wong, and W. Cheung. Group decision making in a multiple criteria environment: A case using the AHP in software selection. *European Journal of Operation Research*, 137:1341–1344, 2002.
- [22] A. V. Lamsweerde and E. Leiter. Integrating obstacles in goal-driven requirements engineering. In *20th International Conference on Software Engineering, Kyoto*, 1998.
- [23] A. V. Lamsweerde and E. Leiter. Handling obstacles in goal-oriented requirements engineering. *IEEE Transactions on Software Engineering*, 26:978–1005, 2000.
- [24] A. V. Lamsweerde and E. Leiter. From object orientation to goal orientation: A paradigm shift for requirements engineering. In *Radical Innovations of Software and Systems Engineering in the Future, Springer-Verlag, LNCS*, volume 2941, pages 325–240, 2004.
- [25] S. L. Lim, D. Quercia, and A. Finkelstein. Stakenet: Using social networks to analyse the stakeholders of large-scale software projects. In *32nd ACM/IEEE International Conference on Software Engineering*, 2010.
- [26] Frank J. Carmone Jr., A. Kara, and S. H. Zanakis. A Monte Carlo investigation of incomplete pairwise comparison matrices in AHP. *European Journal of Operational Research*, 102, 1997.
- [27] M. Miller. *Cloud Computing: Web Based Applications That Change the Way You Work and Collaborate Online*. Que Publishing, 2008.
- [28] J. Mylopoulos, L. Chung, and B. Nixon. Representing and using nonfunctional requirements: A process oriented approach. *IEEE Transactions on Software Engineering*, 18(6):483–497, 1992.
- [29] V. Nallur, R. Bahsoon, and X. Yao. Self-optimizing architecture for ensuring quality attributes in the cloud. In *The 2009 IEEE/IFIP WICSA/ECSA*, 2009.
- [30] E. W. T. Ngai. Selection of websites for online advertising using AHP. *Information and Management, Elsevier*, 40:233–242, 2002.
- [31] C. Potts. Using schematic scenarios to understand user needs. In *The 1st Conference on Designing Interactive Systems: Processes, Methods and techniques*, 1995.
- [32] K. Ryan and J. Karlsson. Prioritizing software requirements in an industrial setting. In *The International Conference on Software Engineering, Boston, USA*, 1997.
- [33] T. L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15, 1977.
- [34] T. L. Saaty. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill, 1980.
- [35] A. Sharma, R. Kumar, and P. Grover. Estimation of quality for software components: an empirical approach. *SIGSOFT Software Engineering Notes*, 33:1–10, 2008.
- [36] M. C. Tama and V. Tummala. An application of the AHP in vendor selection of a telecommunications system. *Omega, Elsevier*, 29:171–182, 2001.
- [37] A. van Lamsweerde. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley, 2009.
- [38] S. Zardari and R. Bahsoon. Cloud adoption: A goal-oriented requirements engineering approach. In *Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing, SECloud '11*, pages 29–35, New York, NY, USA, 2011. ACM.
- [39] S. Zardari, F. Faniyi, and R. Bahsoon. Using obstacles for systematically modeling, analysing, and mitigating risks in cloud adoption. *Aligning Enterprise System and Software Architectures, IGI Global*, pages 275–296, 2012.
- [40] L. Zhu, A. K. Aurum, I. Gorton, and R. Jeffery. Tradeoff and sensitivity analysis in software architecture evaluation using analytical hierarchy process. *Software Quality Journal, Springer*, 13:357–375, 2005.