SLA Design from a Business Perspective

Jacques Sauvé[1], Filipe Marques[1], Antão Moura[1], Marcus Sampaio[1], João Jornada[2], and Eduardo Radziuk[2]

[1] Universidade Federal de Campina Grande, Brazil
{jacques,filipetm,antao,sampaio}@dsc.ufcg.edu.br
[2] Hewlett-Packard-Brazil
{joao.jornada,eduardo.radziuk}@hp.com

Abstract. A method is proposed whereby values for Service Level Objectives (SLOs) of an SLA can be chosen to reduce the sum IT infrastructure cost plus business financial loss. Business considerations are brought into the model by including the business losses sustained when IT components fail or performance is degraded. To this end, an impact model is fully developed in the paper. A numerical example consisting of an e-commerce business process using an IT service dependent on three infrastructure tiers (web tier, application tier, database tier) is used to show that the resulting choice of SLOs can be vastly superior to ad hoc design. A further conclusion is that infrastructure design and the resulting SLOs can be quite dependent on the "importance" of the business processes (BPs) being serviced: higher-revenue BPs deserve better infrastructure and the method presented shows exactly how much better the infrastructure should be.

1 Introduction

Service Level Agreements (SLAs) are now commonly used to capture the performance requirements that business considerations make on information technology (IT) services. This is done both for services provided in-house and for outsourced services. An SLA defines certain Service Level Indicators (SLIs) and restrictions that such indicators should obey. Restrictions are frequently expressed in the form of Service Level Objectives (SLOs), threshold values that limit the value of SLIs. Some typical SLIs are service availability, service response time, and transaction throughput. The problem examined in this paper is that of designing SLAs; the SLA design problem is informally defined as that of choosing appropriate values for SLOs. For example, should service availability be 99.9%, 99.97%? How is one to choose adequate values? There are other aspects to SLA design (choosing SLIs, choosing measurement methods and periods, choosing penalties, etc.) but these are not considered here.

It is interesting to examine how choosing SLOs is typically done today. Naturally, since SLOs are chosen according to how important a service is to the business, the IT client (a senior business manager) is involved in choosing SLOs. However, as reference [11] has vigorously shown, the methods used are almost

always pure guesswork, frequently resulting in drastic loss or penalties. It is clear that one needs more mature and objective models to properly design SLAs. An approach based on Business Impact Management [3,12] is presented in this paper.

The remainder of the paper is organized as follows: section 2 informally discusses the approach while section 3 formalizes it; section 4 considers an application of the method through a full numerical example; section 5 discusses related work; conclusions are provided in section 6.

2 Gaining a Business Perspective on IT Operations

An informal discussion of the approach adopted here will help the reader follow the formal treatment presented in the next section.

2.1 Addressing IT Problems through Business Impact Management

SLOs must be chosen by taking into account the importance of the IT service on the business. In the approach being described here, this is done by capturing the impact of IT faults and performance degradations on numerical business metrics associated with the business. By considering business metrics, one may say that the approach is part of a new area of IT management called Business Impact Management (BIM) [3,12]. BIM takes Service Management (SM) to a new maturity level since metrics meaningful to the customer such as financial or risk measures are used to gauge IT effectiveness rather than technical metrics such as availability and response time.

For BIM to be successfully applied to the problem at hand, one needs to construct an impact model. Since it is quite difficult to bridge the gap between events – such as outages – occurring in the IT infrastructure and their financial effect on the business, an intermediate level is considered: that of the business processes (BPs) using the IT services. Thus, an impact model is used to map technical service metrics to BP metrics such as BP throughput (in transactions per second) and a revenue model to map BP throughput to a final business metric such as revenue throughput.

Thus, this paper essentially investigates how BIM can be useful in addressing some common IT problems. SLA design was chosen as an example of an activity performed by IT personnel that can be rethought from a business perspective using BIM.

2.2 SLA Design: An Optimization Problem

The IT infrastructure used to provision IT services is designed to provide particular service levels and these are captured in SLAs. Intuitively, a weak infrastructure (with little redundancy or over-utilized resources) has the advantages of having low cost but may generate high business losses – as captured by the BIM impact model – resulting from low availability and customer defections due to

high response times. An infrastructure with much better availability and lower response times will possibly generate lower business losses but may have a much higher total cost of ownership (TCO). Thus, in both cases, total financial outlay (TCO plus business losses) may be high. It thus appears that a middle ground can be found that will minimize this sum. Once this infrastructure yielding minimal financial outlay is found, one may then calculate SLOs such as availability and response time. As a result, SLO thresholds will be *outputs* from the method rather than being chosen in an ad hoc way. These SLOs will be optimal in the sense that they will minimize total financial outlay.

3 Problem Formalization

The optimization problem considered aims to calculate the number of load-balanced resources and the number of fail-over resources to be used in provisioning IT services so as to minimize overall cost (TCO plus business losses). The model considers workloads with fixed averages and static resource allocation. Once this infrastructure is found, SLOs such as service availability, average response time, etc. can be calculated and inserted in the SLA. This section formalizes the SLA Design problem.

3.1 The Entities and their Relationships

Figure 1 shows the entities and their relationships used in the problem formalization. It can also be useful to the reader as a quick reference to the notation employed. The model includes entities both from the IT world and the business world. The business (top) layer consists of several business processes. For simplicity, assume that there is a one-to-one relationship between business processes and IT services. Extension to several services is straightforward but would needlessly complicate the formalism for this presentation. We thus have a set BP of BPs and a set S of services: $S = \{s_1, \ldots, s_{|S|}\}$. The infrastructure used to provision these services consists of a set RC of resource classes.

Service s_i depends upon a set RC_i^S of these resource classes. For example, a service could depend on three resource classes: a Web resource class, an application server resource class and a database resource class. Class RC_j consists of a cluster of IT resources. This cluster has a total of n_j identical individual resources, up to m_j of which are load-balanced and are used to provide adequate processing power to handle incoming load. The resources that are not used in a load-balanced cluster are available in standby (fail-over) mode to improve availability.

Finally, an individual resource $R_j \in RC_j$ consists of a set $P = \{P_{j,1}, \ldots, P_{j,k}, \ldots\}$ of components, all of which must be operational for the resource to also be operational. As an example, a single Web server could be made up of the following components: server hardware, operating system software and Web server software. Individual components are subject to faults as will be described later.

An SLA is to be negotiated concerning these services. For service s_i, the SLA may specify Service Level Objectives (SLOs). The impact model to be presented assumes that BP throughput is lost if the service is unavailable or if response time exceeds a certain threshold. The following SLO parameters are considered for service s_i and will constitute the promise made to the customer in the SLA: A_i^{MIN}, the minimum service availability, $\bar{T}_i$, the average response time, T_i^{DEF}, the response time threshold causing customer defection and $B_i\left(T_i^{DEF}\right) = B_i^{MAX}$, the probability that response time is larger than the threshold.

One may thus summarize the SLA as the four sets: $A^{MIN} = \{\ldots, A_i^{MIN}, \ldots\}$, $T = \{\ldots, \bar{T}_i, \ldots\}$, $T^{DEF} = \{\ldots, T_i^{DEF}, \ldots\}$, $B^{MAX} = \{\ldots, B_i^{MAX}, \ldots\}$.

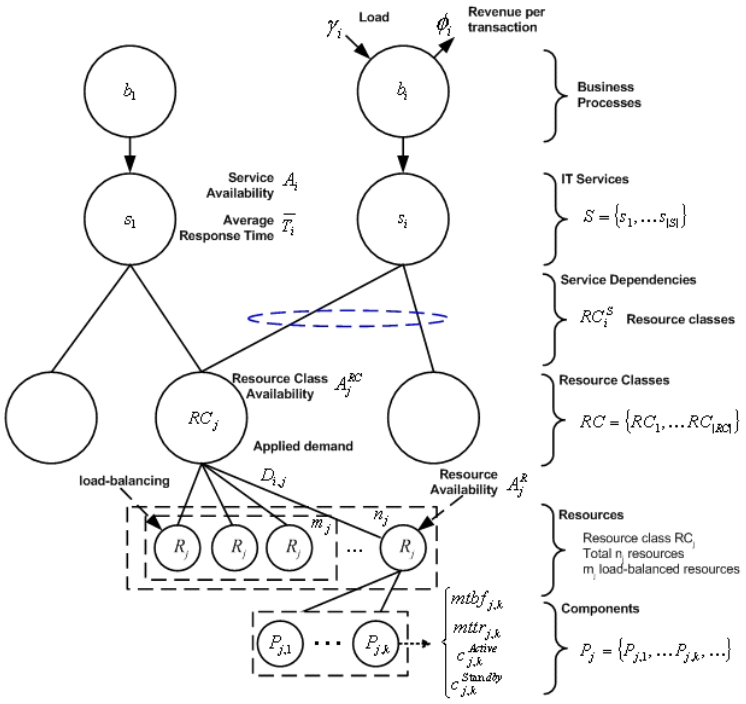

Fig. 1. Entities and their relationships

3.2 The Cost Model

Each infrastructure component $P_{j,k}$ has a cost rate $c_{j,k}^{Active}$ when active (that is, used in a load-balanced server) and has a cost rate $c_{j,k}^{Standby}$ when on standby. These values are cost per unit time for the component and may be calculated as its total cost of ownership (TCO) divided by the amortization period for the component. The cost of the infrastructure over a time period of duration

ΔT can be calculated as the sum of individual cost for all components. In the equation below, j runs over resource classes, l runs over resources and k runs over components.

$$C\left(\Delta T\right) = \Delta T \cdot \sum_{j=1}^{|RC|} \left(\sum_{l=1}^{m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{Active} + \sum_{l=1}^{n_j - m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{Standby} \right) \tag{1}$$

3.3 Loss Considerations

A weak infrastructure costs little but may generate large financial losses due to low availability or high response time. The converse situation is an infrastructure that causes little loss but is expensive to provision. In order to evaluate this tradeoff, financial loss must be calculated. In general, the model used is that at time t, the imperfect infrastructure produces adverse impact on business – or simply business loss – at rate $l\left(t\right)$; the rate is expressed in units appropriate to the business metric used per time unit. As an example, loss rate could be expressed in dollars per second when using dollar revenue as a business metric.

For simplicity, assume that all SLOs are evaluated at the same time and that the evaluation period is ΔT. Thus, the accumulated business impact over the evaluation period is $L\left(\Delta T\right) = \int_0^{\Delta T} l\left(t\right) dt$. Assuming a constant rate (l) of faults over time, we have $L\left(\Delta T\right) = \Delta T \cdot l$. A specific loss model will be discussed below.

3.4 The SLA Design Problem

The SLA Design problem may be stated informally as follows: one wishes to determine the number of servers – both total number of servers and number of load-balanced servers – that will minimize the financial impact on the enterprise coming from two sources: infrastructure cost and financial loss. Formally, a first SLA Design problem may be posed as follows:

Find:	The SLA parameters, the sets A^{MIN}, T, B^{MAX}
By minimizing:	$C(\Delta T) + L(\Delta T)$, the total financial impact on the business over evaluation period ΔT
Over:	$\left\{n_1, \ldots, n_{\left\|RC\right\|}\right\}$ and $\left\{m_1, \ldots, m_{\left\|RC\right\|}\right\}$
Subject to:	$n_j \geq m_j$ and $m_j \geq 1$
Where:	$C(\Delta T)$ is the infrastructure cost over the SLA evaluation period ΔT; $L(\Delta T)$ is the financial loss over the SLA evaluation period ΔT; n_j is the number of resources in resource class RC_j; m_j is the number of load-balanced resources in RC_j.

The set $T^{DEF} = \left\{\ldots, T_i^{DEF}, \ldots\right\}$ which indicates the response time threshold from which defections start to occur is given as input. A typical value is

8 seconds for web-based e-commerce [13]. As a result of the optimization, values for the three sets of SLA thresholds availability: $A^{MIN} = \{\ldots, A_i^{MIN}, \ldots\}$, average response time: $T = \{\ldots, \bar{T}_i, \ldots\}$, and defection probability: $B^{MAX} = \{\ldots, B_i^{MAX}, \ldots\}$ will be found. These are the values to be used in an SLA.

In order to complete the model, one needs to define an impact model and a way to calculate loss $L(\Delta T)$, and the SLOs A^{MIN}, T, and B^{MAX}. The next sections cover this.

3.5 A Specific Loss Model

When IT problems occur, the impact on business may be decreased revenue or increased costs or both. In this paper only decreased revenue is considered, a situation applicable to revenue-generating BPs typical in e-commerce. Each BP has an input load (in transactions per second). Some of this load is lost due to a loss mechanism with 2 causes: service unavailability and customer defection due to high response times. Subtracting lost load from the input load results in the BP transaction throughput (denoted by X). The revenue throughput due to any given business process is $V = X \cdot \phi$ where ϕ is the average revenue per transaction for the business process. The total loss rate, over all BPs is

$$l = \sum_{i=1}^{|BP|} l_i$$

where BP is the set of BPs and l_i is the loss rate due to BP b_i. In the above, we have $l_i = \Delta X_i \cdot \phi_i$. Here, ΔX_i is the loss in throughput (in transactions per second) for BP b_i and ϕ_i is the average revenue per transaction for process b_i.

We consider that the BP is heavily dependent on IT, and thus BP availability A_i is equivalent to the availability of the IT service (s_i) used by the BP. When service s_i is unavailable, throughput loss is total and this occurs with probability $1 - A_i$. We thus have $\Delta X_i^A = \gamma_i \cdot (1 - A_i)$ where ΔX_i^A is loss attributable to service unavailability, γ_i is the input load incident on BP b_i and A_i is the availability of service s_i. When service is available (this occurs with probability A_i), loss occurs when response time is slow. Thus, we have $\Delta X_i^T = \gamma_i \cdot B_i\left(T_i^{DEF}\right) \cdot A_i$ where ΔX_i^T is loss attributable to high response time, $B_i\left(T_i^{DEF}\right) = Pr\left[\tilde{T}_i > T_i^{DEF}\right]$ is the probability that the service response time (the random variable $\tilde{T}_i$) is larger that some threshold T_i^{DEF}. This models customer defection and assumes that a customer will always defect if response time is greater than the threshold (typically 8 seconds for an e-commerce BP).

The total loss in BP throughput is simply the sum of losses due to unavailability and losses due to high response time:

$$\Delta X_i = \Delta X_i^A + \Delta X_i^T = \gamma_i \cdot (1 - A_i) + \gamma_i \cdot B_i\left(T_i^{DEF}\right) \cdot A_i \qquad (2)$$

3.6 The Availability Model

In order to calculate lost throughput, one needs to evaluate the availability A_i of an IT service, s_i. This is done using standard reliability theory [15]. Individual component availability may be found from Mean-Time-Between-Failures (MTBF) and Mean-Time-To-Repair (MTTR) values. Since all components must be available for a resource to be available, the component availabilities are combined using "series system reliability" to yield resource availability A_j^R. Combining resource availability to compute resource class availability (A_j^{RC}) uses "m-out-of-n reliability" since the resource class will be available and able to handle the projected load when at least m_j resources are available for load-balancing. Finally, for service s_i to be available, all resource classes it uses must be available and "series system reliability" is used to calculate service availability (A_i).

3.7 The Response Time Performance Model

The loss calculation depends on $B_i\left(T_i^{DEF}\right)$, the probability that the service response time is larger that some threshold T_i^{DEF}. In order to find this probability, the IT services are modeled using an open queuing model. This is adequate for the case of a large number of potential customers, a common situation for e-commerce. Each resource class RC_j consists of a cluster of n_j resources, of which m_j are load-balanced. Let us examine service s_i. The input rate is γ_i transactions per second. Each transaction demands service from all resource classes in the set RC_i^S. Demand applied by each transaction from BP b_i on class RC_j is assumed to be $D_{i,j}$ seconds. In fact this is the service demand if a "standard" processing resource is used in the class RC_j resources. In order to handle the case of more powerful hardware, assume that a resource in class RC_j has a processing speedup of α_j compared to the standard resource. Thus, service time for a transaction is $D_{i,j}/\alpha_j$ and the service rate at a class RC_j resource for transactions from business process b_i is $\mu_{i,j} = \alpha_j/D_{i,j}$. Finally, since there are m_j identical load-balanced parallel servers used for processing in resource class RC_j, response time is calculated for an equivalent single server [13] with input load $\lambda_{i,j} = \gamma_i/m_j$. Thus the utilization $\rho_{i,j}$ of class RC_j resources in processing transactions from business process b_i is:

$$\rho_{i,j} = \frac{\lambda_{i,j}}{\mu_{i,j}} = \frac{\gamma_i \cdot D_{i,j}}{m_j \cdot \alpha_j} \tag{3}$$

The total utilization ρ_j of class RC_j resources due to transactions from all services is:

$$\rho_j = \sum_{i=1}^{|S|} \rho_{i,j} \tag{4}$$

Observe that, when load is so large that any $\rho_j \geq 1$, then any service depending on that resource class will have $B_i\left(T_i^{DEF}\right) = 1$, since response time is very high for saturated resources.

Now, in order to find $B_i\left(T_i^{DEF}\right)$ when $\rho_j < 1$, let us find the cumulative distribution of response time, $T_i(y) = Pr\left[\tilde{T}_i \leq y\right]$. In this case, the total response time for a transaction from BP b_i is the sum of $\left|RC_i^S\right|$ random variables, one for each resource class used by service s_i. In order to find the probability distribution of a sum of independent random variables, one may multiply their Laplace transforms [14]. In order to make mathematical treatment feasible, assume Poisson arrivals (this is a reasonable assumption for stochastic processes with large population) and exponentially distributed service times. (Observe that although service times may not be independent and exponentially distributed in practice, the optimization step *compares* design alternatives and that is probably insensitive to particular distributions – if they are the same when comparing results.) From queuing theory, the Laplace transform of response time (waiting time plus service time) for a single-server queue is $T^*(s) = a/(s+a)$ where $a = \mu \cdot (1-\rho)$, μ is the service rate and ρ is the utilization. Recall that input load from several services is going to the same resource class. Thus, for the combination of resource classes used by service s_i, we have:

$$T^*(s) = \prod_{j \in RC_i^S} \frac{a_{i,j}}{s + a_{i,j}} \tag{5}$$

where $a_{i,j} = \mu_{i,j} \cdot (1-\rho_j)$. Inverting the transform yields the probability density function of response time, which is integrated to find the cumulative probability distribution function (PDF) of response time, $T_i(y)$. Finally:

$$B_i\left(T_i^{DEF}\right) = Pr\left[\tilde{T}_i > T_i^{DEF}\right] = 1 - T_i\left(T_i^{DEF}\right) \tag{6}$$

Additionally, average response time is typically defined in an SLA and may be found from the Laplace transform as follows:

$$\bar{T}_i = -\left.\frac{dT_i^*(s)}{ds}\right|_{s=0} \tag{7}$$

4 A Numerical Example of SLA Design

The purpose of this section is to go through a complete example and verify the extent to which the method proposed can be useful in designing SLAs, i.e., choosing SLO values. Assume the existence of a single service (the index i is dropped) using three resource classes: a Web resource class (RC_{web}), an application server resource class (RC_{as}) and a database resource class (RC_{db}). In the example, the parameters shown in Table 1 are used, typical for current technology [8]. In that table, tuples such as (a,b,c) represent parameter values for the three resource classes (web, application, database); furthermore, each resource is made up of three components: hardware (hw), operating system (os) and application software (as).

Let us now first get a feeling for the variation of some of these measures. Figure 2 shows how the loss component due to response time (ΔX_i^T) indeed

Table 1. Parameters for example

Parameters	Values	Parameters	Values
T^{DEF}	8 seconds	α_j	(1,1,3)
ϕ	\$1 per transaction	$c_{j,k}^{Active}$ ($/month)	hw =(1100, 1100, 4400) os=(165, 165, 165) as=(61, 30, 660)
γ	14 transactions per second	$c_{j,k}^{Standby}$ ($/month)	hw =(1000, 1000, 4000) os=(150, 150, 150) as=(55, 0, 600)
ΔT	1 month	D_j	(0.05, 0.1, 0.2) seconds
A_j^R (resource availability for R_j)	99.81% (this value is calculated from appropriate MTBF and MTTR values)		

varies as response time rises with increased load. Similarly, one can get a feel for the loss component due to availability (ΔX_i^A) from Figure 3. In that figure, availability is made to improve by changing the number of database machines from 2 to 6, while keeping other infrastructure components constant. The loss due to high response time is very low and is thus not shown in the figure. As one can see, cost increases, loss due to unavailability decreases while "cost + loss" reaches a minimum value for 4 machines.

It is now time to consider the main problem of interest in this paper: that of SLA design. If one were to design the SLA in an ad hoc way, one could approach the problem from the infrastructure side and try to minimize cost while maintaining reasonable service availability and response time. The cheapest infrastructure here is (n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})=(1,1,1,1,1,1). However, this design cannot handle the applied load (average response time is very high) due to saturation of the application server. A second try yields (1,2,1,1,2,1) – more power in the application tier. This yields a monthly cost of \$9141, and SLOs of (average response time=1.5 s, service availability=95.32%). Since this availability is not typically considered adequate, the designer may increase the number of machines in other tiers yielding a design with infrastructure (3,3,3,1,2,1), cost \$22201 and SLOs of (1.5 s, 99.96%). There the designer may rest. We will shortly show that this is not an optimal design.

Alternatively, the designer may base the design on the customer and overdesign with (5,5,5,2,3,1), cost \$37152 and SLOs (0.39 s, 99.998%). None of the above design decisions take loss into account. It is instructive to discover the values for loss for the above designs as well as for the design which minimizes the sum of cost plus loss as shown in section 3.4 (see Table 2).

For the best design, the SLOs are (average response time=0.625 s, availability=99.998%). It has lowest overall financial outlay, and the table clearly shows the high cost of choosing SLOs in an ad hoc fashion: a wrong choice can cost tens or even hundreds of thousands of dollars per month.

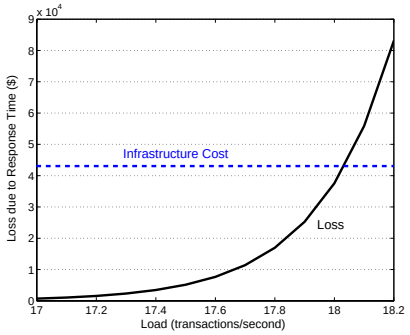

Fig. 2. Effect of Load on Loss

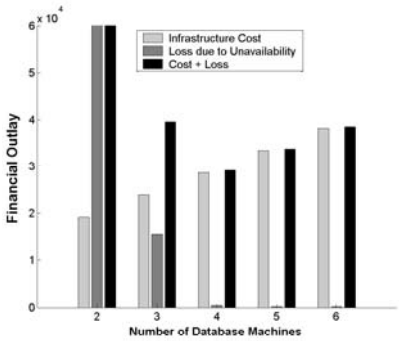

Fig. 3. Sensitivity of Loss due to Redundancy

Table 2. Comparing designs

Infrastructure	**Cost ($)**	**Loss due to Response ($)**	**Loss due to unavailability ($)**	**Cost plus loss ($)**	**The cost of choosing wrong ($)**
(1,2,1,1,2,1)	9141	20886	1697369	1727396	1698274
(3,3,3,1,2,1)	22201	21902	15428	59531	30409
(5,5,5,2,3,2)	37152	0	608	37760	8638
(3,4,4,1,2,2)**(best)**	28576	0	546	29122	0

As a final experiment, it is instructive to see that the best design depends quite heavily on the importance of the business process being serviced. If one lessens the importance of the BP by diminishing the average revenue per transaction by a factor of 10, the best design is (2,4,2,1,2,1), cost $17396, total loss $3243 and SLOs: (average response time=1.5 s, availability=99.97%). In this case, a much lower availability is best and the design is cheaper by $11180 a month than if BP importance were not considered.

5 Related Work

Business Impact Management is a very new area of interest to researchers and practitioners that has not yet been consolidated. In the recent past, some problems typically faced in IT management are being studied through a business perspective [1,2,3,4,5,6,7]. Some examples include incident prioritization [2], management of Web Services [5], Business Process Management [4], etc. These references confirm a general tendency to view BIM as a promising way of better linking IT with business objectives. However, these references offer little in terms of formal business impact models to tie the IT layer to BP or business layers. This is one of our main contributions.

Although this paper stresses aspects of SLA Design, it is also licit to view the work as a method for IT infrastructure design (capacity planning). In this

particular area, [8] describes a tool – AVED – used for capacity planning to meet performance and availability requirements and [9] describes a methodology for finding minimum-cost designs given a set of requirements. However, none of these references consider the problem of capacity planning from a business perspective, using business metrics. Furthermore, response time considerations are not directly taken into account. Finally, [10] considers the dynamic optimization of infrastructure parameters (such as traffic priorities) with the view of optimizing high-level business objectives such as revenue. It is similar in spirit to the work reported here, although the details are quite different and so are the problems being solved (SLA design is not the problem being considered). The model is solved by simulation whereas our work is analytical.

In the area of SLA design, HP's Open Analytics [11] is a response to the downside of designing SLAs with current practices leading to a more formal approach as presented here. Open Analytics dictates that all assumptions leading to a performance decision must be made explicit and that all technical and financial consequences must be explained. "Open auditable mathematics, rather than wet finger in the air responses to requests [...]" must be used although details are not given.

Management by Contract [12] investigates how IT management can decide when it is better to violate an SLA or to keep compliance, according to a utility function that calculates the business impact of both alternatives. It is similar in spirit to our work, although it does not specifically address the problem of SLA design.

6 Conclusions

This paper has proposed a method whereby best values for Service Level Objectives of an SLA can be chosen through a business perspective. Business considerations are brought into the model by including the business losses sustained when IT components fail or performance is degraded. This is done through an impact model, fully developed in the paper. A numerical example consisting of a single e-commerce business process using a single IT service dependent on three infrastructure tiers (web tier, application tier, database tier) was used to show that the best choice of SLOs can be vastly superior to ad hoc design. A further conclusion is that infrastructure design and the resulting SLOs can be quite dependent on the "importance" of the BPs being serviced: higher-revenue BPs deserve better infrastructure and the method presented shows exactly how much better the infrastructure should be.

Much work can be undertaken to improve the results, among which the following are worth noting: a better availability model (such as presented in [8]) can be used to approximate reality more faithfully; the load applied to the business process can be better modeled by following the Customer Behavior Model Graph approach [13]; variations in the load applied to the BPs should be investigated; more complete impact models should be developed to be able to deal with any

kind of BP, not only e-business BPs heavily dependent on IT; finally, the work should be extended to adaptive infrastructures and dynamic provisioning.

Acknowledgments. We would like to acknowledge and thank the Bottom Line Project team. This work was developed in collaboration with HP Brazil R&D.

References

1. V. Machiraju, C. Bartolini and F. Casati (2004), "Technologies for Business Driven IT Management", In L. Cavedon, Z. Maamar, D. Martin, and B. Benatallah (editors) *Extending Web Services Technologies: the Use of Multi-Agent Approaches*, Kluwer Academic Publishers, 2004.
2. C. Bartolini and M. Sallé (2004), "Business Driven Prioritization of Service Incidents", In Proc. *15th IFIP/IEEE Distributed Systems: Operations and Management (DSOM 2004)*, 15-17 November 2004, Davis, CA, USA.
3. P. Mason, A New Culture for Service-Level Management: Business Impact Management, IDC White Paper.
4. F. Casati, M. Castellanos, U. Dayal, M. Hao, M. Sayal and M.C. Shan, Business Operation Intelligence Research at HP Labs, In *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2002.
5. F. Casati, E. Shan, U. Dayal and M.C. Shan, Business-Oriented Management of Web Services, In *Communications of the ACM*, October 2003.
6. Z. Liu, M. Squillante and J. Wolf, On Maximizing Service-Level Agreement Profits, In *ACM Electronic Commerce Conference*, October 2001.
7. Y. Diao and J. Hellerstein and S. Parekh, A Business-Oriented Approach to the Design of Feedback Loops for Performance Management, In *Proc. of the 12th International Workshop on Distributed Systems: Operations and Management*, 2001.
8. G. Janakiraman, J. Santos, Y. Turner; Automated Multi-Tier System Design for Service Availability, In *Proceedings of the First Workshop on Design of Self-Managing Systems*, June 2003.
9. D. Ardagna, C. Francalanci, A Cost-Oriented Methodology for the Design of Web-Based IT Architectures, In *Proceedings of the 2002 ACM symposium on Applied Computing*, 2004.
10. S. Aiber, D. Gilat, A. Landau, N. Razinkov, A. Sela, and S. Wasserkrug, "Autonomic Self-Optimization According to Business Objectives", In *Proceedings of the International Conference on Autonomic Computing*, 2004.
11. R. Taylor, C. Tofts; Death by a thousand SLAs: a short study of commercial suicide pacts, HP Technical Report, January 2005.
12. M. Sallé and C. Bartolini (2004), "Management by Contract", In *Proceedings of the 2004 IEEE/IFIP Network Operations and Management Symposium*, Seoul, Korea, April 2004.
13. D. Menascé, V. Almeida and L. Dowdy, "Performance by Design", Prentice Hall PTR, 2004.
14. L. Kleinrock, *Queuing Systems, Vol I: Theory*, Wiley, New York, 1975.
15. K. S. Trivedi, *Probability & Statistics with Reliability, Queuing and Computer Science Applications*, Prentice-Hall, 1982.