Bus Inf Syst Eng 60(6):513–524 (2018)
https://doi.org/10.1007/s12599-017-0492-2

RESEARCH PAPER

Insurance for Improving User Satisfaction Level

Hossein Morshedlou · Mohammad Reza Meybodi

Received: 1 June 2015 / Accepted: 30 September 2016 / Published online: 26 July 2017

Abstract Service-level agreement (SLA) violations may lead to losses and user dissatisfaction. Despite the fact that a service guarantee can increase the satisfaction level of users, indemnities may not be commensurate with the importance of a service to a user. While predefined penalties may be insufficient to compensate for the losses of one user, another user may not suffer loss from the SLA violation. With an insurance plan, an insurer can reach an agreement with users on the premium and loss coverage volume; insurance can therefore be considered a solution for providing indemnity which is appropriate to the importance of service. An insurer cannot protect users against these losses, which are caused by a single root event, in the same way as it protects them against the losses caused by independent events. In this paper, a novel approach is proposed for providing insurance coverage for such root events by limiting insurance provisions to the users with the highest priority. A criterion is presented for priority assignment to users, and an algorithm is then proposed for providing insurance according to this priority. A game-theoretic analysis is also provided to assess acceptability of the outcome of the proposed algorithm to rational users and insurers. The results of numerical experiments demonstrate the usefulness of the proposed approach for improving the utility of the service.

Accepted after three revisions by Prof. Dr. Schoder.

Electronic supplementary material The online version of this article (doi:10.1007/s12599-017-0492-2) contains supplementary material, which is available to authorized users.

H. Morshedlou (✉) · Prof. Dr. M. R. Meybodi
Department of Computer Engineering, AmirKabir University of Technology, Tehran, Iran
e-mail: morshedlou@aut.ac.ir

Keywords Risk aversion · Loss · User satisfaction · Learning automata

1 Introduction

A service-level agreement (SLA) is a part of a service contract that contains detailed characterizations of particular aspects of the service such as its quality and the provider's responsibilities. The service provider and the user agree on these aspects (Hani et al. 2015; Linlin and Buyya 2012). Violation of this agreement by the service provider, referred to as an SLA violation, may lead to user dissatisfaction and a reluctance to renew the service contract (Sureshchandar et al. 2002). Numerous studies in the literature aim to offer SLA-based solutions that attempt to minimize the number of SLA violations (Garg et al. 2014; Serrano et al. 2013; Wu et al. 2012). However, in many situations, SLA violations are not preventable, due to unpredictable failures in service provisioning or system errors. Service guarantee and penalty payment approaches (Linlin and Buyya 2012; Wu et al. 2012) are therefore proposed in order to decrease user dissatisfaction. Despite existing techniques for the detection of SLA violations (Aceto et al. 2013; Emeakaroha et al. 2012; Shao et al. 2010), real-world cloud service providers do not use such techniques and leave the task of providing proof of the violation to the users (Baset 2012). This policy may not be satisfactory for a user with a critical or enterprise workload. According to Baset (2012), officially-reported SLA violations receive predefined penalties. However, there are many situations in which users may not report the SLA violations to impose these predefined penalties. There are various reasons for this: an optimistic view suggests that these users did not suffer loss from the SLA violations and

have therefore neglected to report the violations; however, a more pessimistic view suggests that the predefined penalties may not be sufficient to compensate the users for the loss. The latter case may have a destructive effect on the loyalty of users with critical missions. The current penalty payment approaches are not capable of satisfying the requirements of users with enterprise workload (Baset 2012). Baset (2012) has proposed that the service provider should reach an agreement with users on service price and penalty volume, rather than paying predefined penalties. In this way, the price and penalty volumes can be defined appropriately, according to the importance of the service to the user. One possible approach for providing such appropriate penalty or indemnity values is to offer an insurance plan along with the primary service (Bhattacharya and Choudhury 2015; Luo et al. 2010). Users can choose an appropriate insurance plan to protect themselves against possible loss and SLA violations. However, protecting users against events which may cause a large number of simultaneous SLA violations requires the setting of high premiums, which is not acceptable. An approach is therefore required which makes it possible to insure such events for a fair premium. In this paper, a new approach is proposed for providing insurance against such events. The contributions made by this paper are (1) the presentation of an approach for insuring users against unpredictable events which may cause a large number of simultaneous losses; and (2) the presentation of a new criterion for priority assignment to users in order to improve the average utility of the service. This criterion also can be employed for user priority assignment within other problems such as scheduling or resource provisioning. In addition, the numerical experiments carried out here demonstrate the usefulness and applicability of the proposed approach and criterion to cloud computing applications.

2 Related Work

A service-level agreement is a part of a service contract which defines the minimal guarantees offered by a service provider to its users. Particular aspects of the service (for example quality, responsibilities, and delivery time) may be agreed between the service provider and the user in a SLA. A typical SLA, especially within cloud applications, has the following components (Baset 2012):

- *A service guarantee*, which identifies the metrics that a service provider must meet during a service guarantee period. Some examples of service guarantees are availability (e.g., 99%), response time (e.g., less than 100 ms), and fault resolution time (e.g., within an hour). The failure to reach these metrics is known as an *SLA violation*, and may result in loss to the user and/or user dissatisfaction. To restore user satisfaction, the loss should be compensated.
- *A service guarantee period* that determines the interval over which a service guarantee must be met. The time period can be long (e.g., a year) or short (e.g., the duration of a transaction). The smaller the time period, the more stringent is the service guarantee.
- *Service guarantee granularity*, which determines the scale of the service guarantee. For example, the granularity can be defined as per service, per data center, per instance, or per transaction basis. For instance, if the uptime of a running instance must be greater than 99.95%, the service guarantee period determines the interval over which this uptime should be met.
- *Service credit* is the amount credited to the user if the service guarantee is not met. The amount is paid to the user in form of penalties or indemnities to reduce user dissatisfaction.

Service violation detection and reporting determines who is responsible for detecting the violation of the service guarantee and the way in which this violation is reported. The violation of SLAs can damage the loyalty of users over the long term (Linlin and Buyya 2012). A service provider which is incapable of attracting new users and retaining its current users cannot continue in business within oligopoly markets (Allon and Federgruen 2009; Grönroos 2007). According to the service management literature (Bowen and Chen 2001; McDougall and Levesque 2000; Sureshchandar et al. 2002), there are direct links between service quality, user utility, user satisfaction, and user loyalty intention. For example, McDougall and Levesque (2000) showed that service quality and service utility are important drivers of user satisfaction; this study also demonstrated a direct link between user satisfaction and user loyalty intention. This relationship between service quality, user satisfaction, and user loyalty is also confirmed by researches within various service fields such as library services (Sureshchandar et al. 2002) and hospitality services (Bowen and Chen 2001).

Since an SLA violation decreases service quality, it may dissatisfy users and cause them to switch to other service providers. Thus, service providers try to prevent SLA violations. Due to unpredictable failures in service provisioning or system errors, a service entirely free of SLA violations is beyond the bounds of possibility (Emeakaroha et al. 2012). Service providers therefore attempt to compensate for the negative effects of low service quality (SLA violations) by increasing utility through service guarantees and penalty payment approaches (Garg et al. 2014; Wu et al. 2012). These approaches can reduce the degree of

user dissatisfaction (Linlin and Buyya 2012). Since existing approaches to penalty payments do not take into account the importance of services to users and involve the same penalties for all users, they cannot satisfy users with enterprise workloads (Baset 2012). Baset (2012) has proposed that SLAs and penalty values must in the future be flexible and appropriate to the importance of the service for users. One option for providing an indemnity which is commensurate with the importance of the service to users is an insurance plan (Bhattacharya and Choudhury 2015; Luo et al. 2010). Since a premium must be paid for insurance, a user to whom SLA violations are not a high priority may simply prefer to use the primary service, without paying an additional fee for the premium; at the same time, a user who is sensitive to SLA violations may pay an appropriate premium and use the appropriate insurance coverage. There are few existing studies (Bhattacharya and Choudhury 2015; Luo et al. 2010; Naldi 2014) in the literature into providing insurance plans within cloud environments. In Luo et al. (2010), an insurance model covering service guarantee, integrity and QoS is proposed for cloud environments. This work establishes a framework and reference model using a value-at-risk approach to establish several suitable mechanisms, and uses a set of quantifiable metrics; these metrics can be used as the basis for risk assessment. Finally, these metrics are also used to calculate premiums for failure of the services. This work does not discuss risks or events that may cause many simultaneous losses. Certain events, such as a failure within a data center, may cause an unacceptable number of simultaneous losses and SLA violations. According to economic concepts in insurance (Hogarth and Kunreuther 1992), an insurer cannot protect users against losses which are caused by a single event in the same way that it protects them against the losses caused by independent events.

Naldi (2014) proposes an insurance policy subscription as a complementary approach towards protecting users from the economic damage resulting from data unavailability. The work investigates the complementary use of cloud multi-homing and insurance to obtain total risk coverage against data unavailability. However, the approach employed for risk assessment and premium calculation is appropriate only for independent events and risks. Other existing work also treats SLA violations as independent events; this is not always the case in the real world, in which events or risks often cause simultaneous occurrence of many losses. In this paper, an insurance approach is presented which aims to provide coverage for numerous losses with a common origin or root. The insurance approach presented in this paper takes into account the links between user utility, user satisfaction, and user loyalty; this approach increases average utility and, as a consequence, can improve the levels of user satisfaction and user loyalty to some extent.

3 Proposed Approach

This section describes an approach for providing insurance coverage for many losses incurred by a single event. For simplicity, such single events are referred to as common root events (CRE). Although CREs are usually infrequent, they may lead to a large volume of losses. Moreover, due to the complexity of cloud environments, such events are hard to predict. In insurance textbooks, events that are infrequent, unpredictable, and lead to large losses are referred to as catastrophes (Dong et al. 1996). An insurer should take these three features into account when insuring a pseudo-catastrophe event. This paper proposes a method for insuring a CRE with an insurance policy such as (premium, loss), instead of insuring users against these events separately. In this way, the insurer can divide or apportion this insurance policy into policies with smaller granularities. Due to the large number of losses, the insurance of a CRE, which is similar to a catastrophe as defined in the insurance literature, requires a quantification of the loss exceedance probability (LEP) (Dong et al. 1996). In LEP quantification, a LEP curve similar to that depicted in Fig. 1 is obtained. Each point on the curve shows the probability corresponding to a particular loss size (that is, p_0 represents the probability that the sum of all losses is equal to or greater than L_0). For each cloud service provider, LEP quantification requires risk assessment involving detailed information on factors such as resource provisioning policies, workloads, data centers and networking. However, such LEP quantification is not the focus of this work. Using the LEP curve and the capital structure of the insurance company,

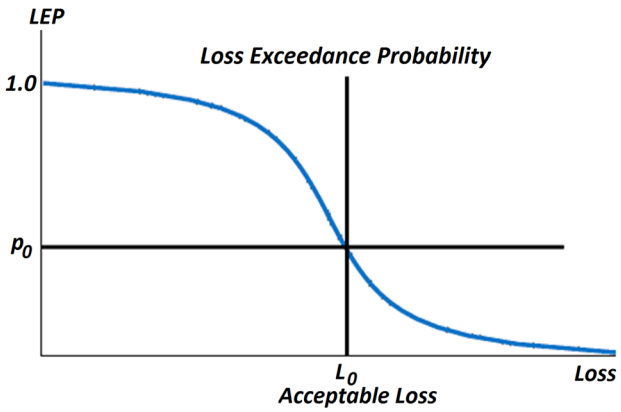

Fig. 1 A sample loss exceedance probability (LEP) curve

the insurer determines the maximum loss it can sustain. If L_0 represents the maximum sustainable loss, the insurer must set a premium pr_0 for insuring L_0. The values of pr_0 and L_0 depend on the status of the service provider (i.e., workload and reserved resources) and the insurer (i.e., premium calculation policy (Zweifel and Eisen 2012)).

To provide insurance coverage for users, the insurer apportions (pr_0, L_0) to k insurance policies with smaller granularity $((pr_1, L_1), \ldots, (pr_k, L_k))$, such that $(pr_0 \leq pr_1 + \cdots + pr_k)$ and $(L_0 \geq L_1 + \cdots + L_k)$. A simple approach to this allocation is to let $(\frac{pr_1}{L_1} = \cdots = \frac{pr_k}{L_k})$. The values $L_1, \ldots, L_k$ constitute the set L, and this represents the finite set of loss or indemnity values, which are obtained by querying users. The insurance company can now provide insurance coverage to k users with (pr_i, L_i) policies $(1 \leq i \leq k)$. If there are n insurers in the system and insurer i provides coverage for $k(i)$ users, then coverage will be available in total for k users, where $k = k(1) + \cdots + k(n)$. In case of n insurers, L is union of their L sets. These insurers have their own insurance policies and risk assessment processes, and may thus offer different policies (premiums and indemnities). The insurer(s) can provide insurance only for k users, and an approach is therefore required for the selection of k users when the insurance service has more than k applicants. It should be noted that the approach presented here does not place any restrictions on k, and that k depends on the capital structure and abilities of the insurer. In this approach, the insurers select the k users according to priorities assigned by the service provider. A criterion is therefore presented in Sect. 3.1 for the assignment of priority to users. Following this, an approach for pairing the k insurance policies to the k selected users is presented in Sect. 3.2.

3.1 Introducing a Criterion for Priority Assignment

In this section, a criterion is introduced for the assignment of priority to users which is referred to as risk aversion. A user is said to be "risk averse" if when confronted with two choices with the same expected utility, the user prefers the smaller and more certain of the options. The utility function of a risk averse user has two main features: $u' > 0$ and $u'' < 0$. The former condition represents a user's preference for higher amounts of money and wealth over smaller amounts. This assumption seems perfectly rational. The latter represents the effect whereby as a user's wealth increases, he or she places less value on a fixed increase in wealth. Given a particular utility function, the user's risk aversion can be calculated by $-u''/u'$. The assumption that human beings behave in a risk-averse manner is plausible and in accordance with socio-biological arguments (Zweifel and Eisen 2012).

This work proves that an approach whereby a service provider assigns a higher priority to more risk-averse users will improve the user satisfaction level (USL). USL is defined based on average utility of users as shown in Eq. (1):

$$USL(t) = \sum_{i \in Users} u_i(w_i(t))/number\_of\_users. \quad (1)$$

The utility function, $u(w)$, measures the utility that a user attaches to the monetary amount w. The term $w_i(t)$ in Eq. (1) represents the monetary profit of user i using the service at iteration t. $w_i(t)$ depends on various parameters such as price of the service and whether or not the user has encountered an SLA violation. For example, when it is likely that a user will experience an SLA violation, the utility of the user decreases to $u_i(w_i(t) - x)$, where x is a random variable whose possible values are 0 in normal conditions, and L in case of an SLA violation. L is the magnitude of economic damage or loss in case of SLA violation.

Proposition 1 *Consider a situation that n similar users have requested insurance. They are similar in every characteristic but risk aversion. When insurers can service only m requests* $(m < n)$, *choosing the more risk-averse m users maximizes USL.*

Proof See Online Appendix A (all appendices available via http://link.springer.com).

A method is now required for the estimation of risk aversion. It has been shown in the insurance literature that a more risk-averse individual has more willingness to pay (WTP) for an insurance premium (Zweifel and Eisen 2012); however, decision making on accepting/rejecting a premium is not always accurate or identical (Einhorn and Hogarth 1988). A user may accept differing values of premiums at different times. Moreover, when a user decides to accept or reject a premium, it is logical to assume that the user considers the proposition of a 1-year contract more closely than the proposition of a 1-h contract. When a user considers a decision more fully, the result better reflects the importance of the service for the user. However, since users are not fully rational (Simon 1982) and information-complete (e.g., unpredictable events may occur), a greater degree of consideration sometimes cannot guarantee that a user will make a better decision. In addition, the length of the SLA (e.g., 1 h vs. 1 year) has a direct relationship with the number and likelihood of SLA violations. Therefore, the users' perception of risks changes with the time horizon. This compels the consideration of several other characteristics of the users [e.g., myopic loss aversion (Benartzi and Thaler 1995)]. Since the focus of this paper is on risk aversion, it is assumed for simplicity

that all users are similar in every characteristic except risk aversion, and that all SLAs have the same length. It should be noted that there is no limitation on the length of the SLA, and that this approach is applicable to any length. When a user must decide frequently on the acceptance or rejection of premiums, the high level of disparity in these decisions shows that the importance of a service cannot be estimated accurately according to these decisions (i.e., the user will accept a high premium at one time, and reject a low premium at another). In view of these factors, a judgment of risk aversion of a user based on a single decision is not accurate, and the average of these decisions over time gives a more precise result. This paper uses the estimation of risk aversion based on a large number of decisions, and is therefore relatively accurate and reliable. Since the decision making of a user is not deterministic, and a user who accepts a premium value at one time may reject the same premium value at another, a learning tool is therefore required which is capable of observing the decisions of a user over several iterations and learning the expected value. A learning automaton is an adaptive learning unit (Narendra and Thathachar 2012) for a random and stochastic environment which is capable of learning through repeated interactions. The choice of action is carried out using a probability vector. By carrying out an action, the automaton receives a response from the environment, and its probability vector is updated according to this response. Let $\alpha_i(k) \in \alpha(k)$ denote the selected action by the learning automaton based on the probability distribution $p(k)$ defined over the action set at instant k. The variable r denotes the number of actions that can be taken. If the selected action $\alpha_i(k)$ receives a reward, then the probability vector $p(k)$ is updated using Eq. (2). If it receives a penalty, Eq. (3) is used instead. The variables a and b are learning rates, which are associated with the parameters of the rewards and penalties. If the learning rate is too low, the convergence may be too slow, and if this rate is too high, the precision may be too low:

$$p_i(n+1) = \begin{cases} p_i(n) + a[1 - p_i(n)] & i = j \\ (1-a)p_i(n) & \forall i, i \neq j \end{cases}, \tag{2}$$

$$p_i(n+1) = \begin{cases} (1-b)p_i(n) & i = j \\ b/(r-1) + (1-b)p_i(n) & \forall i, i \neq j \end{cases}. \tag{3}$$

If $(a = b)$, this learning algorithm is called the linear reward-penalty, or L_{R-P}. If $a \gg b$, it is called the learning reward-ε penalty or $L_{R-\varepsilon P}$, and if $(b = 0)$, it is called the linear reward-inaction algorithm (L_{R-I}). A learning automaton is an appropriate choice for learning within stochastic environments, and has been successfully used in a behavioral model of students (Oommen and Hashem 2010) and various patterns (Barto and Anandan 1985). A learning automaton is therefore utilized here in order to learn the expected value of the premiums that a user accepts. The number of actions in learning automata can be an arbitrary value; however, the values of the first and last actions are represented by $pr_{\min}$ and $pr_{\max}$, respectively. The values $pr_{\min}$ and $pr_{\max}$ are the minimum and maximum values of the premiums pr_i of the (pr_i, L_i) policies $(1 \leq i \leq k)$, where $L_i \in L$. Moreover, for each j $(1 < j < r)$, the corresponding value of the $(j+1)$th action is greater than the corresponding value of the j th action. For each user, the service provider stores the probability vector of the learning automata $(p(n))$ within the user's profile. When the user accepts or rejects the insurance policy (pr_x, L_x), the service provider updates the probability vector of the learning automata in the user's profile, according to the algorithm in Fig. 2. A long-term measure (M_{LT}) is defined using the probability vector of the learning automata as shown in Eq. (4).

$$M_{LT} = \sum_{i=1}^{r} (p_i \times pr^i), \tag{4}$$

M_{LT} represents the expected premium paid by a user. According to insurance economic concepts, the amount of premium paid by a user has a positive correlation with

Fig. 2 Learning algorithm of a learning automaton

Let $p(n) = (p_1(n), p_2(n), \ldots, p_r(n))$ *denote the probability vector of a learning automaton with r actions at iteration n.*

Let $pr = (pr^1, pr^2, \ldots, pr^r)$ *denote the vector of the corresponding values of the r actions of the learning automata.*

When a user accepts or rejects the insurance policy (pr_x, L_x) :

The service provider finds pr^j, which is the closest value to pr_x within the premium values $(\{pr^1, pr^2, \ldots, pr^r\})$

The service provider updates $p(n) = (p_1(n), p_2(n), \ldots, p_r(n))$ *as follows, in the case of acceptance:*

$$p_i(n+1) = \begin{cases} p_i(n) + a[1 - p_i(n)] & i = j \\ (1-a)p_i(n) & \forall i, i \neq j \end{cases}$$

and updates $p(n) = (p_1(n), p_2(n), \ldots, p_r(n))$ *as follows in the case of rejection:*

$$p_i(n+1) = \begin{cases} (1-b)p_i(n) & i = j \\ b/(r-1) + (1-b)p_i(n) & \forall i, i \neq j \end{cases}$$

user's risk aversion. Therefore, M_{LT} can be considered to represent risk aversion. A further discussion of the correlation between risk aversion and the premiums paid by a user is given in Chapter 2 of Zweifel and Eisen (2012).

3.2 Pairing Process

In Sect. 3.1, a criterion was introduced for the assignment of priority to users. Since the insurers can provide insurance for only k users, this criterion is required to assign priority to these users. At the start, the service provider asks the users to determine their required insurance coverage and the maximum premium they want to pay for this coverage. According to the replies from the users, the service provider assigns preliminary priorities to the users. This procedure is also used to generate a preliminary priority for each new user who joins the current users. At the first iteration, in order to implement the assignment of priorities, the k insurance policies are offered to the first k users with the highest priorities. If k' insurance policies are rejected by the k users, these will be offered to the next k' users with highest priorities at the next iteration and so on. At each iteration, the priority of a user may change based on the estimated risk aversion for that user. If a user has set a low premium at the start, and later realizes that it is not possible to obtain insurance coverage at the proposed premium, it is possible to revise this decision in order to increase the user's priority. Now, the remaining issue is that of how k insurance policies should be paired with k users. When two or more users are interested in one insurance policy, one of them must be selected. Moreover, users are autonomous and have the authority to reject an insurance offer, and it is therefore not certain that all of the k users will accept all k insurance policies. Thus, a pairing process is proposed below which pairs users with insurance policies. In addition, a game-theoretic analysis is presented to demonstrate the acceptability of the outcome of the pairing process between rational users and insurers. This pairing process never pairs an insurance policy x with user y when there is an unpaired user who is more risk averse than user y and is interested in insurance policy x. Since this pairing process pairs the insurance policies to the most risk-averse users interested in the offered insurance policies, it therefore maximizes the USL as far as possible, according to Proposition 1.

In the pairing approach introduced here, there are two sets of brokers: the user's brokers (uBrokers) and the insurer's brokers (iBrokers). In an iterative process, when receiving an insurance request from a user, the service provider initializes a uBroker and assigns the request to the uBroker. The uBroker is a user profile-aware agent, which tries to maximize the user's utility against the insurers. On the other side there are iBrokers, which are policy-aware agents of the insurers. Each insurer may have many iBrokers. Each iBroker contains one insurance policy, which is loaded by the insurer. At the beginning of each iteration, the insurer asks the service provider for certain information, such as current workload and the reserved or available resources, to estimate the probability of loss or SLA violation. It then generates the insurance policies (apportioning process) and loads them to the iBrokers.

At each iteration, with k uBrokers and k iBrokers, a uBroker sends its request (which contains a specific indemnity value from L) to the iBrokers and receives their proposals for premiums. Some iBrokers may not respond to the request due to the impossibility of providing insurance for the received request. In other words, if the requested indemnity is more than the specified indemnity of the insurance policy loaded to that iBroker, the iBroker will not respond to that particular request. Moreover, the premium proposed by an iBroker should not be smaller than the premium specified in the insurance policy of that iBroker. A rational uBroker prefers to reach an agreement with the iBroker who has proposed the minimum premium. Thus, using the received proposals, each uBroker creates its own iList, which is an ordered list of iBrokers based on their proposed premiums. On the other side, between two uBrokers, an iBroker prefers to insure the uBroker whose corresponding user has a higher priority for the service provider. This means that all iBrokers have an identical list of uBrokers (uList), which are ordered based on the users' priority for the service provider. In the following, these lists (iLists and uList) are referred to as the favorite lists, or favorites in brief. Note that each iBroker can contract with only one uBroker and vice versa. Figure 3 shows the proposed algorithm for pairing iBrokers and uBrokers. Using a game-theoretic analysis, it is shown that the outcome of this algorithm will be acceptable for rational users and insurers. Before presenting the game theoretic analysis, some preliminary definitions and lemmas are presented.

Definition 1 The favorite lists of brokers are cycle-free if and only if no wrap-around sequence of brokers $b_1, b_2, \ldots, b_k$ (k is even and $k > 2$) exists such that each broker b_i prefers b_{i+1} to b_{i-1} (if $i = k$ then replace $i + 1$ with 1). Notice that in $b_1, b_2, \ldots, b_k$, brokers b_{i-1} and b_{i+1} both have the same type for each i (either uBrokers or iBrokers) and differ from b_i.

Lemma 1 *If iBrokers arrange their favorite lists based on the priority of users, regardless of the favorites of the uBrokers, the obtained favorites are cycle-free.*

Proof See Online Appendix B.

Definition 2 The sets of uBrokers and iBrokers are pairable if in each iteration of an iterative procedure, two brokers (one uBroker and one iBroker) can be found which

```
1.  The brokers create their favorites lists (k iLists and one uList)
2.  An iBrokerList is created which contains all iBrokers (No need to sort this list)
3.  while(uList is not empty) {
4.          uBroker = uList.get_First_uBroker();
5.          iList = uBroker.get_iList();
6.          iBroker = iList.get_First_iBroker();
7.          while (iBroker is not in iBrokerList) {
8.                  iList.Remove_First_iBroker_From_List();
9.                  iBroker = iList.get_First_iBroker();
10.         } // end of While(iBroker is not in iBrokerList)
11.         pair(uBroker , iBroker);
12.         uList.Remove_First_uBroker_From_List();
13.         iBrokerList.Remove_From_List(iBroker);
14. }  // end of while(uList is not empty)
```

Fig. 3 Pseudo-code for the pairing algorithm

prefer each other to all the other existing brokers from the opposite type in the set. These two brokers are eliminated from the set for the following iterations. After the last iteration, the set is empty or contains brokers with the same type.

Theorem 1 The *set of uBrokers and iBrokers is pairable.*

Proof See Online Appendix C.

Theorem 1 proves that sets of brokers are pairable. The pairing algorithm in Fig. 3 gives the procedure for this pairing. Since each broker acts as an agent for either a user or an insurer, it must try to maximize its utility. The aim is then to verify whether or not this pairing satisfies the rational users and insurers. For this verification process, the following discussion of this pairing is provided from a game-theoretic viewpoint. The situation is described in the form of two games: uGame and iGame. Let $r_i(j)$ denote uBroker/iBroker j's rank in the favorite list of iBroker/uBroker i. S is the set of all strategy profiles that players can select and s_i denotes the strategy of player i.

uGame and iGame

uBrokers are players of uGame and iBrokers are considered to be part of the environment. Since brokers of both types are rational, they therefore choose the best strategy. In uGame, the strategy space of the uBrokers is a set of actions in which each action is equivalent to choosing a specific iBroker. For each strategy profile $s \in S$, the utility of uBroker i is $u_i(s) = n - r_i(j) + 1$ if and only if $s_i = j$ (choosing iBroker j) and there is not another uBroker k $(k \neq i)$ such that $s_k = j$ and $r_j(k) < r_j(i)$; otherwise, $u_i(s) = 0$. The definition of iGame is similar to uGame, although here, the iBrokers are the players of the game.

Game-Theoretic Analysis

Theorem 2 *The outcome of the pairing algorithm is a pure Nash equilibrium point of uGame and iGame.*

Proof See Online Appendix D.

Theorem 3 *Both iGames and uGames have a unique pure Nash equilibrium (PNE) point.*

Proof See Online Appendix E.

According to Theorems 2 and 3, the outcome of the pairing algorithm is equivalent to the unique pure Nash equilibrium point of iGame and uGame. In games with a unique pure Nash equilibrium point, playing the best response strategy converges to that unique PNE (Nisan et al. 2011). This means that the outcome of the playing of the best response and the pairing algorithm are equivalent. Therefore, the result of the pairing algorithm satisfies the condition of rationality of uBrokers and iBrokers.

3.3 Applications of the Proposed Approach

The approach proposed in this paper can be employed for providing insurance coverage for various risks or events which are the roots of simultaneous losses for many users,

such as resource provisioning failures, particularly the facing of unexpected loads (Javadi et al. 2012), insecure or incomplete data deletion (Catteddu 2010), a malicious insider (Catteddu 2010), users' security expectations (Catteddu 2010), compromise of the management interface (Catteddu 2010), and isolation failure (Catteddu 2010), to name just a few examples. To employ the proposed approach, the following questions should be considered:

1. What can happen (i.e., what can go wrong)?
2. How likely is it that it will happen (i.e., probability estimate)?
3. If it does happen, what are the consequences (i.e., impact estimate)?

Probabilistic risk assessment approaches can be employed to answer these questions. Using the answers to these questions, an LEP curve can be produced for the risks. Following this, the insurer(s) should determine the maximum loss it (they) can sustain. Finally, the insurer(s) should generate insurance policies (pr_x, L_x), which provide different levels of coverage (L_x) for the users. One alternative for defining indemnities (L_x) which are appropriate to the importance of service to users is to log the indemnities requested by users and revise the insurance policies over the following iterations if necessary.

4 Numerical Experiments

In this section, numerical experiments are conducted to show the benefits of the proposed approach. For this purpose, a user model is first defined in Sect. 4.1, which is employed to simulate a user within the numerical experiments. The definition of such models is usual in the economic literature (Allon and Federgruen 2009). Numerical experiments are also carried out in Sect. 4.1 to evaluate whether the behavior of the proposed user model corresponds to the behavior of a rational user in the real world. Since ranking users based on their risk aversion plays a key role in the approach proposed here, a learning automaton-based method is presented for creating ordered lists of users. In Sect. 4.2, the capability of a learning automaton to learn a value for M_{LT} is first evaluated. M_{LT} is used to create ordered lists of the users. This ordered list is then used to select a subset of the users to whom insurance will be offered. The average utility of the subset of users chosen in this way is compared with the case where the subsets are selected randomly. The results show that the proposed approach maximizes the average utility of the users as far as possible, and as a consequence improves the user satisfaction level.

4.1 User Model

Equation (5) shows the utility function of a user within the user model presented here. This function satisfies the requirements for the utility function of a risk averse user ($u_i' > 0$ and $u_i'' < 0$). The value w_i represents the user's asset and c_i is a constant. Since the risk aversion of a user can be calculated using $-u''/u'$ (see Sect. 3.3), then in the presented model, a_i is the risk aversion of a user.

$$u_i(w_i) = c_i \exp(-a_i w_i) - c_i \quad (c_i < 0,\ a_i > 0). \tag{5}$$

Let (pr, x) denote an insurance policy where pr is the premium for insuring a loss x. The user will decide whether to accept or reject this insurance policy. In the model presented in this paper, the utility function is used for decision making. As illustrated in Fig. 4, the utility function of a risk-averse user $(u(w))$ is an increasing and concave function. Under normal conditions, the utility of a user is $u(w_0)$. However, if there is an SLA violation (e.g., service unavailability or low service quality), the utility of the user decreases to $u(w_0 - x)$. Now let the probability of an SLA violation or abnormal conditions be p. The expected utility of the user in such risky conditions is $p \times u(w_0 - x) + (1 - p) \times u(w_0)$, which is equal to $u(w_s)$ as shown in Fig. 4. Due to this equality, the premium $(w_0 - w_s)$ makes a risk-averse user indifferent to the choice between a certain asset w_s and one with a risky asset w_0 (which may decrease to $(w_0 - x)$). A fully rational user with complete information accepts any premium smaller than $(w_0 - w_s)$ and rejects one greater than $(w_0 - w_s)$.

Since users in the real world are not fully rational (Simon 1982) and may make errors in decisions, a level of noise is therefore added to the decision of the user in this model. For simplicity, it is assumed that the user accepts the premium $(w_0 - w_s)$ with probability 0.5 (since at this value of the premium, the user is indifferent to the choice between certain and risky conditions) and this probability

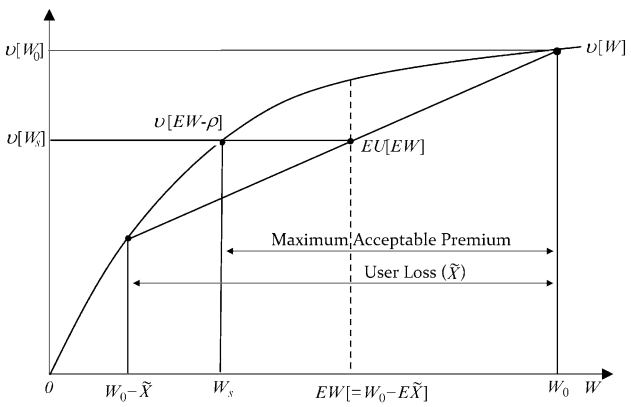

Fig. 4 Utility function of a risk-averse user

increases as the premium decreases. Equation (6) shows the simple equation used in the user model to determine the acceptance or rejection of an insurance policy:

$$p_{accept} = \frac{2(w_0 - w_s) - premium}{2(w_0 - w_s)}. \quad (6)$$

To evaluate the proposed user model, insurance policies are offered to five different users, and this is modelled using the proposed method.

These users are similar in every characteristic except risk aversion. For all users, $w_i = 1$ and $c_i = -100$. The risk aversion (a_i) is 1, 3, 5, 7 and 9 for user types 1, 2, 3, 4 and 5, respectively. The insurance policies offered to these users are (0.01, 0.1), (0.01, 0.5), (0.1, 0.8) and (0.2, 0.8). The first and second insurance policies differ in terms of coverage volume, and the third and the fourth policies differ in terms of premium. The vertical axis in Fig. 5 shows the probability (p_{accept}) of a particular user type accepting an insurance policy. The horizontal axis shows the probability of incurring loss x (p_x). According to Zweifel and Eisen (2012), a risk-averse user is more likely to accept an insurance policy (pr_i, x_i) than a less risk-averse user. As shown in Fig. 5, the behavior of the proposed user model corresponds to that in the real world. Moreover, these figures show that when pr_i has a negative correlation with p_{accept}, x_i has a positive correlation with p_{accept}. This means that the behavior of the proposed user model also corresponds to the behavior of a rational real user in this respect. Moreover, according to these diagrams, when the insurance coverage is non-trivial and p_x has a small value (e.g., $p_x < 0.1$), the probability p_{accept} shows significant differences between different user types. As p_x increases, the differences tend towards zero.

Since the probabilities of loss in real services are usually very small, p_{accept} therefore shows significant differences for different user types. In view of these differences, the probabilities of accepting insurance policies are expected to be closely related to the risk aversion of users.

It should be noted that the service provider is unaware of the user's utility function and decision-making method; it simply observes the acceptance or rejection of an insurance policy by a user and estimates the risk aversion of a user based on these observations.

4.2 Evaluations and Results

In this section, the capability of a learning automaton to learn M_{LT} is presented. The evaluation involves five users, modelled with the user model described above. A set of insurance policies, {(0.01, 0.1), (0.01, 0.5), (0.03, 0.3), (0.03, 0.5), (0.05, 0.5), (0.1, 0.5), (0.1, 0.8), (0.2, 0.8)}, are offered to the users in an iterative procedure. A learning automaton with five actions is applied for each user. The corresponding values for actions 1–5 are 0.01, 0.03, 0.05, 0.1 and 0.2, respectively. When an insurance policy is offered to a user, it is either accepted or rejected. Based on the user's response, the probability vector of the learning automata is updated according to the algorithm shown in Fig. 2.

Figure 6a illustrates the evolution of the probability vector of the learning automaton over 1000 iterations for a

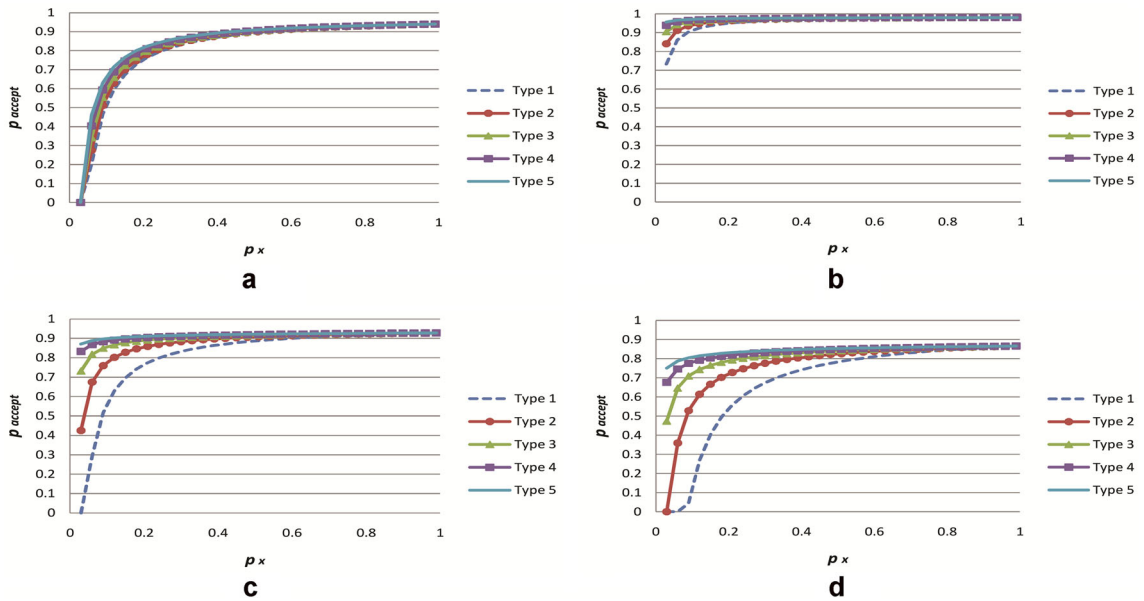

Fig. 5 Comparison of acceptance probabilities of insurance policies by different user types: **a** (0.01, 0.1); **b** (0.01, 0.5); **c** (0.1, 0.8); **d** (0.2, 0.8)

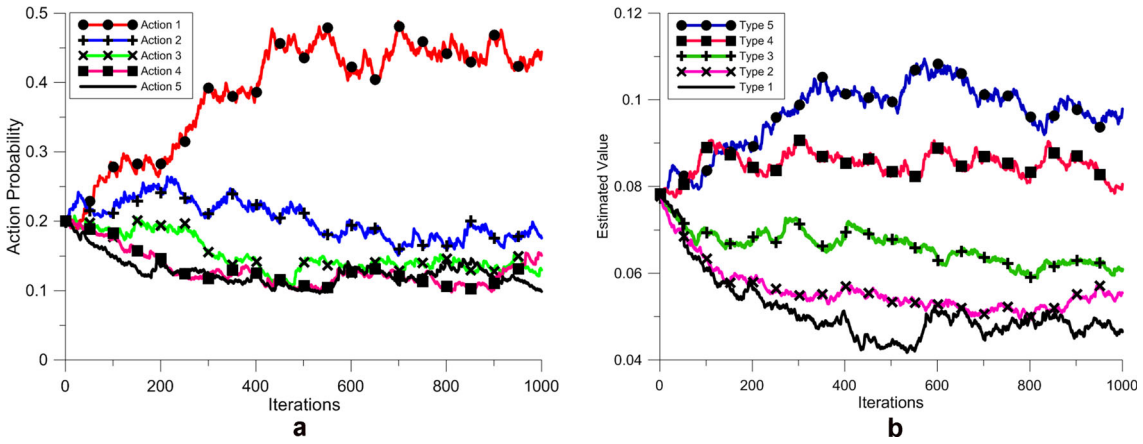

Fig. 6 **a** Evolution of the probability vector of a learning automaton when employed for learning MLT for user type 1; **b** the learnt MLT for different user types

user of type 1. The learning rate of the algorithm L_{R-P} [a and b in Eqs. (2) and (3)] in this experiments is 0.01. Figure 6b illustrates the learned values for M_{LT} using Eq. (4) for the different user types. As illustrated in this diagram, the value of M_{LT} has a positive correlation with the risk aversion of a user. For user selection purposes, only an ordered list of users based on their risk aversion is required, rather than the exact values of risk aversion; therefore, M_{LT} can be used as a criterion for creating an ordered list of users according to their risk aversion. In order to evaluate the impact of the proposed approach on the average utility of the users, it is assumed that there are 500 users, modelled using the user model described in Sect. 4.1. These users have the same parameter values and differ only in the degree of risk aversion. The values for risk aversion for these users are {0.02, 0.04, 0.06, …, 9.96, 9.98, 10}. For simplicity, it is assumed that in this iterative scenario, all users are interested in buying insurance, but that it is not possible to provide insurance coverage for all of them and that a subset of users must therefore be selected. The average utility of users (USL) is then compared in two different modes: a random mode, in which members of the subset are selected randomly from the users, and a risk aversion-based (RA-based) mode, in which members of the subset are selected from ordered lists and are the most risk-averse users.

Figure 7a illustrates the average utility of users in these modes. The horizontal axis shows the possibility of losses,

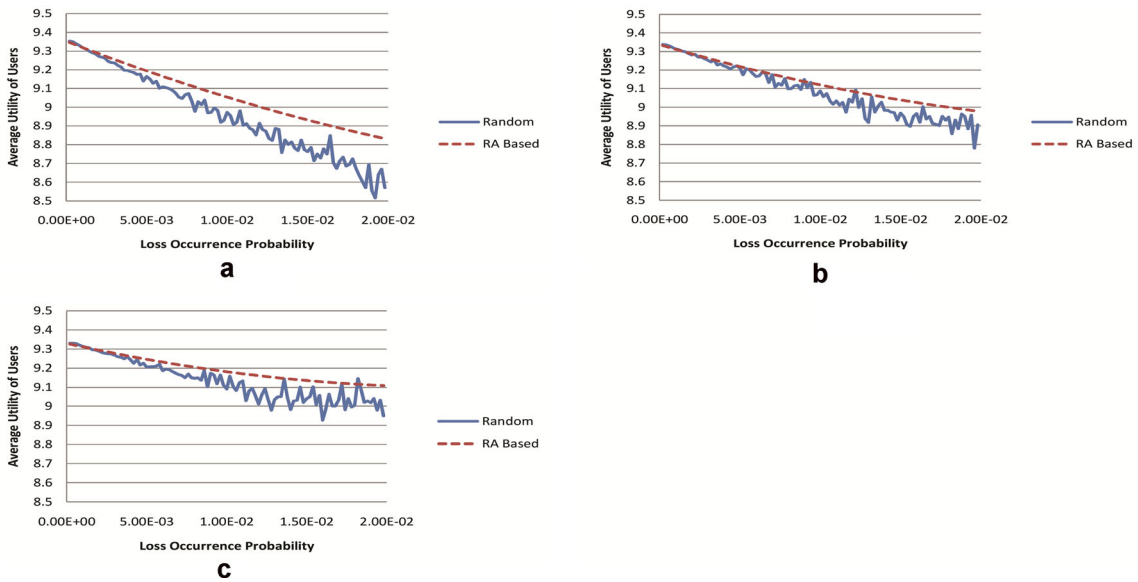

Fig. 7 Comparison of the average utility of users (USL) in random and RA-based modes, when the percentage of the insured users is **a** 25%; **b** 50%; **c** 75%

Table 1 Results of t test on the average utility of users using random selection vs. risk aversion-based selection

Insurance coverage (%)	*p* level (%)	*t* score
25	0.01	3.88
50	0.78	2.69
75	0	4.81

which has a small value in [0, 0.02]. Small values (interval [0, 0.02]) are selected since losses are infrequent in the real world. The vertical axis shows the average utility of users when only 25% of the users can buy insurance. As illustrated by this figure, users can always obtain better average utility in the RA-based mode, and this difference increases as the possibility of losses increases. Figure 7b, c shows the same diagrams for the cases where 50 and 75% of users have the opportunity to buy insurance. As illustrated by these figures, users in these cases also always obtain better average utility in the RA-based mode. To analyze the difference between the average utility of the users in both modes, a statistical *t* test is employed over the obtained average utilities. Table 1 shows the obtained p level values. The p level is the significance level of the difference between average utilities in random and RA-based modes. The difference in average utility between the two modes is considered to be significant if the p level is less than 5%. Since the p levels for the 25, 50 and 75% cases are 0.01, 0.78 and 0%, respectively, these differences are therefore significant. For reference, if the p level is 1%, there is a 1 in 100 chance that this difference is produced by chance. The results of the t test (t score) for the 25, 50 and 75% cases are 3.88, 2.69 and 4.81, respectively.

5 Limitations and Conclusion

In this paper, an approach is presented for providing insurance coverage for events which may lead to many simultaneous SLA violations and losses. According to the insurance literature, an insurer cannot protect users against many losses caused by a single event in the same way that it protects them against the losses caused by independent events. To provide insurance coverage for such events, the insurer should set a premium which is high enough not only to cover the expected losses but also to protect itself against the possibility of experiencing catastrophic losses. Setting high premiums in environments such as the cloud is not appropriate and increases the total cost of service. The existing studies on providing insurance in cloud environments have not considered these correlated losses and SLA violations. Since such events are infrequent, this feature can be used to provide acceptable insurance coverage for users. To protect an insurer against the possibility of experiencing catastrophic losses, a restriction is placed on the number of users who can use insurance in the proposed approach. Using this method, the maximum total loss that the insurer must pay to users falls below a sustainable amount. The number of users who can take advantage of insurance depends on the capital structure of the insurer; for a large insurance company, all users may buy insurance coverage. In this paper, a pairing process is proposed for the selection of this subset the users. This pairing process never selects user A when there is another unselected user who is more risk-averse than user A and is also interested in the insurance offer. Since the pairing process selects the most risk-averse users, it maximizes the average utility of users according to Proposition 1. This paper also demonstrates that risk aversion can be used as a useful criterion for assignment of priority to users.

In addition to risk aversion, there are several other concepts from the field of behavioral economics and psychology of decision making which can be used in service management applications. For example, the combination of loss aversion and a short evaluation period, which is referred to as myopic loss aversion (Benartzi and Thaler 1995), can be used to define and specify the appropriate length for SLAs. When users are loss-averse, they will be more willing to accept risky conditions if they evaluate their performance infrequently. Therefore, given risky conditions and users with high loss aversion, it appears that offering short-length SLAs will not be profitable for a service provider in the long term. However, more research into such concepts and psychological characteristics is needed to make them useful for service management applications. Another issue in the provision of insurance coverage is the relationship between a service provider and an insurer (insurance provider). A service provider can provide insurance coverage itself or through a third party insurer. In the latter case, the problem of trust between the insurer and the service provider must be considered. For example, the insurer must ensure that SLA violations are not intentional. Since the problem of trust does not fall within the scope of this paper, it is not discussed here; however, it is an important problem in real applications. Numerous prior works in the literature exist (Atif 2002; Siyal and Barkat 2002; Xiong and Liu 2002) regarding building trust within different systems, and these can be used to inspire the building of trust between a service provider and an insurer. The results obtained in numerical experiment illustrate the usefulness of the proposed approach for providing insurance coverage in improving the average utility of users. A game-theoretic analysis is also provided to verify the acceptability of the approach using rational users and insurers.

References

Aceto G, Botta A, de Donato W, Pescapè A (2013) Cloud monitoring: a survey. Comput Netw 57:2093–2115. doi:10.1016/j.comnet.2013.04.001

Allon G, Federgruen A (2009) Competition in service industries with segmented markets. Manag Sci 55:619–634. doi:10.2139/ssrn.907322

Atif Y (2002) Building trust in e-commerce. IEEE Internet Comput 6:18–24. doi:10.1109/4236.978365

Barto, Anandan P (1985) Pattern-recognizing stochastic learning automata. IEEE Trans Syst Man Cybern SMC 15:360–375. doi:10.1109/tsmc.1985.6313371

Baset S (2012) Cloud SLAs: present and future. ACM SIGOPS Oper Syst Rev 46:57. doi:10.1145/2331576.2331586

Benartzi S, Thaler R (1995) Myopic loss aversion and the equity premium puzzle. Q J Econ 110:73–92. doi:10.2307/2118511

Bhattacharya A, Choudhury S (2015) Service insurance: a new approach in cloud brokerage. In: Chaki R, Saeed K, Choudhury S, Chaki N (eds) Applied computation and security systems. Advances in intelligent systems and computing, vol 305. Springer, New Delhi

Bowen J, Chen S (2001) The relationship between customer loyalty and customer satisfaction. Int J Contemp Hospitality Manag 13:213–217. doi:10.1108/09596110110395893

Catteddu D (2010) Cloud computing: benefits, risks and recommendations for information security. Web Appl Secur. Springer, Heidelberg

Dong W, Shah H, Wong F (1996) A rational approach to pricing of catastrophe insurance. J Risk Uncertain 12:201–218. doi:10.1007/bf00055794

Einhorn HJ, Hogarth RM (1988) Decision making under ambiguity: a note. In: Munier BR (ed) Risk, decision and rationality. Theory and decision library (Series B: Mathematical and Statistical Methods), vol 9. Springer, Dordrecht

Emeakaroha V, Netto M, Calheiros R et al (2012) Towards autonomic detection of SLA violations in cloud infrastructures. Future Gener Comput Syst 28:1017–1029. doi:10.1016/j.future.2011.08.018

Garg S, Toosi A, Gopalaiyengar S, Buyya R (2014) SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. J Netw Comput Appl 45:108–120. doi:10.1016/j.jnca.2014.07.030

Grönroos C (2007) Service management and marketing: customer management in service competition. Wiley, Hoboken

Hani A, Paputungan I, Hassan M (2015) Renegotiation in service level agreement management for a cloud-based system. CSUR 47:1–21. doi:10.1145/2716319

Hogarth R, Kunreuther H (1992) Pricing insurance and warranties: ambiguity and correlated risks. Geneva Pap Risk Insur Theory 17:35–60. doi:10.1007/bf00941956

Javadi B, Abawajy J, Buyya R (2012) Failure-aware resource provisioning for hybrid cloud infrastructure. J Parallel Distrib Comput 72:1318–1331. doi:10.1016/j.jpdc.2012.06.012

Linlin W, Buyya R (2012) Service level agreement (SLA) in utility computing systems. In: Performance and dependability in service computing: concepts, techniques and research directions. IGI Global, USA

Luo M, Zhang L, Lei F (2010) An insurance model for guaranteeing service assurance, integrity and QOS in cloud computing. In: IEEE international conference on web services (ICWS). IEEE, Florida, USA, pp 584–591

McDougall G, Levesque T (2000) Customer satisfaction with services: putting perceived value into the equation. J Serv Mark 14:392–410. doi:10.1108/08876040010340937

Naldi M (2014) Balancing leasing and insurance costs to achieve total risk coverage in cloud storage multi-homing. In: Altmann J, Vanmechelen K, Rana O (eds) Economics of grids, clouds, systems, and services. GECON 2014. Lecture notes in computer science, vol 8914. Springer, Cham

Narendra K, Thathachar M (2012) Learning automata: an introduction. Courier Corporation, USA

Nisan N, Schapira M, Valiant G, Zohar A (2011) Best-response mechanisms. In: The second symposium on innovations in computer science (ICS), Beijing, China, pp 155–165

Oommen B, Hashem M (2010) Modeling a student's behavior in a tutorial-like system using learning automata. IEEE Trans Syst Man Cybern B 40:481–492. doi:10.1109/tsmcb.2009.2027220

Serrano D, Bouchenak S, Kouki Y et al (2013) Towards QOS-oriented SLA guarantees for online cloud services. In: 13th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid). IEEE, Delft, Netherlands, pp 50–57

Shao J, Wei H, Wang Q, Mei H (2010) A runtime model based monitoring approach for cloud. In: IEEE 3rd international conference on cloud computing. IEEE, Florida, USA, pp 313–320

Simon H (1982) Models of bounded rationality. MIT Press, Cambridge

Siyal M, Barkat B (2002) A novel trust service provider for internet based commerce applications. Internet Res 12:55–65. doi:10.1108/10662240210415826

Sureshchandar G, Rajendran C, Anantharaman R (2002) The relationship between service quality and customer satisfaction—a factor specific approach. J Serv Mark 16:363–379. doi:10.1108/08876040210433248

Wu L, Kumar Garg S, Buyya R (2012) SLA-based admission control for a software-as-a-service provider in cloud computing environments. J Comput Syst Sci 78:1280–1299. doi:10.1016/j.jcss.2011.12.014

Xiong L, Liu L (2002) Building trust in decentralized peer-to-peer electronic communities. In: Fifth international conference on electronic commerce research (ICECR-5), Montreal, Canada

Zweifel P, Eisen R (2012) Insurance economics. Springer, Berlin