Handling Dynamic Requirements in Cloud Computing

Ana Sofia Zalazar[1,2], Sebastian Rodriguez[1,2], and Luciana Ballejos[3]

[1] CONICET, Crisóstomo Álvarez 722, 4000, Tucumán, Argentina
[2] GITIA (UTN-FRT), Rivadavia 1050, 4000, Tucumán, Argentina
{ana.zalazar,sebastian.rodriguez}@gitia.org
[3] CIDICI (UTN-FRSF), Lavaisse 610, 3000, Santa Fe, Argentina
lballejos@santafe-conicet.gov.ar

Abstract. Cloud Computing is an Internet-based business paradigm, within which cloud providers offer resources (e.g. storage, computing, network) and cloud consumers use them after accepting the associated agreements. The demand of a particular functionality can rapidly change in this paradigm, so organizations need to count with a method to elicit, analyze, specify, verify, and manage dynamic requirements in a systematic and repeatable way. Existing requirements engineering (RE) approaches for Cloud Computing are generally focused on a limited number of non-functional characteristics (e.g. security, privacy, performance), and service consumers have no guideline to cover multiple dimensions of a requirement in cloud environments. To address this problem, a conceptual model is initially presented to analyze cloud requirements and services. Then, a workflow is proposed for handling those requirements and supporting cloud service adoption. In this paper, we explain our contribution based on practical experience in projects and existing RE approaches.

Keywords: Cloud Computing, Requirements Engineering, Dynamic Requirements

1 Introduction

Cloud Computing is a business paradigm that considers two important roles: provider and consumer. The cloud provider offers his resources as a cloud service (e.g. storage, computing, application, platform, network) and the cloud consumer uses them after accepting the associated agreements (i.e. service level agreements (SLA), terms of use (ToU), terms of service (ToS), user licensing agreements (ULA) and legal contracts).

The National Institute of Standards and Technology (NIST) defines Cloud Computing as *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"* [10].

The success of business in Cloud Computing depends on the requirements specification. Cloud requirements cannot be static and may change over time due to frequent changes in the business and the environment [3]. Moreover, errors made while determining requirements increase exponentially the total cost and affect the business [12], so cloud consumers need structured methods that guide and support requirements management.

Requirements engineering (RE) approaches for Cloud Computing have been considered in the literature [13, 15, 25]. But the existing approaches need to be extended, because Cloud Computing includes dynamic requirements, more dimensions and processes that we consider in this paper.

The goal of this research is to propose an alternative for handling cloud requirements and planning cloud services adoption. The motivation for addressing this goal is the need of a RE method for covering cloud adoption activities [15, 25], especially when the requirements change unpredictably to adapt to new business context.

This paper is an extension of the approach presented in [24] about migration of legacy systems to Cloud Computing, where a conceptual model and a workflow are also proposed. The contribution of our work is part of a project about RE processes in Cloud Computing, where the main objective is to build a framework to manage requirements, to support cloud adoption and to negotiate agreements. This work is based on a model to analyze agreements and contracts by different views (i.e. contractual, financial, service, and control) [22]. In this paper, we redefine views to dimensions, and we consider that requirements can be analyzed by different dimensions (i.e. compliance, financial, contractual, operational, and technical) and each dimension may need a different method for handling its own characteristics. In consequence, cloud agreements and services can be also analyzed by those dimensions.

Based on the works presented in [21] and [25], we assume that goals are more general and stable than consumer requirements. Thus, the goals are specified in early stages of the proposed workflow, and cloud requirements are always adjusted to those goals and the current context, during the business running.

The rest of this paper is organized as follows. Section 2 introduces some insights about RE and existing methodologies considered in our contribution. Section 3 presents the conceptual model and workflow. In Section 4, a sample scenario for using the proposed approach is explained. Finally, Section 5 discuses the final remarks and future works.

2 Background: Requirements Engineering in Cloud Computing

RE is the field of software engineering (SE) dedicated to identify, analyze, specify and validate software requirements [1]. The software requirements represent the needs and constraints considered in the software solution of a real problem.

RE is related to software design, software testing, software maintenance, configurations management, quality assurance and other processes of SE. Pohl in his book [12] considers elicitation, analysis, specification, validation/verification and management of requirements, as RE processes. In addition, Flores et al. assume that RE process for general services involves the next activities [5]:

1. *Requirements Specification:* service requirements are identified.
2. *Requirements Analysis:* requirements are analyzed in detail and possible conflicts between them are examined.
3. *Requirements Validation:* requirements consistency and completeness must be evaluated.
4. *Requirements Management:* it supports all activities and solutions during RE process.

Wieger in [19] classifies RE artifacts in: business requirements, scenarios and uses cases, business rules, functional requirements, quality attributes, external interface requirements, constraints, data definitions, and potential software solutions. Moreover, Pohl specifies goals, scenarios, and requirements oriented to the software solutions, as part of the RE artifacts [12].

The cloud environments are stochastic and dynamic, so it is complex to identify, clarify, and manage cloud requirements in a systematic way [7, 25], especially when services and requirements change in an unpredictable manner. The main causes why cloud requirements change are: (a) organizational policies change the business priorities, so the requirements have to be aligned with the new scope and goals of the organization; (b) environment or marketplace changes by the addition of competitors or new business target; (c) legislation changes may request new forms, features, and algorithms in cloud applications.

The biggest challenge in Cloud Computing is the lack of standards that help meet the objectives covering many different aspects of cloud services [14]. Existing RE processes for Cloud Computing are generally about a limited number of non-functional requirements. Moreover, most of the approaches about cloud requirements are focused on particular characteristics, as security [7], privacy [11], availability, and other performance aspects [2, 3].

Some other authors propose frameworks and methods, but there is no available empirical evidence on the elicitation methods utilized by cloud providers [16]. For instance, Repschlaeger et al. [13] present a framework that includes evaluation criteria to adopt cloud services, and Schrödl and Wind [15] propose a framework to validate established process models for RE in regards to the implementation for Cloud Computing. Schrödl and Wind conclude that none of the common models (V-model, Volere, Extreme Programming, and Rational Unified Process) is suitable to cover the needs of RE under Cloud Computing.

We believe that existing RE processes can be adapted to Cloud Computing and we base our contribution on service-oriented requirements engineering (SoRE) and goal oriented requirements engineering (GoRE).

2.1 Service Oriented Requirements Engineering (SoRE)

SoRE is focused on modeling, specification, and analysis of requirements and constraints under the service-oriented architecture (SOA). It has some common activities of object-oriented RE processes, but SoRE is concentrated in services (no class). Service compositions can be done during the modeling phase or at run-time. The composition, orchestration, choreography during execution are present in cloud environments.

SoRE integrates the next phases to its systematic methodology [5]:

1. *Business Process Modeling:* The objectives and business processes support the goals identified in high-level models, and the tasks of this phase are comprehension of business goals, capture of business process, and capture of strategic information.
2. *Flow-Down Process:* This phase relates business processes considered in the modeling. The analysis flows down until discovering the architecture of business processes.
3. *Formal Requirements Specification:* It establishes requirements, characteristics and business processes to be evaluated in SLAs.

SoRE activities are the dynamic linking and structuring of services in business processes; which change constantly the workflow of business processes, user interfaces, policies, and data configurations [17].

2.2 Goal Oriented Requirements Engineering (GoRE)

GoRE uses business goals and objectives to capture, elaborate, structure, specify, analyze, negotiate, document and trace requirements [20, 25]. The goals are formulated in different abstraction levels according to the functionality and quality of services. GoRE is a promising approach for modeling and reasoning about alternative solutions during early requirements and the alternative ways by which high-level goals can be fulfilled [6].

A goal is a prescriptive affirmation of an intention, that the system or service must satisfy considering stakeholder needs. The goals can have different level of abstraction: high-level goals are related to business strategy and organization, and low-level goals are linked to technical objectives and design options. GoRE allows one to have different versions of the services or software under consideration, so they can be integrated as alternatives into one single solution [25].

Van Lamsweerde [18] analyzes in details this approach. Requirements and models can be derived systematically from defined goals, which propose a way to reason and trace vertically decisions from high-level strategies to technical values. We conceive service goals as an abstraction of cloud requirements in our contribution.

3 Contribution

After the comprehension of the stochastic and dynamic nature of requirements in Cloud Computing (e.g. adaptability, scalability, and rapid provisioning), no simple and standard procedure was found for managing dynamic requirements and matching them to offer of services and SLAs [13]. A procedure that covers all the RE aspects of cloud services, especially when the requirements evolve over time to adapt to new business scenarios, is needed.

In this section, the approach for handling cloud requirements is presented. Initially, a conceptual model about requirements specification is introduced to understand general terms related to cloud domain. Finally, a workflow for managing dynamic cloud requirements is proposed based on existing research [9, 21, 25] and common practices in technology consulting projects.

The workflow is an extensible procedure that supports activities of requirements specification and contracting of cloud services. It also gives organizations the freedom to decompose a goal into multiple requirements and to analyze a requirement in different dimensions.

The conceptual model and the proposed workflow are tools that help organizations and cloud stakeholders to make decisions, analyze their current contexts, schedule main activities and choose the best service.

Since the cloud services can be deployed in third party servers (i.e. "off-premise"), the organizations have the fear of exposing confidential and sensitive data. Thus, the automation of our paper contribution requires a deep evaluation by each organization, because sensitive data and activities may need special consideration in order to comply with particular policies and legal aspects in Cloud Computing [23].

3.1 Conceptual Model

The conceptual model is presented in Fig. 1. This model introduces Cloud Computing from a conceptual level. It shows the most important requirements aspects in Cloud Computing.

In this model, the relation between *Goal* and *Requirement* is clearly represented. A *Goal* is more general and stable than a *Requirement*, and it embraces the main objectives of an *Organization* to complete its business mission. General *Goals* can be defined into three categories: (a)*Strategy*: those goals are long term objectives that focus on the business and the survival of the organization; (b)*Management*: those goals are concerned about users and processes; and (c)*Operational*: those goals are directly related to technology, infrastructure and operations.

Requirements are more susceptible to be modified and *Service Consumers* define them under the needs of the organization. *Requirements* in Cloud Computing often change according to the business context (e.g. peak time, event, target, etc.), legal aspects (e.g. international jurisdiction, copyright, agreements, etc.), and new technologies solutions (e.g. applications, hardware, digital signature, etc.). For instance: the goal "Obtain electronic bank account information" may never change in the organization, however it may specify temporal requirements like "Obtain encrypted information considering law 254/2015 of the National Security code" and "Personal data should be showed on web form 255/2015".

Requirements can be analyzed by different perspectives or dimensions. *Dimensions* are the aspects or factors to be specifically considered in cloud service

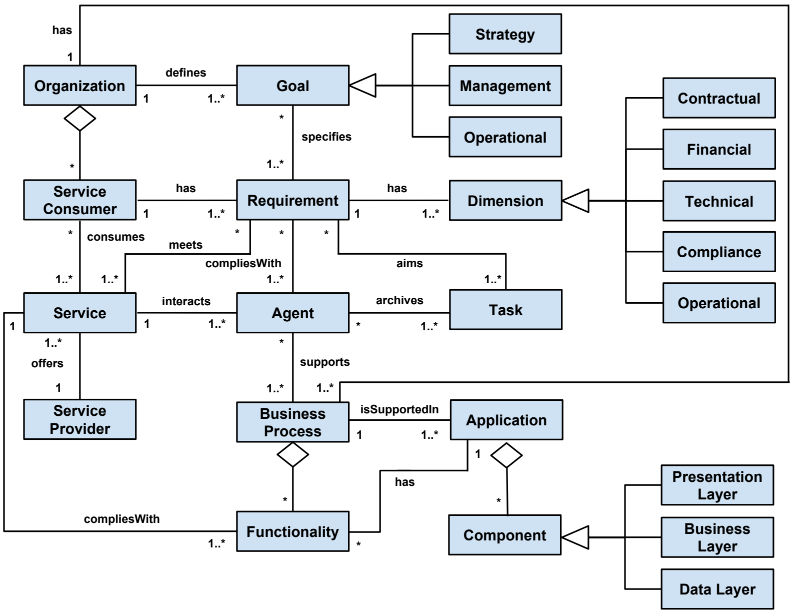

Fig. 1. Conceptual Model of Requirements Specification

adoption. During this research, five core dimensions are identified: (a) *Contractual*: contract trails that specify stakeholders, disclaims, and general agreements between parties; (b) *Financial*: economic aspects of cloud services that are involved in billing, pricing, and return on investment; (c) *Technical*: measurable and technical factors that may need functions, values, constraints, metrics, and units; (d) *Compliance*: regulations that restrict cloud services such as legal, standards, and proceedings; and (e) *Operational*: characteristics that cover specifications about service management, deployment, and access control.

The *Cloud Services* involve *Functionality* that the organization requires into its *Business Process*. The functionality can be part of a whole *Application* or a *Component*. An *Application* basically consists of three layers: (a) *Presentation*: it is the most external layer and where is the user interface; (b) *Business*: it represents the business logic and its functionality implemented by algorithms, software components or other services; and (c) *Data*: it is the information layer containing schema and application data. Dynamic artifacts are required at every layer of Cloud Computing in order to make the services operationally efficient and on demand.

Generally, a *Service Provider* offers *Services* publishing them in the public portfolio. When a *Service Consumer* specifies the current *Requirements*, an *Agent* identifies the available services in the public portfolio to integrate them to the business process execution. Finally, *Service Consumer* uses the services and informs the inaccuracies during the execution.

In Fig. 1, an *Agent* is self-aware and context-aware. An *agent* acts as a broker in charge of the acquisition of *Services* and it archives organizational *Goals* by executing its related *Tasks*. An *Agent* requires to understand the business context (temporal and spatial) requesting unknown information to other stakeholders. The *Agent* role can be played by a software agent or by a domain expert (i.e. developer or consultant) with background in Cloud Computing.

The *Agents* also manage the orchestration or the choreography of cloud services in order to comply with cloud consumer requirements. Thus, the *Agents* are continuously informed about the business environment, and they should only consider updated services and replacing obsolete applications according to current requirements. The activities of this role also include linking processes, modules, and application components.

3.2 Proposed Workflow

Requirements specification is the first activity in conventional software projects, considering waterfall approach. Nevertheless, in this contribution, the first activity is to know the organization high level goals. It is considered that the organization mission is permanent, but requirements change continuously to adapt to new consumer needs, organization contexts, and functionality. Requirements can be functional or non-functional, and they are limited by the general goals to accommodate changes into the organization business scope. Tracing policies, procedures and techniques helps to ensure that consumer requirements and organizational goals are achieved.

The proposed workflow is presented in Fig. 2 and it gives support to the dynamic nature of cloud requirements. This workflow is an extension of the workflow presented in [24], and it indicates methodical and repeatable activities that help organizations and cloud service consumers to iterate and modify requirements during the service acquisition. Four roles (i.e. Organization, Consumer, Agent, and Provider) and twelve activities are considered in this workflow. Our objective to be aimed in future works is to give support to the complete RE process for Cloud Computing, by offering a way to manage requirements in all dimensions and also support cloud adoption, RE processes, and negotiation with cloud providers. In the workflow, we introduce a preliminary contribution about this objective.

1. Define Scope and Goals. The organization should clarify its scope from the beginning of cloud service adoption. The scope describes the final solution driven by business goals, and it represents the desired instance of the organization. The goals are general expressions of the requirements, and the organization can estimate resources and efforts after this high level definition. This activity delivers a document of goals, that gives the proper level of abstraction and bases for future decisions under this approach.

2. Elaborate Specific Requirements. The service consumer should describe the required services, components or applications. The consumer should pay attention to all requirement dimensions at a given time and describe in details some

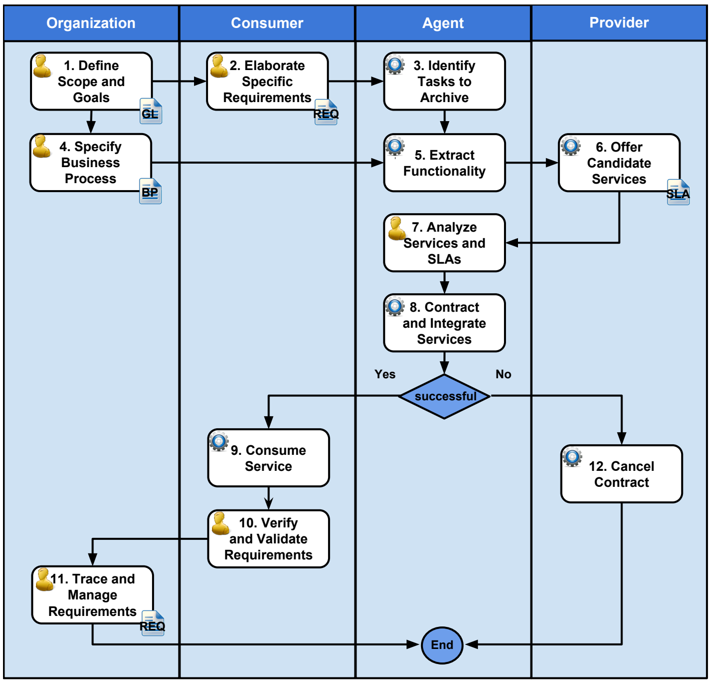

Fig. 2. Workflow for Handling Dynamic Cloud Requirements

specifications about security, privacy, data manipulation, performance and availability under the current organizational environment. Requirements are temporally static in order to provide a level of stability to the agents, and agents must understand requirements considering the delivered comprehensive documentation. The service consumer still needs to do some initial requirements modeling in this activity, because requirements have to be documented to ensure that the service provider offers exactly what is needed by the consumer. In [21] and this paper, we show early stage of cloud requirements specification and some insights about how to elaborate those requirements. In future work, we plan to merge and to integrate the most applicable RE approaches in a single method (Integrate Framework) for requirements specification, because some existing approaches may be focused on just few RE activities and services dimension. The method to be proposed should show relationships and dependencies among the integrated artifacts, in order to pursue adaptability, traceability, and consistency.

3. Identify Tasks to Archive. The agent understands new requirements and changes, and it has the opportunity to explore and discover the possible alternatives into the solution space in a proactive and self-aware manner. The agent constantly adapts itself and adjusts its operations to the changing environmental conditions. It also identifies which operations are good to reach the best solution, which tasks to execute and which requirements to archive. For example, a possibility is that the agent just configures a known deployed service, and other alternative is that the agent collaborates with other agents to reach a new service into the public portfolio. The agent has also an overall view of the organization environment, goals and scope, so it can autonomously encourages service integration, choreography, and orchestration. The agent identifies its tasks at run-time to get better solutions considering the actual contextual knowledge.

4. Specify Business Process. The organization gets insight into its business process by defining and structuring activities in terms of goals to address, rather than tasks to execute. The business process should gives freedom to scale up and down relatively fast according the current requirements. Thus, it should also anticipate environment changes and external events to come up with improvements and to archive high-level goals, after the definition of agent tasks and the service consumers requirements. The organization business process is fixed over time to support the context changes and the needs of service consumers. The organization becomes more context-aware and needs to support this flexibility in managing business process instances and resources. The activities in a process can be either human-based, system-based or a combination of both.

5. Extract Functionality. The agent decomposes the requirements and business process in fine-grain functionality. The functionality is the basic unit to reach cloud services in the service public portfolio and it also helps to consider operations, attributes and resources. In this step, the functionality has to be defined very well so it is easy to match available services to a necessary functionality.

6. Offer Candidate Services. Rimal et al. [14] define three characteristics that cloud services should have: (a) Autonomic: service should be designed to adapt dynamically to changes in the environment to improve the quality, fault-tolerance and security; (b) Self-describing: service should notify the client application how to call it for execution and what data they will return; and (c) Low cost composition: it should provide infrastructure for multi-party interactions and collaboration. However, the service provider offers its services by publishing them in the service public portfolio. The services that support a requested functionality are considered candidate services. To comply with a functionality, it may be necessary to contract one or more services. Thereafter, the agent negotiates with cloud service provider using the SLA documents obtained in this activity. Factors to assess the worthiness of each cloud service provider for selection are value, cost, and risk [25].

7. Analyze Services and SLAs. The agent analyzes the service portfolio, and it puts attention in service functions and the possible alternatives to combine functions. The agent determines the providers' best offer with respect to SLAs and price, by evaluating the cloud agreements and matching obligations against policies. This activity should be supported by particular strategies, formats and protocols. In [22], we present an alternative to evaluate contractual aspects in Cloud Computing.

8. Contract and Integrate Services. During contract negotiation, measurement of quality parameters is defined, and rewards and penalties are considered. The agent firstly selects available resources and it is possible that certain services can be needed upon special conditions. Then, it integrates cloud services to the solution in a dynamic way according to temporal, spatial and quality requirements. The services integration also includes selection of deployment place, installation of the service code, configuration of the service instances and activation of the service to be consumed [9].

9. Consume Service. The consumer should have all the information to invoke one or more services. The cloud service consumer uses the selected resources and services in an automatic manner. The cloud consumer should perform risk analysis and monitor the results of the execution.

10. Verify and Validate Requirements. The verification and validation are conducted in two ways: internally and externally. The internal manner is measured by automatic quality methods (e.g. matching results with expected solution, cases studies, controlled execution, monitoring, etc.). The external manner is conducted by service consumer feedback. The objective is to be sure that the proposed workflow easily allows service consumers to understand, check, and track their requirements.

11. Trace and Manage Requirements. Because the requirements change frequently, a streamline is needed. They should be monitored and traced using some traceability mechanisms (i.e. backward-from, forward-from, backward-to, or forward-to). Consequently, the requirements can undergo changes over the time and are normally covered under change management. A simple change in a requirement implies modifications in the parameters, configuration, and components, and the solution may no longer support the necessary functionality. In our future work, the objective is to give support to the complete RE process for Cloud Computing, by offering a framework to manage requirements in all dimensions and also support cloud adoption, RE processes, negotiation with cloud providers, and requirements management.

12. Cancel Contract. When the service integration or contract is not possible, all the negotiation is canceled and a new workflow iteration takes place. Nevertheless, the service provider can offer new solutions in a new iteration.

4 Sample Scenario: Security Guard Company

To understand how the proposed contribution will be used in practice for cloud environments, we present a brief scenario of a security guard company. The company has recently shown consistent growth, so it aims to reduce the workload of the organization by adopting services for video and data storage in Cloud Computing. The aspects of security, compliance and privacy are directly associated with the performance of cloud services, so those requirements will play a vital role in the organization to develop their business strategies over the cloud.

The company offers an extensive line of security and personal protection services in order to solve the weakness and inadequacies in local and regional security. For this purpose, the organization recruits the most qualified staff and implements the most current technology (software, cameras, servers). The organization has the following purposes that may be supported by cloud services: to monitor clients with video surveillance equipment and to detect emergencies, accidents, and risk presence (*Step 1. Define Scope and Goals* in Fig. 2). The video storage is continuously growing and the company needs the information for audits and clients statistics.

The service consumer, who knows exactly what its organization needs, uses a natural language description (e.g. something similar to the description just provided) as an input to specify the basic needs of monitoring by camera systems. The terms are mapped to the proposed conceptual model in order to understand the cloud requirements context. The proposed model shows relationships that can be associated to additional dimensions. For example, "security certificates", "company policy" and "law regulation" are very important for the security guard company and they are linked to the compliance dimension (*Step 2. Elaborate and Specify Requirements* in Fig. 2).

In this stage, service consumer should also follow the most suitable RE methods considering cloud dimensions and the nature of cloud requirements. For *"Compliance Dimension"*, we find several approaches that may be suitable for analyzing the current scenario. Mouratidis at al. [11] present a methodology that supports just elicitation and security of privacy requirements in Cloud Computing, by the understanding of the organizational context (i.e. goals, actors, tasks, resources, and plan) and the analysis of constraints, threats, and vulnerabilities. Ingolfo at al. [8] extend Nòmos, a modeling methodology for law founded on the concept of norm and situation, which supports modeling of and reasoning with stakeholder preference and priorities. Fabian et al. [4] compare different security RE methodologies (SecureUML, Secure i*, Tropos, KAOS, and SQUARE) and present a conceptual framework with a strong focus on security requirement elicitation and analysis. The service consumer, as a requirements engineer, may combine the approaches to come to the definition of the requirements, because some approaches may be focused on just few RE activities. In our future work, we will present an integrated framework for this purpose in order to integrate relevant requirements methods, activities and concepts (such as norms, threats, vulnerabilities, and situations).

The company relies on the RE cloud deliverable (e.g. supporting documents, models, and descriptions) produced as a result of this stage. This deliverable helps the company to document initial requirements, to understand cloud services, to manage its dynamic requirements, to find inconsistency and risks, to compare cloud offers of different providers, and to contract the best cloud solution according to the company business goals and mission.

The agent organizes its task and extracts the functionality to be implemented in cloud services according to the organization business process and the RE cloud deliverables (*Step 5. Extract Functionality* in Fig. 2). Finally, the agent decides for a storage service to save the video recording under the model pay-per-use, and the company only pays for the use of the provider network in the data transferring and storage unit (*Step 6. Offer Candidate Service*, *Step 7. Analyze Services and SLAs*, and *Step 8. Contract and Integrate Services* in Fig. 2). Then, the service consumer uses the storage for the the video and data recording, and verifies that the service complies with the initial requirements, described also in the SLA (*Step 9. Consume Service* and *Step 10. Verify and Validate Requirements* in Fig.2). The RE deliverable is periodically saved in order to monitor the requirements evolution and changes (*Step 11. Trace and Manage Requirements* in Fig.2).

In this scenario, we show the coherence and practicability of our contribution. The adopted cloud service by the security guard company is one of the most common solution in Cloud Computing, because it is cost effective and easy to implement. The company saves money in infrastructure and it can access locally or remotely to the data recording, just sending some data request to cloud providers.

5 Conclusion and Future Work

Despite of significant advances, there are several open issues in Cloud Computing field and the biggest issue in Cloud Computing is the lack of a standard for requirements specification and service adoption.

In this paper, we presented a structured workflow to handle cloud requirements, based in RE approaches (SoRE and GoRE) and practitioners experience in technology consulting projects.

The contribution presented in this paper is part of a research project about a framework to support dynamic requirements, services adoption, RE processes, and negotiation with cloud providers.

In this paper, we introduced requirements aspects in Cloud Computing and presented a conceptual model to understand dynamic requirements and a workflow to support activities during cloud service adoption. Finally, we concluded with an example about how the contribution can be used in a common scenario.

As future work, we are going to built a requirements specification framework that also supports weakness and risks analysis of cloud specifications (i.e. security, privacy, and data integrity). Therefore, we also aim to define guidelines and recommendations on how organizations should consider the threats associated to Cloud Computing, and how such threats should be addressed within the requirements framework provided.

Acknowledgments. This work is funded jointly by CONICET and UTN. The support provided by these institutions is gratefully acknowledged.

References

1. A. Abran, P. Bourque, R. Dupuis, and J. W. Moore. *Guide to the software engineering body of knowledge-SWEBOK*. IEEE Press, 2001.
2. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
3. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.
4. B. Fabian, S. Gürses, M. Heisel, T. Santen, and H. Schmidt. A comparison of security requirements engineering methods. *Req. Eng.*, 15(1):7–40, 2010.
5. F. Flores, M. Mora, F. Álvarez, L. Garza, and H. Duran. Towards a systematic service-oriented requirements engineering process (s-sore). In *ENTERprise Information Systems*, pages 111–120. Springer, 2010.
6. P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani. Reasoning with goal models. In *Conceptual ModelingER 2002*, pages 167–181. Springer, 2003.
7. I. Iankoulova and M. Daneva. Cloud computing security requirements: A systematic review. In *Research Challenges in Information Science (RCIS), 2012 Sixth International Conference on*, pages 1–7. IEEE, 2012.
8. S. Ingolfo, A. Siena, I. Jureta, A. Susi, A. Perini, and J. Mylopoulos. Choosing compliance solutions through stakeholder preferences. In *REFSQ*, pages 206–220. Springer, 2013.
9. A. Kertész, G. Kecskemeti, and I. Brandic. An interoperable and self-adaptive approach for sla-based service virtualization in heterogeneous cloud environments. *Future Generation Computer Systems*, 32:54–68, 2014.
10. P. Mell and T. Grance. The nist definition of cloud computing. 2011.
11. H. Mouratidis, S. Islam, C. Kalloniatis, and S. Gritzalis. A framework to support selection of cloud providers based on security and privacy requirements. *Journal of Systems and Software*, 86(9):2276–2293, 2013.
12. K. Pohl. *Requirements engineering: fundamentals, principles, and techniques*. Springer, 2010.
13. J. Repschlaeger, R. Zarnekow, S. Wind, and T. Klaus. Cloud requirement framework: Requirements and evaluation criteria to adopt cloud solutions. *20th European Conference on Information Systems*, 2012.
14. B. P. Rimal, A. Jukan, D. Katsaros, and Y. Goeleven. Architectural requirements for cloud computing systems: an enterprise cloud approach. *Journal of Grid Computing*, 9(1):3–26, 2011.
15. H. Schrödl and S. Wind. Requirements engineering for cloud computing. *Journal of Communication and Computer*, 8(9):707–715, 2011.

16. I. Todoran, N. Seyff, and M. Glinz. How cloud providers elicit consumer requirements: An exploratory study of nineteen companies. In *Requirements Engineering Conference (RE), 2013 21st IEEE International*, pages 105–114. IEEE, 2013.
17. W.-T. Tsai, Z. Jin, P. Wang, and B. Wu. Requirement engineering in service-oriented system engineering. In *CEBE 2007, IEEE International Conference on*, pages 661–668. IEEE, 2007.
18. A. Van Lamsweerde et al. Requirements engineering: from system goals to uml models to software specifications. 2009.
19. K. Wiegers and J. Beatty. *Software requirements*. Pearson Education, 2013.
20. S. Wind and H. Schrödl. Requirements engineering for cloud computing: a comparison framework. In *Web Information Systems Engineering–WISE 2010 Workshops*, pages 404–415. Springer, 2011.
21. A. S. Zalazar, S. M. Gonnet, and H. P. Leone. Especificación de requerimientos para sistemas que emplean servicios web en cloud computing. In *1er Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNaIISI 2013)*, 2013.
22. A. S. Zalazar, S. M. Gonnet, and H. P. Leone. Un modelo para contratos de cloud computing. In *42JAIIO - 14th Argentine Symposium on Software Engineering (ASSE 2013)*, pages 303–317, 2013.
23. A. S. Zalazar, S. M. Gonnet, and H. P. Leone. Aspectos contractuales de cloud computing. In *Tercer Congreso Iberoamericano de Investigadores y Docentes de Derecho e Informática (CIIDDI 2014)*, Mar del Plata, Argentina, 2014.
24. A. S. Zalazar, S. M. Gonnet, and H. P. Leone. Migración de sistemas heredados a cloud computing. In *43JAIIO - 15th Argentine Symposium on Software Engineering (ASSE 2014)*, pages 66–79, 2014.
25. 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*, pages 29–35. ACM, 2011.