Utility functions, prices, and negotiation

John Wilkes
HP Laboratories
HPL-2008-81

Keyword(s):
utility theory, price functions, risk, negotiation, penalties, SLAs

Abstract:
This paper provides an introduction to the use of utility theory with service level agreements between a computer-based service provider and a client. It argues that a consistent approach to utility, together with a flexible notion of pricing, can go a long way to clarifying some of the hidden assumptions that pervade many existing contracts and decisions around them. The goal is to enhance understanding of a surprisingly tricky area, identify a few consequences for services providers and their clients, suggest a set of terminology that reduces ambiguities, and make some suggestions for future work.

External Posting Date: July 6, 2008 [Fulltext] Approved for External Publication

Internal Posting Date: July 6, 2008 [Fulltext]

A version of this paper will be published in Market Oriented Grid and Utility Computing, edited by Rajkumar Buyya and Kris Bobendorfer, published by John Wiley & Sons, Inc

Utility functions, prices, and negotiation

John Wilkes

HP Laboratories
john.wilkes@hp.com

1 Introduction

This paper provides an introduction to the use of utility theory with service level agreements between a computer-based service provider and a client. It argues that a consistent approach to utility, together with a flexible notion of pricing, can go a long way to clarifying some of the hidden assumptions that pervade many existing contracts and decisions around them. The goal is to enhance understanding of a surprisingly tricky area, identify a few consequences for services providers and their clients, suggest a set of terminology that reduces ambiguities, and make some suggestions for future work.

Keywords: utility theory, price functions, risk, negotiation, penalties, SLAs

The environment assumed here contains a set of *service providers* that offer computer-based *services* to their clients, which may themselves be service providers. Each service provider is assumed to be an independent entity, motivated by business concerns such as achieving profitability, and so services must be paid for somehow – e.g., pay per use, subscription, advertising, or a subsidy from a sponsor. Service providers are assumed to be at least partially self-managing, and thus able to operate autonomously without human intervention, although a human is always going to be held responsible for their actions. The purpose of this paper is to discuss some of the considerations involved when a client and a service provider determine how to agree on a price for a contract between them. To keep things simple, it largely restricts itself to the interactions between one client and one service provider, although the larger context is assumed as part of the background.

This paper is meant as a tutorial, rather than a survey, and while there is a great deal of work on using economic mechanisms to control computer systems of various kinds that could be included in such a survey, space precludes discussing most of it here. Nonetheless, a few notable, relevant examples include the following: [30] for one of the earliest uses of economic mechanisms to limit access to a shared resource, Mariposa [29] for a distributed database that used profit to motivate data placement and query execution choices, and Spawn [36] and Tycoon [16] for distributed, spot-auction-based pricing of compute resources. A paper aimed at economists [12] argued for bringing economic mechanisms to bear on distributed control of computer systems, and pointed out some associated perils, such as potential inefficiencies compared to centralized schemes, and oscillations from delayed access to partial information.

The field of self-managing systems (called autonomics by some) has been around for a long time. Such systems need clear objectives to achieve, and these have often been specified as "utility" or even more explicitly "profit". A few examples from a crowded field include [2], [13], [14], [24], and [37]. The Trading Agent Competition (http://www.sics.se/tac/) pits implementations of autonomous agents against one another

Note: a version of this will appear as Chapter 4 in: *Market Oriented Grid and Utility Computing*, edited by Rajkumar Buyya and Kris Bubendorfer, and published by John Wiley & Sons, Inc., 2008.

in a series of competitive games that foster learning about how such systems should operate.

References to other supporting material are interleaved into the rest of the paper's text.

The remainder of the paper begins with a review of service level agreements and prices, before introducing the notion of utility and how that affects prices and negotiation. This is followed by a discussion of how risks and penalties affect prices and contracts. The paper concludes with suggested directions for future work, some overall observations, and a summary.

2 Service Level Agreements (SLAs)

A service provider offers service to a client under the terms of a service level agreement (SLA), which is a bi-lateral contract that governs the terms of the interaction between the parties. An SLA is typically negotiated before service will be provided, although some services may be offered under an implicit SLA, which the client is deemed to have agreed to if it uses the service.

For self-managing computer entities, the SLA will take the form of a computer-readable document. It will typically contain some or all of the items listed below (this list is loosely extended from [38]):

1. The identities of the provider and the client.
2. The start and duration of the contract.
3. The service that is to be delivered. This is typically described in an informal manner, although interface specifications may be incorporated directly or by reference (e.g., if they are written in WSDL [8]). Reporting, escalation and remediation processes belong here, although they are often bundled into "what happens if things go wrong". See [21] for a survey of many aspects of service specification.
4. Service level indicators (SLIs), which are carefully-defined, observable, measures on the service or the behavior of the client or provider. Examples might include "the 95th percentile of the per-second message-arrival rate accumulated over a 5 minute interval", or "the number of discrete 1 second or longer intervals in the last minute in which another (named) SLI had a value of less than 50, when sampled every 100ms". Sometimes processes for taking the measurements are also defined in an SLA, and may include the use of a particular trusted third party to make the measurements [9].
5. The amount or level of service that is expected, often expressed in the form of service level objectives, or SLOs. These represent bounds on both the client's and service provider's behaviors (e.g., highest load level permitted, target response time, maximum delay before a technician is dispatched if an outage occurs). The term "SLO violation" means that a desired SLO is not being met during the execution of a contract. SLOs are defined in terms of SLIs.[1]
6. The price: how much money will change hands as a result of this SLA being executed. By convention, positive values represent payments from a client to a

[1] People often mistakenly talk about "an SLA being violated" when all they mean is that the expected SLOs aren't being met. If the SLA itself is being violated, the usual recourse in the USA is to the law courts.

service provider; negative values the reverse. Thus, penalties imposed on a service-provider are specified as negative values, and added to the price.

7. When payment should occur, such as when or if invoices will be issued (e.g., whichever comes first: SLA completion, monthly, or once the debt exceeds $1000), and how soon payment is expected (e.g., before the service is used, or within 1 minute of an invoice).

8. What happens if things go wrong? For example, penalties may be specified that will be incurred if a party's behavior fails to conform to what is expected. The most common is if the service provider fails to meet an SLO level, but a good SLA also constrains client behavior. Such penalties are often financial, and can range from a partial rebate of service fees, through recompense for economic damage, to punitive. There may also be implicit penalties, such as damage to a supplier's reputation.[2]

9. Additional conditions (e.g., the jurisdiction in which the agreement is to be interpreted; the use of binding arbitration to adjudicate disputes, confidentiality terms that apply to the SLA itself).

10. Irrefutable evidence that both parties have agreed to the terms of this particular SLA, such as a pair of digital signatures of the SLA's contents.

SLAs are often simplified by relying on "out of band" terms and conditions that must be interpreted by humans, leaving the SLA to describe only those parts that both sides feel might be mis-interpreted by the other. This has some risks, however, since omitting something from an SLA is tantamount to giving the other party permission to behave in any way it pleases for that part of the agreement.

It helps to assume that the two parties are selfish, i.e., interested in protecting their own interests, mildly distrusting of one another, as well as what economists call rational – taking a dispassionate view of gains and losses. This will motivate them to specify everything in an SLA that they feel the other party might adopt a contrary position on.

Specificity carries costs (e.g., generating, understanding, and checking clauses), so great thoroughness tends to be employed in an SLA only when the stakes get large. This is just like traditional commerce, where a simple one-off purchase may be done without a contract, but a multi-billion-dollar commercial deal is typically governed by a contract that runs to hundreds of pages, much of which is concerned with handling contingencies and other risk mitigation measures. Such terms may explicitly be included in the SLA itself, or incorporated by reference, or simply assumed. In practice, it is best to think of an SLA as a legally-binding contract. If something matters, include it explicitly, or by reference.[3]

---

[2] Although these implicit penalties are not usually recorded directly in an SLA, they may be hinted at by attempts to limit them, such as restrictions on the types of recourse a party is allowed to pursue such as binding arbitration and confidentiality clauses.

[3] This often causes surprise, provoking observations of the form "computers cannot generally understand natural language text". This is immaterial: legal contracts are always interpreted in a larger context, which often includes niceties that are basically incomprehensible to a lay person. What matters is that the details of interest to the parties are accurately, and completely, spelled out in the SLA, and that the actionable elements of the contract are represented in a way that the parties can interpret, preferably automatically.

3 Price functions

Most SLAs are written as if the SLOs are the governing terms. Instead, this paper takes the position that because SLAs are business contracts, what actually matters is the financial consequences to client and service provider of the outcomes that are achieved for an SLA.[4] Those financial consequences are defined by the *price* of each outcome – how much money changes hands between client and service provider for that outcome.

Prices are the governing terms in a contract, not SLOs: an SLO is merely a hint that specifies a desired outcome. How desirable? That's determined by the price: each possible outcome has a single price. This leads naturally to calling this approach *outcome-based pricing*. The range of prices that cover all possible outcomes is represented by means of a *price function*, which maps the space of outcomes to prices.

Since a price function reflects the interests of the service provider and its client across the entire range of potential outcomes, it is typically more complicated than a constant value to represent a simple flat fee. For example, it might include volume-related usage fees, discounts for use of multiple service features at one time (a form of product bundling), a monthly subscription charge, penalties for misbehavior – or all of these.

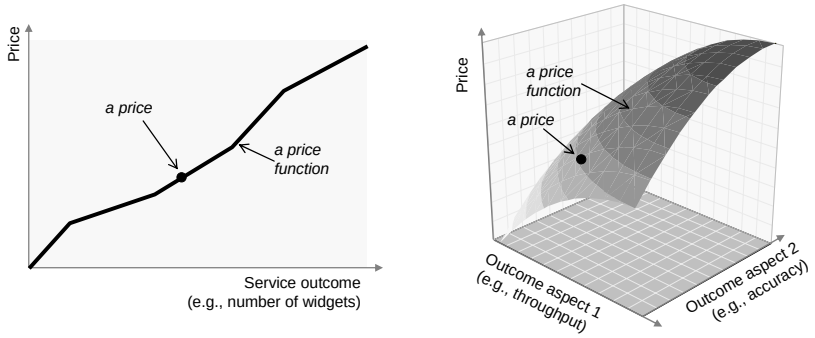

Figure 1: a price and a price function for (a) a single outcome dimension, for which the price function is a line, and (b) two outcome dimensions: with N outcome-aspects, it's a surface in an N+1-dimensional space.

The price function can be proposed by the service provider or by the client – as long as they both agree in the end, it doesn't really matter. In practice, many service providers are reluctant to negotiate a pricing structure with each client, so price-setting service providers are more common than clients, especially for services with many clients, although clients may be allowed to specify values for parts of the pricing function, such as penalties. Since client and service provider distrust one another, outcomes, and hence price functions, must be defined over measures that both parties have agreed to and are mutually visible – i.e., SLI values.

A price function is represented in an SLA document by means of a *price function model*, or simply *price model*, which allows the price function to be communicated between the service provider and the client. Limitations in the capabilities of the price model may in turn limit the set of price functions that can be supported – for example, it is common to simplify the price model to make it easier to understand or write, such as by formulating it as a sum of independent terms (e.g., a fee for the base service plus any penalties). The

[4] This paper uses the word "outcome" to mean everything but the price. Although a case could be made for including the price in the outcome, excluding it simplifies some of the descriptions.

complexity of the price function can also be reduced by limiting the range of alternatives that an SLA can describe, e.g., by imposing constraints on what can go into an SLA.

Outcome-based pricing has several desirable characteristics, especially in the context of self-managing service providers. It eliminates ambiguities about how to make tradeoffs between different alternatives – what if two SLOs are being violated, but only one can be fixed? It provides a numerical scale (a common currency) that can be used to rank-order [the consequences of] different service provider and client behaviors. And it provides a clear input to the objective functions, such as "maximize profitability", that are used when a service provider is determining how to respond to different situations.

3.1 What-if prices

One benefit of embodying a price function in a price function model that is embedded in a proposed SLA is that the model can then be explored in private by a client, before a contract is accepted, to investigate what will happen under different outcomes. I sometimes call a price function a "what-if price" for this reason.

A flexible way to structure the price function model is to represent it in an executable form, such as an interpretable language, so that it can be executed to provide a price in response to a set of inputs. Those inputs are potential values of the SLIs, together with flags for events such as "the service provider unilaterally cancelled the SLA". In fact, such flags are best thought of as additional, perhaps implicit, SLIs. Definitions of the inputs to the price function are a necessary part of the price function model.

How best to trade off the expressive power of a price function against the ability of the recipient to reason about the result is still an open question. If a client cannot understand the behavior of a price function, it should probably not agree to an SLA that contains it. A common, simple pattern is to describe the price function as a sum of independent parts. At the opposite extreme are the complex functions that are needed to describe rich price structures with interdependent parts – such as personal taxes in industrialized nations, with their subsidies, discounts, caps, rebates, and other special cases that reflect a government's policies about wealth (re)distribution.

Executable specifications offer great flexibility, but risk imposing equally great complexity in reverse engineering their behavior. Perhaps this is the real value of SLOs: they can provide hints about boundaries in the outcome space where the price function model should be expected to emit relatively rapidly-changing values.

3.2 Setting prices

There are many ways of determining how to set the price for a service. This paper approaches the problem from the perspective of a rational, self-interested service provider, which is interested solely in maximizing its profitability in direct negotiations with a single client. For simplicity, it largely elides details of longer-term issues such as market segmentation, building a customer-base, competition, and customer retention, all of which might cause a service provider to offer different prices at different times to different clients, and focuses on a single interaction between a client and a service provider.

There are a great many different ways in which services can be priced; [21] provides a survey, and section 5 touches on the topic of price negotiation. This paper will simply assume that the service provider has already chosen the structure of its *pricing function*, which determines what inputs the price function will use for a particular service. For example, the service provider might charge on a per-unit basis, plus a flat fee or a

monthly retainer; it might offer a subscription service up to some maximum amount of service; it might offer bulk discounts for large purchases or offset some charges against others; it might charge its most loyal customers less (or more); it might allow its clients to select from several different forms of pricing function; and so on. Although the structure(s) may be fixed, there's typically still room to negotiate the parameters and constants used in the pricing function – e.g., the actual rates charged, the break-points that correspond to desired service levels (SLO targets), and potential penalties to be paid.

Ultimately, whether a service provider is able to impose its choice on customers depends on the latter's willingness to pay its prices – which in turn depends on the number of potential customers and their valuations of its services, as well as the presence of competitors and their behavior. The amount of demand for a service is generally assumed to be a function of the price at which it is offered, with lower prices bringing greater demand – the so called demand curve of microeconomics, also known as the elasticity of demand with price. There is a great deal of literature in this space; [18] provides an approachable introduction to the topic.

Estimating a demand curve is difficult – doubly so in the relatively new market for computer-based services. The *price at risk* methodology [23] attempts to control this uncertainty by using estimates of both demand curves and uncertainty in that demand to produce distributions of likely profitability, using models such as Monte Carlo simulations. This approach allows a service provider to make statements of the form "I will accept a contract if the expected return (e.g., gross profit margin) is greater than $X and the likelihood of a loss is less than Y%".[5] More on this below.

4 Utility

Like many before me, I find the notion of utility a useful one to guide decisions in automated service providers. This section discusses what is meant by utility.

In microeconomics, *agents* (i.e., service providers and their customers) are assumed to have *preferences*, which represent (partial) orderings on outcomes, and those preferences are assumed to guide behaviors: rational agents will always attempt to achieve the outcomes they most prefer. In many cases, a party's preferences can be mapped to values of *utility*, where higher utility means greater preference. Utility is measured in subjective units, and so it does not make sense to compare or add two parties' utility values, or even to try to normalize them into a common range. A *utility function* is simply a mapping from a space of outcomes onto utility values.

[5] Price at risk is related to *value at risk*, which is used to articulate the amount of risk associated with a trading portfolio. A value-at-risk estimate is a prediction, at a certain confidence level, of the maximum amount of money that might be lost given a model of likely future price movements in a set of financial instruments. Known difficulties include estimating the likelihood of rare events using historical data, and determining the correct models to use [31].

Figure 2: utility functions for a range of prices for a fixed outcome: (a) two simple mappings between price and utility; (b) the author's *de facto* internal utility function when he last negotiated the price for a car.

In the simplest possible case (see Figure 2(a)), the service outcome is fixed and the utility is just a function of the price to be paid – it might be as trivial as a linear function of the profit an agent would expect to make, but many other options are available, especially when people are involved, as Figure 2(b) suggests.[6]

For computer services, it is normal to have at least one variable outcome such as response time or throughput, in addition to the price for the service. Indeed, there are often multiple dimensions (or aspects or attributes) to the service outcomes – e.g., throughput and response time and accuracy for a web service, or memory and bandwidth and startup delay for a virtual machine rental service. The extension is straightforward: the utility function is defined over all these aspects plus the price, forming a surface in an N+2 dimensional space for N outcome aspects plus a price.[7]

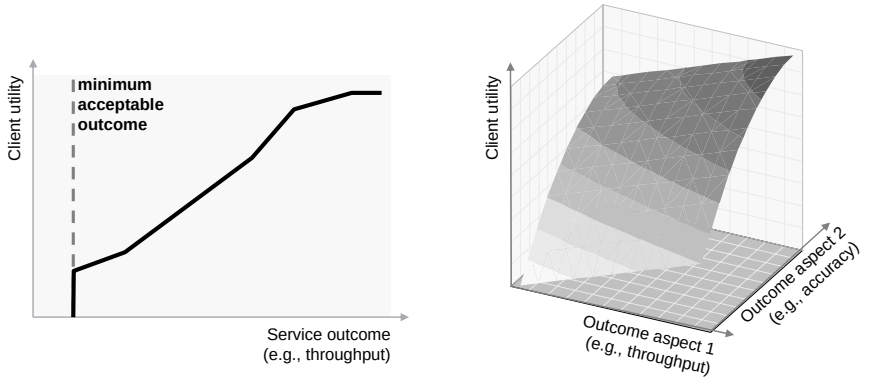

Figure 3: utility functions for (a) a single-dimension outcome and (b) multiple outcome aspects.

[6] In the USA, "list price" is specified by the car manufacturer, and "invoice price" is what the dealer pays the manufacturer – before any adjustments for volume discounts or other sales incentives. Note that utility dropped as the price got too low because of the implied risk of faulty or counterfeit goods or provider mis-behavior in other areas, such as ability to deliver.

[7] The assumption made here is that neither party discovers a significant loophole that allows them to achieve the outcome they desire for a much lower (higher) price, such as by violating an SLO without getting caught.

In what follows, I will use an example with only one kind of outcome to make things easier to visualize. I further simplify the discussion by assuming that contracts are independent, and the parties retain no memory of previous negotiations.

Soliciting complete multi-dimensional utility functions from people is hard, so simpler approaches are often used. For example, multi-attribute utility theory (MAUT) [15] typically assumes independence of each outcome aspect or attribute, and generates N+1 independent utility functions – one for each outcome aspect, plus one for price. These are weighted and then added together, although other ways of combining the individual values are possible. Obviously, utility functions formulated in this manner are less general than the full form – for example, they cannot readily express a utility function that requires minimal values of several aspects to be satisfied simultaneously – but they may be useful in some circumstances, especially when people are involved.[8] Since the focus of this paper is automated clients and service providers, I prefer the full formulation for its flexibility and greater capability.

It is common to constrain or approximate the utility function to make it more mathematically tractable, by insisting on features such as smoothness, continuity and differentiability.

4.1 The client perspective

Consider the example of a request-processing service, such as a web service or a job-execution service, with some appropriate, agreed-upon measure of throughput, such as the count of requests completed in some averaging interval. Suppose there is only one price to be paid at the end of the contract, and this price is determined by a single outcome value measured along one dimension – the request throughput. A client of the service is likely to prefer higher throughputs over lower ones, although there may come a point of diminishing returns on the high end, and there may be a lower bound below which the service becomes unusable (see Figure 4).

[8] A common form of MAUT in human-to-human interactions is to have each participant write down their per-aspect utility functions in the form of a table, provide weights for summing them, and normalize the result (e.g., into a [0,100] range). The functions and weights are then communicated to a trusted third party mediator, who can use this data to calculate an "optimal" operating point that maximizes the sum of the overall utility functions for both sides. Although this is makes good sense when it results in an agreement where none was possible before, it relies on many assumptions that I prefer to avoid.

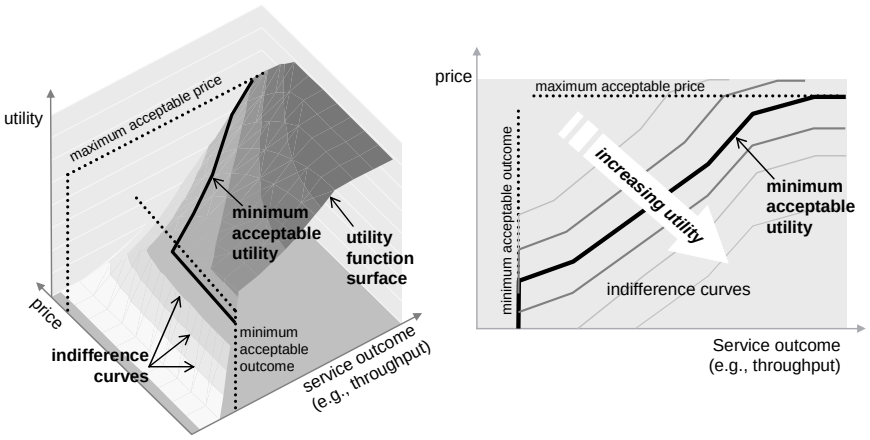

Figure 4: a sample utility function: (a) utility as a function of service throughput and price, with indifference curves on the surface representing the function; (b) contour plot of the same utility function projected onto the price versus throughput plane.

When an SLA is negotiated, the client's objective is to persuade the service provider to provide it with the best possible utility – in this case, by maximizing the throughput offered. The client's utility can be expressed as a function of throughput, $utility_{client}(throughput)$.

But there's no free lunch. It is likely that the service provider will expend more resources to provide a higher throughput, and to maintain profitability it will want to recoup its extra costs by raising the price for the service. Now the client needs to rank-order its preferences over the 2-dimensional space of response time and price, which can be accomplished with a utility function $utility_{client}(throughput, price)$.

The client is said to be *indifferent* to (i.e., equally happy with) all outcomes that produce the same value of utility, and the line connecting such a set of points is called an *indifference curve*. The indifference curves might look like those shown in Figure 4.

A client will only accept contracts that [are likely to] provide at least some minimum utility – the value at which they are indifferent as to whether to accept a contract or not. Since utility is measured in arbitrary units, and can be rescaled and renormalized at will, it is sometimes convenient to set this minimally-acceptable utility value at zero; positive values imply the client will be happy with the outcome; negative values mean they will be unhappy.

When mapped onto the price-outcome plane of Figure 4(b), that minimum utility represents a worst-case price curve for the client – it is the *client max price function*, also called the client *price boundary function* and the client *reservation price function*. It is used to define the client's acceptable region during contract negotiation: no contract that lies in a region with lower utility (e.g., a higher price or lower output), will be acceptable. In Figure 4(b), only the area to the right and below the client's max price function will be acceptable. It is also common – as shown here – for there to be an absolute upper bound on the amount the client is willing to pay, regardless of the amount of service obtained.

For multi-outcome utility functions, the same trick can be applied to generate a client max price function from a minimum-acceptable utility indifference-curve. With N aspects, the client indifference curves now traverse an N+2-dimensional surface, and they can be projected down onto an N+1 dimensional space.

4.2 The service provider perspective

A service provider performs a calculation similar to the client's, to calculate its utility for the range of possible outcomes and prices. When these are mapped into the price/outcome space, the result is a *minimum* acceptable price (*min price*), below which it will not enter into a contract.

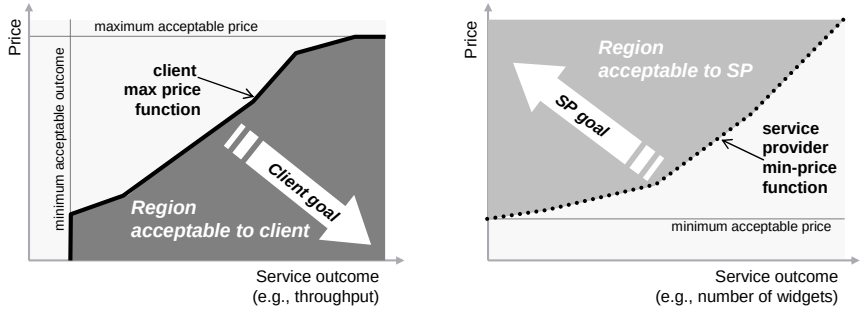

Figure 5: the client and service provider (SP) perspectives: each has a viable operating region bounded by a *max_price/min_price* function.

A tenet of outcome-based pricing is that price functions in SLAs must be written only in terms of outcomes that are visible to both parties: it is not acceptable for a price function to depend on "magic values" that the service provider can set to any number they like.

One effect is that although "cost plus" price functions are quite common in human-mediated contracts, they are less likely to be useful in SLAs agreed between mildly distrusting automated agents: they require the service provider's costs to be made visible and (potentially) audited to check that they are being reported correctly, which is itself expensive. Instead, service provider costs are usually folded into a price function by means of flat fees and service-consumption-related factors of one kind or another.

A second effect is that the service provider's utility function is likely to depend upon hidden "outcome" data that only the provider is privy to, and cannot directly be included in an SLA. For example, the service provider will probably estimate its costs for a particular client-visible outcome using a model of client behavior, service provider outlays, and desired service level. Service provider outlays typically fall into two categories: (a) direct costs associated with a particular SLA, such as those for renting resources, license fees, and power and cooling bills, and (b) indirect costs, which include fixed overheads such as resource purchases that must be amortized across many contracts, overheads such as personnel, and additional business expenses such as taxes. A service provider may choose to include other factors in its utility calculation, such as the opportunity cost of tying up a resource (using it now may preclude a lucrative future use of it), and it may demand an adequate rate of return on its investments, not just a non-zero profit. The result will be a private utility function over the service provider's outcome space. Given a minimal acceptable utility value and its associated indifference curve, the service provider can then determine a minimum acceptable price.

Working out how to do this mapping between its hidden utility function and the client-visible price function is one of the hardest parts of determining how to set the price for a service. By using a prediction rather than *post facto* measurements, the service provider is taking on some risk; for example, it may not be able to determine in advance how many resources will be necessary to achieve a particular service level.

It is sometimes convenient to construct functions to convert a utility value calculated over the non-price outcomes into a client max price and a service provider min price: $max\_price_{client}(utility)$ and $min\_price_{provider}(utility)$. The inverse functions are known as

the utility of money, or of consumption: $utility_{client}(price)$ and $utility_{provider}(price)$. For a rational party, this function will be strictly monotonic. One of the commonest formulations is to assert that these functions are 1:1 mappings, which is occasionally misunderstood to imply that utility is measured in monetary units.

5 Negotiation

The purpose of SLA negotiation is to arrive at a mutually acceptable contract for both parties, which will contain an *agreed-upon price function*. This paper focuses on bilateral negotiation between a single client and a single service provider. Additional forms of negotiation are certainly possible – auctions are quite popular – but they typically require that clients trust the provider to act as a trustworthy broker, market-maker, or auctioneer, or require a trusted third party to fill this role.

An underlying assumption is that the agents are self-managing, autonomous computer services that are rational in the economic sense. This distinguishes much of the discussion that follows from research into people-to-people negotiation, where additional aspects such as respect for the other party and relative position power can complicate the issue [17], [25].

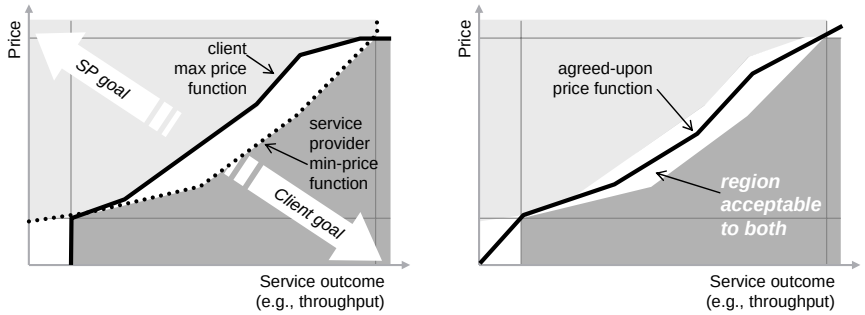

Figure 6: the goal of negotiation: (a) maximizing each side's excess utility; (b) a possible acceptable outcome, expressed as a price function that traverses the mutually-acceptable region.

During a negotiation, each party will try to maximize their excess utility – the additional utility they receive above their minimally-acceptable indifference value. To do this, both parties try to drag the final agreed price function closer to their own preferred operating regime. For example, in Figure 6, the client should try to pull the price function down and to the right, while the service provider should try to pull it up and to the left.

There is a great deal of existing work on negotiation strategies, tactics and protocols, and this paper will not attempt to discuss it in detail; see [3] and [10] for some representative examples.

Reaching agreement requires guessing – or determining (e.g., by probing) – an approximation to the other side's boundary function. In general, the goal of a negotiation strategy is to extract as much information as possible about the shape of the other's utility function for the minimum concession in the proposer's utility. [11] describes one way of making tradeoffs during this process, under an assumption of mutual benefit, which may or may not apply. Negotiation strategies, such as proposing smaller concessions in later rounds, are ways to communicate (possibly fallacious) hints about the shape of a party's utility function, and how close they are to reaching agreement, in an attempt to influence the other party's decision-making. The maximum movement towards a desirable outcome for one party will occur in area where the slope of the utility function for the other party is lowest, and one purpose of negotiation strategies is to find – and exploit –

such regions. This is easier if the parties engage in multiple interactions and past history can be used as a guide to future behavior [39].

Mutual exploration of the utility functions of distrusting partners is not always very efficient: it takes time, can result in sub-optimal answers, and offers no guarantee of "fairness" or even success. Such is the nature of negotiation. Note that the common expedient of summing the two parties' utility values and solving a differential equation to determine the maximum common utility is not usually possible, because (a) utility values are subjective, and so not directly comparable, and (b) neither side is willing to furnish the other with their utility function in this kind of direct negotiation. A party that volunteers its utility function or price-boundary function to the other will "lose" the negotiation, by giving up any ability to achieve a better outcome for itself.[9]

Some systems aim to distribute excess utility more or less "fairly" between the two parties. For example, K-pricing offers one approach [4], by splitting the excess price (not utility) beyond each party's price boundary functions in a fixed ratio between them. It requires a third party to perform the calculation if the client and service provider do not fully trust one another, and hence falls outside this two-party scenario.

Purely rational agents do not need to achieve fairness in order to reach agreement, but fairness is an important property in human-to-human negotiations. People will refuse to enter into an agreement if they view it as sufficiently unfair, even if they would benefit (the ultimatum game [22]). This suggests that future automated agents should offer the option of being programmed to reflect this, on the grounds that their human masters may otherwise come to regret the consequences.

A negotiation process or mechanism is said to be *incentive compatible* if it is in the interests of the parties to reveal private information that the process requests. This property is not strictly necessary for reaching agreement, although it may speed negotiations and increase the likelihood of reaching it.

A service provider might not want to offer a service that is defined at all possible outcomes, but it needs to provide a price function that is valid for all of them. For example, a virtual machine rental service may choose only to offer virtual machines at certain fixed capacity points, even though the underlying virtual machine monitor might be capable of providing a near continuum of offerings. Since the price function deals with outcomes, not desired levels of service, it needs to specify what happens if (say) more or less processor power is made available than was expected. There are many ways it can do so – for this example, it might simply round up the price to that of the next-larger service delivery unit.

A client may want to bias the service provider to offer outcomes that the client would prefer, to speed up the negotiation process or increase the likelihood that a desirable outcome is reached. At the same time, the client must avoid giving away its max-price function. One way to do this is to provide a hint to the service provider about which outcomes the client prefers. A *client ranking function* provides such a hint by ordering some or all of the outcomes in client-desirability order [27]. A bigger ranking value should imply the client's willingness to pay a higher price – but no data about how much higher a price. This can be accomplished by a non-linear mapping from ranking value to the client max-price value for the same outcome.

Sometimes a "good enough" agreement is better than none at all, or better than spending too long trying to get a marginally better one. That is, there is a utility aspect to the

[9] Imagine going into a used car dealership and announcing the highest price you are willing to pay before starting haggling over the price …

negotiation itself. This is captured in the notion of *satisficing*, which can either mean (1) picking an agreement point that is at least minimally acceptable (typically the first), on the grounds that it is likely to be close enough to the optimal; or (2) including the cost of the negotiation process itself in the decision-making about when to end the negotiation process. It may even be the case that the cost of reaching an agreement is greater than the potential value of the result to a party, in which case it isn't worth starting a negotiation.

Fortunately, many situations are relatively straightforward: many service providers will be price-setting, meaning that they will propose a price in response to a request for a particular level of service. Such a price is likely to reflect a previously-published skeleton price function that allows clients to estimate what it is reasonable for them to ask for. Typically, a service provider will advertise an SLA template [38] that includes many of the service terms; a client will populate it with the details of their service request and ask for a quote; the service provider will respond with a specific service offering and a price function; and the client will either accept the result, or modify the request and repeat the process until they are satisfied or abandon the attempt to reach an agreement. Many variations of this basic protocol are possible. For example, the client may be the active party, proposing a pricing function in the SLA they propose; or both parties may make arbitrary modifications to the SLAs they propose, which may speed up the process of reaching agreement by communicating more information in each round.

5.1 Expected utility and risk

Never [try to] cross a river because it is on average 4 feet deep – Nissam Taleb

An *ideal* agreed-upon price function will lie within the acceptable region (boundary function) for both parties across all possible outcomes. These are rare, for several reasons. For one, SLAs often specify penalties. By definition, these lie outside a preferred operating range. For another, it is unlikely that a client would be able to persuade a service provider to offer a price function that lies within its preferred operating region for all possible outcomes. For example, the service provider might charge too much at high service levels for the client's preference. But if the client doesn't expect those service levels to be reached, that may not matter. What the parties are really trying to maximize is their *expected utility value* over the actual outcomes that they believe will be experienced under the SLA [28].

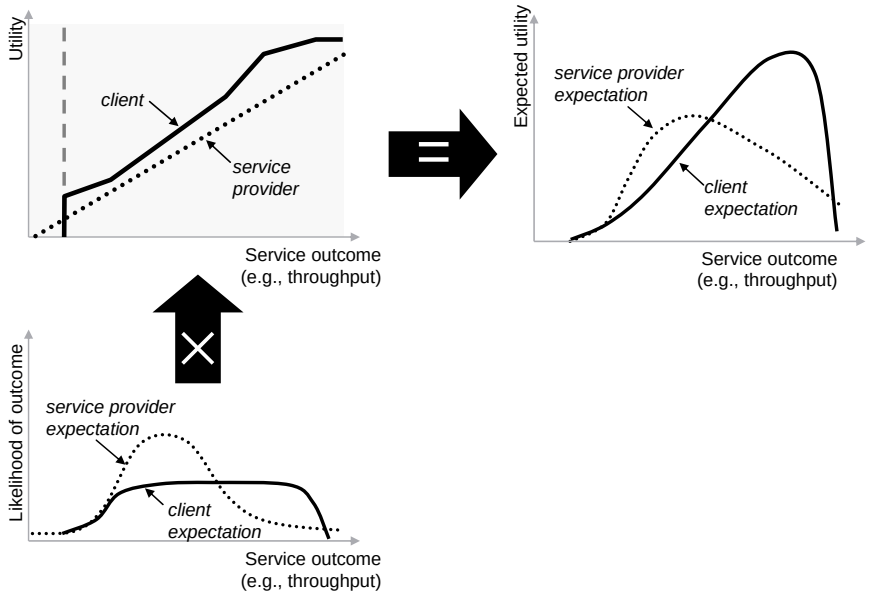

Figure 7: calculating expected utility for a predicted set of outcome likelihoods. The client and service provider lines are shown together on one graph only for convenience; they are each independently – and privately – calculated.

Conceptually, the expected utility is simple to calculate: over the space of possible outcomes, sum (or integrate) the product of each outcome's likelihood and its utility.[10] In practice, of course, things are not as simple as this suggests.

Firstly, there is likely to be intrinsic uncertainty in the outcomes – just how many cellphone minutes will I use in a month? how much load will my customers impose on the service? what will the response time be under such load? These are manifestations of *risk*, which is defined as variance in utility across outcomes. Most people associate risk with negative outcomes (i.e., losses compared to an expected outcome), but that is not its strict definition, although the downside risk is often more important.

Client downside risk can be reduced by minimizing the price for each additional unit of service. Service provider downside risk can be reduced by ensuring that the incremental costs for delivering more service are adequately covered. Both parties can reduce their risk by capping the amount of service that is to be delivered (e.g., by throttling or shedding excess load), and by improving the quality of their predictions.

Secondly, the client and service provider may have asymmetric information: one may know more about the likelihood of certain outcomes than the other. For example, a client may never have used this service before, but the service provider has been in business a long time, and had many clients. Clients can reduce the asymmetry by researching alternate information sources such as reviews, surveys, analyst reports, or reputation ratings from third parties, and they can reduce their downside risk by making conservative assumptions about the relative frequencies of undesirable outcomes. Service providers can reduce their risk by pushing for SLAs that bound the downside of

[10] The ability to perform this calculation relies on some detailed properties of the agent's preferences – the so-called von Neuman-Morgenstern conditions or axioms [35]. Most observers seem to agree that these conditions are usually reasonably well approximated in practice.

mis-estimating their real costs, and penalize undesired client behaviors in order to discourage them.

Thirdly, the formulation offered above assumes that the agents (client and service provider) are indifferent to all outcomes that achieve the same expected utility. In practice, the agent's utility value might itself be affected by the probabilities of the individual outcomes. *Risk aversion* is how this affect can be quantified: it offers a measure of how an agent values sets of possible outcomes in comparison to a fixed single outcome [1].

A classic example of risk aversion occurs with mortgage loans: a risk-averse borrower is willing to pay a fixed interest rate that is higher than the expected variable interest rate average, in order to reduce the risk associated with interest rate variation over the lifetime of a loan. Formally, economists talk about how agents react to participating in a *lottery* (a hypothetical game) that offers different outcomes – for example, a 50:50 chance of receiving either $0 or $100, or a guaranteed $40. A risk-averse agent might prefer the latter; a risk-seeking agent might prefer the lottery over a guarantee of $59. The difference between the expected value and the guaranteed one ($10 in the first case, –$9 in the second) is known as the agent's *risk premium* for this lottery: it's a measure of how risk averse they are. Risk aversion may increase or decrease with the size of the potential payout.

When people are involved, behaviors associated with risk quickly get complicated. For example, the common explanation for risk aversion [35] is that the marginal utility of wealth (money) decreases – as a payout gets larger, each additional $1 contributes less utility than the one before it (economists say that the utility of wealth function is concave). But this is insufficient to explain the difference in people's behavior between moderate- and large-scale bets [26]. Cumulative prospect theory [34] attempts to address this, by offering models of people's risk behavior. It accounts for several observed phenomena:

- people are *loss aversive* (they weight a loss as more significant than a gain of the same amount);
- they are more receptive to risk when they are below an expected reference point or target, on the grounds that "I have to try *something* to get ahead", and significantly more averse to risk when they are above the reference;
- people suffer from *long-shot bias* – they overweight very rare, extreme events, such as winning the lottery, a terrorist attack, or a nuclear meltdown, and discount "average" occurrences.

Autonomous service providers may want to take these factors into account when dealing with human clients, and these factors may be a part of the business goals that their owners and operators would like to impose on them. Risk-related behavior will cause both clients and service providers to modify the utility they associate with a potential SLA away from the straightforward expected utility value. In turn, this moves their boundary functions, and thus changes the pricing functions and SLAs that they will find acceptable.

5.2 Losses and penalties

One particular area where risk is associated with SLAs is when there are sets of outcomes (operating regimes) in which losses or financial penalties occur.[11] No *ideal* price function will include losses or penalties, since no agent is indifferent to loss. For example, the price function in Figure 6 is not ideal because it lies outside the acceptable region at the two extremes of service outcome.

It is likely that both the service provider and the client will have to compromise if they are to reach a price function that they can both live with. The degree of risk that each side is willing to accept, plus the likelihoods of the various outcomes, will determine whether agreement is possible.

A common approach to losses is to distribute them between the two parties in some fashion; for example, by using the k-pricing technique introduced earlier, but applied to price deficit, rather than excess. Pushing all the losses onto one party or another can introduce *moral hazard* – whereby one party can impose a bad operating point on the other and yet is largely insulated from the consequences. Penalties are a special case of this: they are intended to impose an undesirable outcome, so sharing losses makes no sense. The risk of moral hazard means that an agent should be wary of accepting SLAs with penalties unless it largely controls whether or not the operating point enters a regime that will trigger the penalty.

A penalty is intended to discourage the service provider from operating in a region of outcomes that is undesirable to the client. A penalty will only be effective if it is large enough to make other, reachable, operating points more profitable for the service provider. A strict form of penalty design calls for every incremental movement away from a bad operating point to result in a net benefit to the service provider [4], but this may not be necessary if the control system for the service provider can recognize better outcomes and move to them, even if they are not adjacent to the problematic one.

How should a penalty be priced in an SLA? In just the same way as with any other risk. The service provider should make sure that two additional terms are reflected in the price function somehow for each penalty: (1) compensation for the reduction in their expected utility from the expected penalty payout (the product of its likelihood and its size), plus (2) their risk premium. Formulating things in this fashion allows a client to specify the penalties they want in an SLA, as long as the service provider can determine the remainder of the price function. Typically, the higher the penalties, the larger the expected value of the price function.

It may be possible to reduce the effective risk premium by buying insurance against loss. Insurance is a way to pool downside risk across multiple entities; since the likelihood of all the downsides coming true simultaneously is low, the intrinsic cost of loss coverage plus an appropriate profit for the insurer may be less than a single entity's underlying risk premium. Thus, the price function's "risk premium" may be set from the provider's raw risk premium or the cost of insurance – whichever is lower.

To simplify the description, the discussion above focused on penalties paid out by a service-provider, but it applies just as well to a client, since an SLA may stipulate penalties for clients too – most commonly if they impose more load on a service than was agreed to, or are tardy in providing payment. The client, too, should make similar

[11] Using financial penalties everywhere simplifies many things, and is not particularly limiting. Other types of penalty, such as additional service in lieu of a payment, can usually be given an economic equivalent, such as the cost to the service provider of providing the additional service.

calculations about the expected value of the price function as well as any risk premiums it may care about.

6 Future research directions

There is a long way to go before most autonomous service providers are capable of the kinds of economic analysis presented here. Even a basic approach is beyond many implementations [7], and the use of even moderately advanced financial instruments such as futures and options is still further away. All of these require a clear understanding of the value that the mechanisms and policies are trying to achieve, and how effective they are. There seem to be many opportunities to leverage existing work in other domains and apply it to the field of self-managing systems.

More systematic management of risk in automated service providers and the SLAs they write deserves greater attention. It is not enough to simply measure the outcomes from a contract, or set of contracts: what is needed is for the risk associated with those outcomes to be used as input to the control system for the service provider, and their clients. This may not be enough: existing work tends to assume that decision-making should be rational, but this fails to incorporate models of people's attitudes towards risk, such as cumulative prospect theory. The process of eliciting and representing user preferences in these areas is remarkably difficult, by no means fully understood, and merits further work [18].

There has been much work on human-to-human negotiation, but it's still an open question whether it is a good idea to mimic these processes in automated agents.

Finally, it should be noted that utility theory is not the only form of representing inputs to decision-making under uncertainty. It is appealing for automated systems because it can readily be mapped onto numerical optimization-based approaches, but other formulations, such as target-oriented decisions analysis [33], have been shown to be helpful in getting people to think about their choices [5], and may be applicable to autonomous computer systems, too.

7 Conclusion

Individuals, businesses and other organizations have come to rely heavily upon automated computer systems, including service providers and autonomous agents. The trend is likely to continue apace, meaning that more, and larger, decisions will be placed in the hands of such systems. It is becoming increasingly important that we have a clear way to delegate our intentions to these systems, so they can act on our behalf, with some assurance that unpleasant surprises won't occur.

This paper has attempted to address one aspect of this, by discussing ways to capture the complicated mapping between happiness, service outcomes, and prices. It has provided an introduction to the topic of utility and risk in the management of SLAs for a pair of partially-distrusting computer-based systems. It has shown how utility theory provides a basis for automated decision making for contracts and their prices, including some approaches to handling different types of risk. And it has suggested a number of extensions and opportunities that would allow automated service providers and their automated clients to do a better job of serving their human masters.

The use of utility as a guiding principal for self-managing systems has long been recognized. Nonetheless, fulfilling these opportunities, and putting the ideas into practice, will be a significant challenge for some years to come.

8 Acknowledgements

Sharad Singhal provided the impetus to write this paper, and the basis of the graphical representation used here of the overlap between client and service-provider acceptable-regions. Elaine Wong implemented a prototype of the executable what-if pricing functions described in section 3.1, and, together with Subu Iyer, acted as a sounding board for several of the ideas presented here. Yang Yinping provided many helpful references and comments on the topic of negotiation. Claudio Bartolini and Kay-Yut Chen helped educate this neophyte author on some of the basics of the economic perspective. Any remaining errors are, of course, entirely my own.

9 References

In addition to the references cited below, Wikipedia (http://wikipedia.org) has several useful articles that cover many of the economics-related topics discussed in this paper.

[1] K. J. Arrow, *Essays in the Theory of Risk-Bearing*, North-Holland, Amsterdam, 1971.

[2] A. AuYoung, L. Grit, J. Wiener, and J. Wilkes, Service contracts and aggregate utility functions, in *Proceedings of 15th IEEE International Symposium on High Performance Distributed Computing (HPDC-15)*, pp. 119-131, Paris, France, June 2006.

[3] C. Bartolini, C. Preist, and N. R. Jennings, *A software framework for automated negotiation (revised and updated)*, Technical report HPL–2006–33, HP Laboratories, Palo Alto, CA, USA, February 2006.

[4] M. Becker, N. Borissov, V. Deora, O. Rana, and D. Neumann, Using k-pricing for penalty calculation in Grid markets, in *Proceedings of the 41st Hawaii International Conference on System Sciences (HICSS)*, paper 97, Waikoloa, Big Island, Hawaii, USA, January 2008.

[5] R. F. Bordley and M. LiCalzi, Decision analysis using targets instead of utility functions, *Decisions in Economics and Finance*, 23(1), pp.53-74, Springer-Verlag Italia, Milan, 2000.

[6] R. F. Bordley, Teaching decision theory in applied statistics courses, *Journal of Statistics Education* 9(2), 2001, http://www.amstat.org/publications/jse/v9n2/bordley.html.

[7] G. Cheliotis and C. Kenyon, Autonomic economics, in *Proceedings of IEEE International Conference on E-Commerce (CEC'03)*, pp. 120–127, Newport Beach, California, USA, June 2003.

[8] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, Web Services Description Language (WSDL) 1.1, W3C Note, http://www.w3.org/TR/wsdl, World Wide Web Consortium, 15 March 2001.

[9] A. Dan, D. Davis, R. Kearney, R. King, A. Keller, D. Kuebler, H. Ludwig, M. Polan, M. Spreitzer, and A. Youssef, Web Services on demand: WSLA-driven Automated Management, *IBM Systems Journal*, 43(1), pp.136–158, March 2004.

[10] P. Faratin, C. Sierra, and N. R. Jennings, Negotiation decision functions for autonomous agents, *Robotics and Autonomous Systems*, 24(3), pp.159–182, September 1998.

[11] P. Faratin, C. Sierra, and N. R. Jennings, Using similarity criteria to make negotiation trade-offs, in *Proceedings of the 4th International Conference on Multi-Agent Systems*, pp. 1–19, Boston, MA, USA, July 2000.

[12] B. Huberman and T. Hogg, Distributed computation as an economic system, *The Journal of Economic Perspectives*, 9(1), pp.141–147, Winter 1995.

[13] D. E. Irwin, L. E. Grit, and J. S. Chase, Balancing risk and reward in a market-based task service, in *Proceedings of 13th IEEE Symposium on High Performance Distributed Computing (HPDC)*, pp. 160–169, Honolulu, HI, USA, June 2004.

[14] J. O. Kephart and R. Das, Achieving self-management via utility functions, *IEEE Internet Computing* 11(1), pp. 40–48, January 2007.

[15] R. L. Keeney and H. Raiffa, *Decisions with multiple objectives: preferences and value*, Cambridge University Press, 2003.

[16] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman, Tycoon: an implementation of a distributed, market-based resource allocation system, *Multiagent Grid Systems*, 1(3), pp.169–182, August 2005.

[17] R.J. Lewicki, D.M. Saunders, and J.W. Minton, *Negotiation*. Irwin/McGraw-Hill, Boston, USA, 3rd Edition, 1999.

[18] X. Luo, N. R. Jennings, and N. Shadbolt, Acquiring user tradeoff strategies and preferences for negotiating agents: A default-then-adjust method, *International Journal of Human-Computer Studies*, Vol. 64, pp. 304–321, 2006.

[19] R. P. McAfee, *Introduction to economic analysis*, version 2.0, July 2006. Available from http://www.introecon.com or Lulu Press, Morrisville, NC, USA, 2006.

[20] J. O'Sullivan, D. Edmond, and A. H. M. ter Hofstede, The price of services, in *Proceedings of the 3rd International Conference on Service-Oriented Computing (ICSOC 2005)*, Amsterdam, Netherlands, December 2005, *Lecture Notes in Computer Science*, Vol. 3826, pp. 564–569, Springer, Berlin, Germany, November 2005.

[21] J. O'Sullivan, *Towards a precise understanding of service properties*, PhD thesis, Faculty of Information Technology, Queensland University of Technology, Australia, 2006.

[22] H. Oosterbeek, R. Sloof, and G. van de Kuilen, Differences in ultimatum game experiments: evidence from a meta-analysis, *Experimental Economics*, 7(2), pp.171–188, June 2004.

[23] G. A. Paleologo, Price-at-risk: A methodology for pricing utility computing services. *IBM Systems Journal*, 43(1), pp.20–31, January 2004.

[24] F. I. Popovici and J. Wilkes, Profitable services in an uncertain world, in *Proceedings of Supercomputing (SC|05)*, Seattle, WA, USA, November 2005.

[25] H. Raiffa, J. Richardson, and D. Metcalfe, *Negotiation analysis: the science and art of collaborative decision making*, Belknap Press (imprint of Harvard University Press), Cambridge, MA, USA, 2002.

[26] M. Rabin, Diminishing marginal utility of wealth cannot explain risk aversion, in *Choices, Values, and Frames*, D. Kahneman and A. Tversky (eds.), Cambridge University Press, Cambridge, UK, pp. 202–208, 2000.

[27] R. Raman, M. Livny, and M. Solomon, Matchmaking: Distributed resource management for high throughput computing. In *Proceedings of the 7th IEEE*

International Symposium on High Performance Distributed Computing (HPDC'98), Chicago, IL, USA, pp. 140–146, July 1998.

[28] L. J. Savage, *The foundation of statistics*, Wiley, New York, USA, 1954.

[29] M. Stonebraker, P. M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, and A. Yu, Mariposa: a wide-area distributed database system, *The VLDB Journal*, 5(1), pp.19–34, January 1996.

[30] I. E. Sutherland, A futures market in computer time, *Communications of the ACM*, 11(6), pp. 449–451, June 1968.

[31] N. Taleb and A. Pilpel, Epistemology and risk management, *Risk & Regulation*, Vol. 13, pp. 6–7, London School of Economics, London, UK, Summer 2007.

[32] G. E. Tesauro, W. Walsh, and J. O. Kephart, Utility-function-driven resource allocation in autonomic systems. In *Proceedings of the Second international Conference on Automatic Computing (ICAC'05)*, Seattle, WA, USA, pp. 342–343, June 2005.

[33] I. Tsetlin and R. L. Winkler, On equivalent target-oriented formulations for multiattribute utility. *Decision Analysis archive*, 3(2), pp.94–99, June 2006.

[34] A. Tversky and D. Kahneman, Advances in prospect theory: cumulative representation of uncertainty, *Journal of Risk and Uncertainty*, Vol. 5, pp.297–323, 1992.

[35] J. von Neumann and O. Morgenstern, *The theory of games and economic behavior*, Princeton University Press, Princeton, USA, 1944 (republished 1980).

[36] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta, Spawn: A Distributed Computational Economy, *IEEE Transactions on Software Engineering*, 18(2), pp.103–117, February 1992.

[37] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das, Utility Functions in Autonomic Systems, in *Proceedings of the International Conference on Autonomic Computing (ICAC'04)*, New York, NY, USA, pp. 70–77, May 2004.

[38] *Web Services Agreement Specification (WS-Agreement)*, Open Grid Forum (OGF), USA, September 2006.

[39] D. Zeng and K. Sycara, Bayesian learning in negotiation, *International Journal of Human Computer Studies*, 48(1), pp.125–141, 1998.