Cloud Adoption: A Goal-Oriented Requirements Engineering Approach

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

ABSTRACT
We motivate the need for a new requirements engineering methodology for systematically helping businesses and users to adopt cloud services and for mitigating risks in such transition. The methodology is grounded in goal oriented approaches for requirements engineering. We argue that Goal Oriented Requirements Engineering (GORE) is a promising paradigm to adopt for goals that are generic and flexible statements of users' requirements, which could be refined, elaborated, negotiated, mitigated for risks and analysed for economics considerations. We describe the steps of the proposed process and exemplify the use of the methodology through an example. The methodology can be used by small to large scale organisations to inform crucial decisions related to cloud adoption.

Categories and Subject Descriptors
D.2.7 [Software Engineering]: Requirements Engineering, Cloud Computing

General Terms
Management, Legal Aspects.

Keywords
Goal Oriented Requirements Engineering, Cloud based GORE

1. INTRODUCTION
The ever increasing need for data processing, storage, elastic and unbounded scale of computing infrastructure 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 [1]. 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 [5]. The growing trend of cloud computing has led many organisations and even individuals moving 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 [3]. 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 [4]. Such relatively quick conversion would be very expensive if done in-house. The term cloud computing may simply refer to different applications over the Internet or the hardware shared between different users [2]. 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 or more unified computing resources based on service level agreements established through negotiation between the service provider and consumer*" [9]

In a cloud, hardware/software are shared and utilized as services at lower cost. Many services are now offered in the realm of cloud computing. These are:

- Infrastructure as a service (IaaS),
- Database as a service (DaaS),
- Platform as a service (PaaS),
- Software as a service (SaaS).

In cloud context, a service level agreement (SLA) is expected to mediate consumers' expectations with respect to cloud service provision. Despite the fact that businesses have already started to exploit the potentials of the cloud as a paradigm for services, there is a total absence for foundation and methodologies for informing the process of cloud adoption. As a result, the current practice for adopting clouds (and their services) could be described as ad hoc, lacking rigour and systematic guidance. Selection is often biased by the reputation of the cloud provider, their SLA statements, terms, conditions and promises, recommendations and clouds' reputations, past experiences and the like from biased or subjective ad hoc inputs. In particular, there is a general lack of requirements engineering methodologies, which could be suited for cloud adoption. The objective of our research is to develop a requirements engineering framework, which could help users screen, match, and negotiate their requirements against cloud

SECLOUD'11, May 22, 2011, Waikiki, Honolulu, HI, USA

services' provision. The framework will also assist in the problem of managing the tradeoffs associated with matches and mismatches of users' requirements against cloud service provisions.

To bridge the gap, practitioners are very likely to beg, borrow, and steal from requirements engineering in-the-small to benefit the ultra-large-scale paradigm as it is the case of the cloud. Though the fundamentals of engineering requirements in both paradigms exhibit resemblance, requirements engineering for the cloud requires novel flexible and scalable approach, which address the interplay between technical, economics-driven considerations and shifts requirements engineering towards a utility-based engineering for consumers' satisfaction for cloud service provisions such as software as-, infrastructure-, data storage-, and/or platform as- services.

2. MOTIVATION

We looked at a case study of a leading cloud service provider-referred to as *Indus* throughout this paper. The case study revealed that there are many risks associated with cloud adoption. Since the cloud is perceived as a "black box", a user has little or no control over the promises set by cloud's SLAs. For instance, the user cannot negotiate the SLAs with the cloud service provider and hence has to agree with the set terms and conditions.

As an example, Indus sufficiently describes their security mechanisms in the agreement accompanying their SLAs "Indus Web Services Customer Agreement." Interestingly enough, Indus mentions that the nature of communication over the Internet is unpredictable and largely insecure. Given the vulnerability of the Internet, Indus cannot guarantee the security of user's content. While Indus strives for a secure environment, the security responsibility and accountability lie solely on the users and the organisation using the services. In the event of any breach in security requirements, Indus is not entirely liable to the user for any unauthorized access, use, deletion, corruption or destruction of user's content. The service provider has attempted to win the confidence and trust of the users by publishing the agreement, yet it has failed to ascertain the individual needs of different users. Indus has a shared responsibility environment for the safety of user's content, where one of the inherent problems with such "joint" responsibility is that it makes accountability difficult. Referring to SLAs terms and conditions, Indus absolves itself of any responsibility in the wake of anything goes wrong. For instance, Indus recommends customers to encrypt data transferred over the network. There are tradeoffs involved with large data encrypted over the network: this will lead to higher processing time, which might affect cloud performance and consequently violate the promises set in the SLA. Despite the promised dynamic elasticity of cloud architecture, resources continue to be scarce. E.g. enhancing security provision may cause performance bottleneck. As a result, the promised Quality of Service (OoS) can't be often met as the SLA terms and conditions stipulate.

We call for a novel requirements engineering methodology for cloud adoption, which could assist businesses in screening, selecting cloud service providers and negotiating their services and qualities of provision. The framework aims at helping businesses screen, match, and negotiate their requirements against cloud services' provision. The framework will also assist in the problem of managing the tradeoffs associated with matches and mismatches of users' requirements against cloud's provision. Such provision aims at objectively evaluating the strategic decisions, satisfaction of the technical and operational goals, cost and value of such decisions and the tradeoffs involved in moving to the cloud. Despite rapid growth of cloud use, there is a general lack of systematic methodologies aiming at such. 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. The lack of such methodologies exposes businesses considering the cloud for unpredictable risks. It would be expensive to get "locked in" with a wrong cloud. Evaluating pre adoption choices at early stages is cost-effective strategy to mitigate risks of probable losses due to wrong or unjustified selection decisions. Furthermore, the framework aims at assisting users in assessing their requirements against cloud provision. Due to the dynamic nature of the cloud, mismatches may occur between what is required by the user and what is provided by the cloud provider. The framework will assess the suitability of cloud service providers by exploring mismatches, managing risks, and suggesting possible tradeoffs. We use goal-oriented approach for cloud provision. The expected beneficiaries of the work are small to large businesses, educational institutes and even individuals, who wish to exploit the cloud. Such work is novel and bridges an important gap in making the process of cloud adoption more transparent, systematic and user oriented. It would be worth noting that ongoing research on cloud selection has addressed the problem of dynamic selection of the cloud services with respect to QoS. Up to our knowledge, no research has looked at cloud adoption from a requirements perspective.

3. REQUIREMENTS ENGINEERING FOR CLOUD ADOPTION

Let us assume that University of Birmingham (UoB) wishes to outsource its email services to the cloud. This is a crucial decision as the university has plenty of specific requirements, which clouds' providers need to satisfy. For example, the UoB shall continue to conform to the data protection act of the United Kingdom. It shall also conform to the university regulations for processing institutional data. It shall avoid any risks or liability due to breaches in data protection and confidentiality. Consequently, UoB remains apprehensive about the implications of outsourcing. The cloud provider shall also satisfy UoB's functional requirements. The cloud provider shall also satisfy UoB's requirements related to non-functionalities such as reliability, safety, 24/7 availability, response time in peak usages and unbounded scalability of the e-mail service provision. Furthermore, UoB shall also analyse the cost versus benefit of the decision in moving to the cloud as when compared to in-house provision. The selection should also address UoB's requirements for maintainability, future upgrades along with the need for reducing future operational cost. The decision involves concerns of many users regarding cultural shift with cloud adoption. UoB would strive for systematic evaluation for all the tradeoffs and risks involved in considering specific clouds. As the decision concerns many users and would result in a cultural shift in practice; UoB would strive for systematic evaluation for all the tradeoffs and risks while considering specific clouds.

Requirement Elicitation: The requirements for cloud adoption should not cover all the details in the initial stage. The initial stages of requirements engineering for cloud should be flexible enough to permit negotiations and further refinements owing to the evolving needs of the user. One of the reasons to keep requirements generic is that we don't want to eliminate promising cloud service providers at the initiation. Cloud service provider cannot meet all the requirements of the user; it is therefore, unwise to spend time and efforts on eliciting a comprehensive set of requirements that might not eventually be satisfied. To start with, we should acquire the core requirements which should not be rigid; they should allow future negotiations and elaboration. As soon as the basic requirements are elicited, the search for a potential cloud service provider can start; next requirements can be elicited with the evaluation of the promised cloud providers and services provisions. UoB can begin search for prospective cloud service providers after eliciting the core requirements (e.g. security of the data, backups, availability, core functional requirements for the e-mail service etc.)

Requirement Analysis and Negotiation: The requirements analysis is an interactive and iterative process, where the refinement of the organisation's requirements is driven by the availability of the cloud service providers and the accessibility of information related to the provided features of service usages, the quality of service provisions and other terms and conditions etc. Information documenting features and service provisions of different clouds are accessible through various sources. Examples include and not limited to the cloud white papers, SLAs, available benchmarks e.g. CloudHarmony, the Internet, reviews, evaluation and recommendations of users, experiences, cloud marketing representatives and the like. By looking at the features provided by the cloud service providers, the user/organisation may come up with new set of requirements that were not initially identified. The examination of the cloud service providers' features is a better technique to refine and understand how high-level requirements of the stakeholders can be actually satisfied. As cloud services and SLAs are designed to satisfy the generic and wider requirements of the market, some users' requirements may not be satisfied. Users' should, therefore, be prepared to engage in extensive process of requirement prioritization and negotiation. Negotiation can be either active or passive. In active negotiation the user negotiates with the cloud service provider or with an agent representing the cloud service provider. Agent acts as an intermediary between user and the cloud service provider [27]. Passive negotiation refers to when the user uses readily available information like benchmarks, expert judgment and reputation to assess the features of cloud service provider. The cloud service provider is prepared to enter into a negotiation stage if the value added of attracting an additional user is substantial or it outweighs the cost of negotiation. Moreover it is a common practice for some cloud service providers such as Amazon to allow negotiation of SLAs terms and conditions [28]. In order to successfully negotiate their requirements, users have to perform continuous tradeoffs and risk analysis in satisfying a particular requirement against the limitations of the available cloud providers. As an example, UoB has to abide by the Data Protection Act of the United Kingdom (UK) but the cloud service provider's storage location is outside United Kingdom. UoB looks at the other possible storage locations and agrees to store data in any European Union member country. This will allow UoB to use cloud services while not violating the Data Protection Act of UK. In other words, cloud's available feathers may help in refining and informing more realistic set of user's requirements. Moreover, UoB can use this phase to investigate and possibly negotiate about the location of data, access controls, backups and archiving mechanisms, QoS provision and so forth.

Requirements Evaluation: In the process of selecting candidate cloud service provider(s), the evaluation of requirements is a continuous process of cloud evaluation and requirements negotiation, where the final set of requirements will undoubtedly be a compromise and fair reflection between what users want to achieve and what is offered by the cloud service provider. The purpose of requirement evaluation is to short list the cloud candidates and to ignore some service providers. The requirements evaluation phase will try to screen candidate cloud service providers for further evaluation.

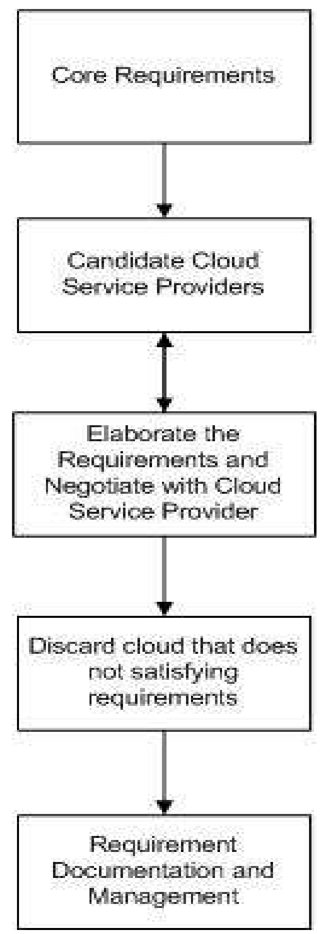

Figure 1. Requirements Engineering steps for Cloud Adoption

Requirements Documentation and Management: The requirements documentation can be regarded as semi-formal prebinding connection between the user and the cloud service provider; it acts as an informal contract for cloud adoption and users' requirements/cloud features negotiation. In the process of cloud service provider's selection, the requirements documentation will serve as the basis for cloud evaluation. Once UoB has refined its requirements, it will prepare a document wherein all the requirements will be mentioned. This set of requirements will be agreed upon by the cloud service provider and UoB. Requirements are documented to ensure that the cloud service provider gives exactly what is required by the user. In case of a cloud the requirements engineering should be a continual process involving tradeoffs analysis, negotiation and risk management. The reason behind doing so is that if the cloud service provider brings any changes to its system then the user will have to update the system according to the changes. For example this will require all the users of UoB's email service to upgrade their operating system to gain access to their emails. Figure 1 shows requirements engineering steps for cloud adoption.

4. CLOUD- BASED GOAL ORIENTED REQUIREMENTS ENGINEERING

The specification of users' requirements is the first step towards selecting an appropriate cloud service provider. Boehm estimates that the late corrections of requirements errors could cost 200 times as much as corrections during requirements engineering phases [12]. Though the Boehm's argument is related to the case of projects and software developed for relatively defined set of requirements, it would be argued that late rectification of unwise selection may lead to higher cost, which could be unproportional to the benefits of using the cloud and will place such investment into risk.

Our approach starts with high-level goals. Initially when any organisation decides to move to the cloud, the goals are general expression of the requirements, whether these are strategic requirements, functional and/or non-functional. The higher level goals (such as secure payment transactions) are more stable than low level ones (such as use Secure Socket Layer (SSL) Technology). Defining goals is the first step of requirements elicitation in software engineering. Pamela Zave in her research defines requirements engineering as something which is "concerned with the real world goals for functions of and constraints on software systems" [11]. A goal is a target that a system must achieve. Goals are generally more stable than the requirements to achieve them [13]. Goals cover different types of issues - both functional and non-functional. The functional goals can be associated with the services to be provided whereas non-functional goals can be associated with the quality of service e.g. security, reliability, availability etc [6]. Goals represent the roots for detecting conflicts among requirements and for resolving them in time [14]. GORE uses goals for eliciting, elaborating, analysing, documenting, modifying and refining the requirements [7]. We specify the requirements of the stakeholders in the form of goals.

To make a sensible decision for selecting a cloud service provider, businesses require a methodology which could help them make a choice. In conventional software development, the requirements engineering basically consists of eliciting stakeholders' needs, refining the goals into non-conflicting requirements, followed by validating these requirements with stakeholders [8]. The main objective of the requirements engineer is to ensure that the requirements specifications meet stakeholders' needs and represent a clear description of the cloud services that are to be adopted. Stakeholders' requirements play a deciding / leading role in cloud service provider's selection. Before a final selection of the cloud service provider is made goals would have to go through certain steps as shown in Figure 3. The lifecycle defined in Figure 3 is based on the steps involved in Requirements Engineering for Cloud Adoption as shown in Figure 1.

4.1 Acquire and Specify Goals

Requirements act as criteria to compare and evaluate cloud service providers. The goals can be divided into three categories (i) strategic or business goals, (ii) high level or core goals, (iii) and low level or operational goals. Strategic goals are concerned with the survival of the enterprise/business. Assume that UoB wants to drop the cost of email operations by 20% by the next five years; this is the strategic or business goal of UoB. High Level goals are nothing but core goals described in requirements engineering phase for cloud adoption. Suppose UoB desires that the cloud should be able to handle 500 users during the peak hours and provides high level of security- these are high level goals to be achieved. First high level goals will be acquired from UoB that are refined into more objective sub goals until they reach the level of operational goals. Operational goals can interact among each other. Operational goal can be encryption of data, password protection, and backup mechanisms etc. Sometimes, operational goals need to be assessed and evaluated against high level core goals in certain instances.

We will have to define the acceptance level for each goal as these cannot be absolutely met; instead they are satisfied to a certain degree within acceptable limits [15]. The acceptance interval ranges from target level i.e. the optimum value that users consider to be fully satisfied, to the worst level, i.e. the lowest value of the acceptable interval in which the goal starts to be considered unsatisfied. For example UoB wants the response time to be 10 seconds ideally as against 15 seconds which is unacceptable. UoB has set the acceptance limit to 12 seconds which is within the worst and target range. The goals need to be prioritized once they are specified, users have to engage in an extensive prioritization process in order to distinguish core goals (i.e. critical needs that should always be satisfied) from desirable goals (i.e. the ones that could be traded off).

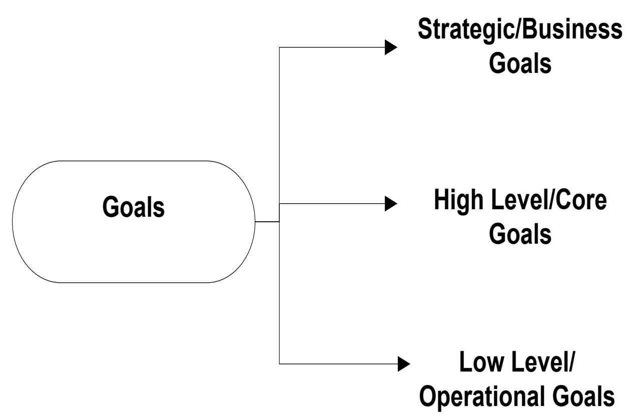

Figure 2. Categories of Goals

4.2 Cloud Service Provider

Once the generic goals have been defined the search for potential cloud service provider will start. At initial stage goals are kept generic so that the search for a cloud service provider is not limited by unnecessary constraints. At initial phase of cloud service provider identification it should be ensured whether the cloud satisfies the critical goals; total cost of ownership match the available budget, reputation of service provider with other users is compared and that the service provider is willing to participate in collaborative partnership etc. The UoB may arrange demonstration sessions to ascertain as to how well cloud satisfies the specified goals. The assessment of cloud service provider's features involves a great deal of uncertainty, where key information to assess the satisfaction of goals can be difficult to obtain. To ensure possible accuracy of the assessment process, each observed functionality must have a confidence degree associated with it.

This confidence degree is based on well justified arguments and evidence that a desired functionality (i.e. operational goal) is sufficiently satisfied. The highest confidence degree is obtained when the goal satisfaction is verified while the lowest one happens when the goal satisfaction is informed. After selection of the cloud service provider the UoB may ask for a trial period which could help verify that a cloud satisfies the requirements and UoB does not have to sign a contract at the beginning without experiencing the services.

4.3 Perform Matching

This phase involves gathering sufficient information about the cloud service provider and its services in order to assign the satisfaction scores to operational goals. Based on the results of matching the evaluation team can aggregate individual satisfaction score (i.e. how the cloud satisfies each operational goal) into global satisfaction scores (i.e. how the cloud satisfies the set of operational goals). Therefore, it is possible to compare cloud service providers and then, inform the decision making process. The matching process involves analysis of cloud service provider satisfaction and further discussion with users to determine whether or not a particular cloud sufficiently satisfies their needs. The matching of a particular cloud is considered satisfactory if the cloud satisfies the operational goals within acceptable range. The degree of satisfaction can come from individual forums like CloudHarmony [26]. Some clouds may offer the user trial period where the user can see the degree of satisfaction of the cloud. For example, for UoB, goal is satisfied if the response time is less than 12 seconds.

4.4 Analyse Mismatches and Manage Risks

A systematic approach is needed for the evaluation team to understand the effects of mismatches, analyse conflicts between goals, explore tradeoffs and manage risks. In the context of cloud service provider's selection, risks are defined as unacceptable outcomes, generally caused as a result of conflicting goals and mismatches. Given that mismatches represent non-adherence of cloud to operational goals and conflict arise when satisfying one goal damages the satisfaction of another goal, we deduce that risks arise when the loss caused by unsatisfied goal is intolerable. A fundamental issue in handling mismatches is the capacity to systematically structure tradeoffs. The tradeoffs analysis forms the basis of risk management strategy. The objectives of risk management strategy are to understand and handle risk events prior to their conversion to threats. Risk management helps in successfully selecting and integrating the chosen cloud service provider. Risk management process has three steps: risk identification, risk analysis and risk mitigation. Risk mitigation action may cover options such as: change goals, negotiate cloud services features, or choose other alternatives. There is no one solution for all the risks; therefore, decision for the best risk mitigation strategy relies on the judgments and experience of the evaluation team. This process is similar to the process *Elaborate the Requirements and Negotiate with the Cloud Service Provider* defined in Figure 1. In this phase UoB will match its goals with the cloud service providers' features and decide which cloud service provider best fulfills the user requirements.

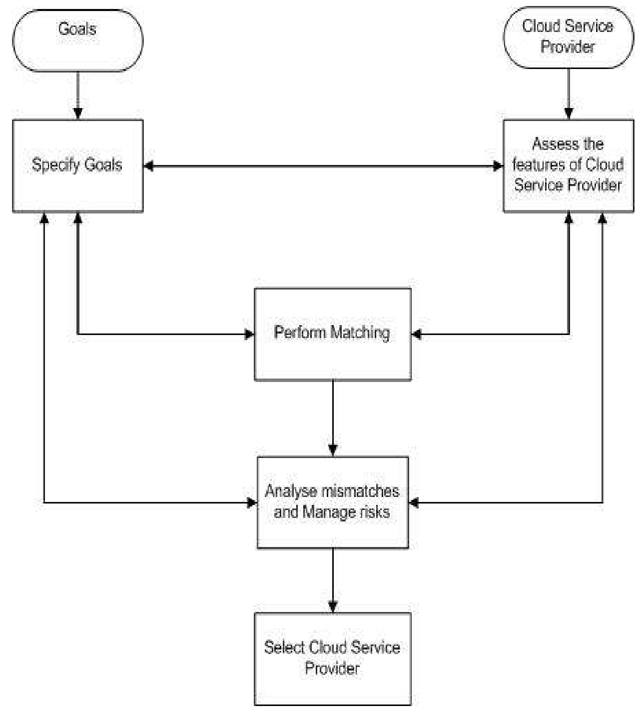

Figure 3. Steps for Cloud Adoption

4.5 Cloud Service Provider

The ultimate objective of this phase is to choose the "optimal cloud". By optimal selection we mean that the selected cloud not necessarily needs to be optimal but satisfying. A satisfying cloud is the one that sufficiently satisfies the set of goals defined by stakeholders, where as an optimal cloud aims to maximize the satisfaction of goals at any cost. Factors to assess the worthiness of each cloud service provider for selection are value, cost, and risk. Cost represent the monetry investment, time and effort. The overall merit of a cloud based system is the combined measure of the value, cost, and risk. Value refers to ascertaining level of happiness of the stakeholders if a given alternative successfully satisfies the desired goals. Notably this is related to the priority stakeholders have assigned to goals. The risk is assessed using goal mismatches, conflicting goals and involves risk scenario elements. UoB during this phase will identify the cloud service provider that satisfy the user goals within the budget constraints.

5. DISCUSSION AND FUTURE WORK

Cloud computing has gained popularity among businesses in recent years. Users are as excited as nervous about using the cloud. They are excited as most of the services provided by the cloud are low cost and readily available. At the same time, in spite of many promises by the cloud service providers, users remain much concerned about the general risk associated with the adoption of the cloud such as security and privacy of the data held, processed and exploited. We have proposed a framework which would help in addressing the genuine concerns of the cloud users. We looked at the case study of a cloud service provider and found that there were indeed many risks associated with cloud adoption. We advocated the necessity of a systematic methodology, for cloud adoption. The methodology can help the users in refining their requirements and negotiating with the cloud service provider while using goal oriented approach for our research. Negotiating and changing the SLA is situation dependent. For large investments cloud service provider may engage into negotiations. There is an interesting clause in Amazon Web Services Customer Agreement which prohibits the user from disclosing the agreement for three years after the end of the agreement: "the nature, content and existence of any discussions or negotiations between you and us or our affiliates" [28]. This statement is clearly indicative of some background negotiations between the cloud service provider and user. Our approach will help the users to identify the conflicts between the requirements and to reduce them. There might be certain obstacles involved in achieving some goals. Obstacles can be any hindrance in achieving any goal or the undesired properties [17]. Obstacle analysis needs to be done in the initial phase of requirements engineering i.e. at the goal level [16]. Obstacles can also be regarded as expression of risks. Our aim is to manage the obstacles in achieving our goals. We intend to evaluate our framework by looking at a case study. The methodology can be complemented by economics-driven approaches for evaluating the cost effectiveness of the decisions and the tradeoffs concerned.

6. RELATED WORK

A comprehensive use of Goal-oriented requirements engineering can be found in [6]. 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 modeling functional requirements [7], non-functional requirements [24], and Agent-oriented systems [18]. For example, Mylopoulos et al used goal oriented approach for eliciting, specifying and refining the non-functional requirements [15]. [18] 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 [19, 20]. One interesting application of GORE, which has inspired our work is that of [21, 22, 23], 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 [21, 22, 23] the problem of cloud adoption is by far more challenging as we are dealing with "open loop "environments, with dynamic, unbounded and elastic scale where continuous service evolution is the norm.

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 [25] and self-optimizing architectures [10] etc. Up to the authors' knowledge, there has been no research on cloud procurement and adoption from requirements engineering perspective. 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. Such helps in managing the tradeoffs associated with matches/mismatches of users' requirements against cloud provision and mitigating likely risks.

7. CONCLUSIONS

We have defined the steps involved in the requirements engineering phase for cloud adoption. This paper contributes to a novel lifecycle, which aims at providing systematic guidance for an organisation evaluating the choice and risks in moving and adopting a cloud. We have used goal oriented approach for eliciting and modeling the requirements of the user. We have presented systematic guidance for the identification and evaluation of cloud service providers. The evaluation process concludes in the selection of potentially best cloud service provider available. Key phase of the method is the matching phase where mismatches between the requirements and cloud service providers' features are identified. The analysis of the mismatches may inform the existence of risks. Our approach advocates risk mitigation in early stages of cloud adoption. The approach also tries to manage the tradeoffs involved with cloud adoption. The Service Level Agreements presently are too static and non-negotiable. The SLAs do not address the individual needs of every user. This framework would help the cloud service provider and the user to negotiate their requirements and cloud features and beyond static SLAs. It would help making SLAs that are specifically designed for a particular organisation.

Our future work will further refine the approach; report on mechanisms for negotiations, risk analysis and mitigation and economics-driven selection. We aim to evaluate the work using case studies, where we will report on the method maturity and steps, applicability, limitations, scalability and so forth. We are also in the process of investigating possible tool support and Web 3.0-like collaborative environments to facilitate the requirements engineering process for cloud adoption.

8. REFERENCES

[1] IBM, "IBM Perspective on Cloud Computing", Whitepaper Published on the Web, ftp://ftp.software.ibm.com/software/tivoli/brochures/IBM_Perspective_on_Cloud_Computing.pdf

[2] Micheal Armburst, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A View of Cloud Computing. In Communications of ACM. Volume 53, No. 4, Pages: 50-58

[3] Micheal Miller. 2008. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online. ISBN:0789738031 9780789738035, Que Publishing

[4] Derek Gottfrid. 2007. Protrated Supercomputing Fun! In The New York Times. 1st November 2007.

[5] Stephen Biggs, and Stilianos Vidalis. 2009. Cloud Computing: The Impact on Digital Forensic Investigations. In Proceedings of Internation Conference for Internet Technology and Secured Transactions. Pages: 1-6, London.

[6] Axel van Lamsweerde. 2001. Goal-Oriented Requirements Engineering: A Guided Tour. In Proceedings of 5th IEEE

International Symposium on Requirements Engineering. Pages: 249-263, Toronto.

[7] Axel van Lamsweerde. 2004. Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice. In Proceedings of 12th IEEE International Requirements Engineering Conference. Kyoto.

[8] Bashar Nuseibeh, and Steve Easterbrook 2000. Requirements Engineering: A Roadmap. In The Future of Software Engineering. Pages: 35-46, ACM Press.

[9] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal. 2008. Market Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. In Proceedings of The 10th IEEE International Conference on High Performance Computing and Communication. Pages: 5-13, CS Press.

[10] Vivek Nallur, Rami Bahsoon, and Xin Yao. 2009. Self-Optimizing Architecture for Ensuring Quality Attributes in the Cloud. In Proceedings of 2009 IEEE/IFIP WICSA/ECSA, Pages: 281-284, CS Press.

[11] Pamela Zave. 1997. Classification of Research Efforts in Requirements Engineering. In ACM Computing Survey (1997), Vol. 29, No.4, Pages: 315-321.

[12] Barry W. Boehm. 1981. Software Engineering Economics. Prentice-Hall (1981)

[13] Annie I. Anton, W. Micheal McCracken, and Colin Potts. 1994. Goal Decomposition and Scenario Analysis in Business Process Reengineering. In Proceedings of 6th International Conference on Advanced Information System Engineering. Pages: 94-104.

[14] Axel van Lamsweerde, Robert Darimont, and Emmanuel Letier. 1998. Managing Conflicts in Goal-Driven Requirements Engineering. In IEEE Transactions on Software Engineering. Vol. 24, No. 11, Pages: 908-926, November 1998.

[15] John Mylopoulos, Lawrence Chung, and Brian Nixon. 1992. Representing and Using Nonfuntional Requirements: A Process Oriented Approach. In IEEE Transactions on Software Engineering. Vol. 18, No.6, Pages: 483-497, June 1992.

[16] Axel van Lamsweerde. 2000. Handling Obstacles in Goal Oriented Requirements Engineering. In IEEE Transactions on Software Engineering. Vol. 26, No. 10, Pages: 978-1005, October 2000.

[17] Colin Potts. 1995. Using Schematic Scenarios to Understand User Needs. In Proceedings of 1st Conference on Designing Interactive Systems: Processes, Methods and techniques. ISBN:0-89791-673-5, Pages: 247-256, ACM Press.

[18] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia,, and John Mylopoulos. 2004. Tropos: An Agent-Oriented Software Development Methodology. In Autonomous Agents and Multi-Agent Systems. 8, Pages: 203-236, Kluwer Academic Publishers. Netherlands.

[19] Daniel Gross, and Eric Yu. 2001. Evolving System Architecture to Meet Changing Business Goals: an Agent and Goal-Oriented Approach. In Proceedings of Fifth IEEE International Symposium on Requirements Engineering, CS Press.

[20] Annie I. Anton, and Colin Potts. 1998. The Use of Goals to Surface Requirements for evolving systems. In Proceedings of International Conference on Software Engineering. Pages: 157-166, CS Press.

[21] Lawrence Chung, and Kendra Cooper. 2002. A Knowledge-Based COTS-Aware Requirements Engineering Approach. In the Proceedings of 14th International Conference on Software Engineering and Knowledge Engineering. Pages: 175-182, ACM Press.

[22] Carina Alves, and Anthony Finkelstein. 2002. Challenges in COTS Decision-Making: A Goal-Driven Requirements Engineering Perspective. In Proceedings of 14th International Conference on Software Engineering and Knowledge Engineering. Pages: 789-794, ACM Press.

[23] Carina Alves, Xavier Franch, Juan P. Carvallo, and Anthony Finkelstein. 2005. Using Goals and Quality Models to Support the Matching Analysis During COTS Selection. In Lecture Notes in Computer Science. Volume 3412/2005, Pages: 146-156, Springer-Verlag Berlin Heidelberg.

[24] Daniel Gross, and Eric Yu. 2001. From Non-Functional requirements to Design through Patterns. In Requirements Engineering. Vol. 6, Number 1, Pages: 18-36, Springer-Verlag London.

[25] Vivek Nallur, and Rami Bahsoon. 2010. Design of a Market-Based Mechanism for Quality Attributes Tradeoff of Services in the Cloud. In Proceedings of the 2010 ACM Symposiumon Applied Computing. Pages: 367-371, ISBN: 978-1-60558-639-7

[26] http://cloudharmony.com/

[27] Kwang Mong Sim. 2009. Agent-based Cloud Commerce. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management. Pages: 717-721, ISBN: 978-1-4244-4869-2

[28] AWS Customer Agreement, http://aws.amazon.com/agreement/#11#11