Managing eBusiness on Demand SLA Contracts in Business Terms Using the Cross-SLA Execution Manager *SAM*

Melissa Buco, Rong Chang, Laura Luan, Chris Ward, Joel Wolf, and Philip Yu
IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532, USA
{mjbuco,rong,luan,cw1,jlwolf,psyu}@us.ibm.com

Tevfik Kosar
Dept. of Computer Science
University of Wisconsin-Madison
1210 West Dayton Street, Madison, WI 53706
kosart@cs.wisc.edu

Syed Umair Shah
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
umair@cs.cmu.edu

Abstract

It is imperative for a competitive e-business service provider to be positioned to manage the execution of its service level agreement (SLA) contracts in business terms (e.g., minimizing financial penalties for service-level violations, maximizing service-level measurement based customer satisfaction metrics). This paper briefly describes the design rationale of an integrated set of business-oriented service level management (SLM) technologies under development in the SAM project at IBM T.J. Watson Research Center. The e-business SLA execution manager SAM, (1) enables the provider to deploy an effective means of capturing and managing contractual SLA data as well as provider-facing non-contractual SLM data; (2) assists service personnel to prioritize the processing of action-demanding quality management alerts as per the provider's SLM objectives; and (3) automates the prioritization and execution management of approved SLM processes on behalf of the provider, including assigning SLM tasks to service personnel.

1: Introduction

In recent years it has become increasingly desirable for companies to outsource the development and management of their e-business applications and/or processes to leverage rapid innovations in Web computing technologies and mitigate the serious worldwide shortage of information technology (IT) skills. Various e-business service providers (e.g., ASP's, ISP's, MSP's, xSP's) are evolving to help these e-business companies to focus on the growth of their core competency in a cost-effective manner. One mechanism to improve their competitiveness is to manage service functions and qualities in business terms based on the Service Level Agreements (SLAs) signed with customers.

An SLA is a contract that specifies the minimum expectations and obligations that exist between a service provider and a service customer [1, 2, 3, 4, 5]. An e-business SLA contract specifies, among others, the outsourced service functions, service quality measurement criteria, service-level evaluation rules, and ramifications of failing to meet (or indeed exceeding) quality standards (or *service-level targets*, SLTs). SLTs in an SLA contract can be stated based upon objective quantitative measurement of computing system availability or performance (e.g., monthly availability of Individual Web Server will be no less than 99.7%) or business process efficiency or effectiveness (e.g., no less than 93% of Severity 1 problems are responded within 30 minutes monthly). The refund policies for service-level violations can be specified relative to the service cost (e.g., credit customer one day of the service cost if the outsourced e-business infrastructure is unavailable more than 15 minutes a day) or in absolute terms (e.g., credit customer two thousand dollars if a monthly average network latency across the provider ISP access links to the ISP's backbone is higher than 95 milliseconds).

From the provider's viewpoint, offering a few sets of customer-neutral service functions atop a common service delivery infrastructure exploits economy of scale better than pursuing a high degree of customizability of its service functions for every potential customer. However, this customer-neutral approach to establishing SLAs seems effective only for primitive e-business services (e.g., server co-location services offered by WorldCom/UUNET [6]). The other classes of e-business SLA contracts (including the ones that incorporate one or more primitive e-business services) normally require non-trivial customer-specific customization of the provider's service offerings to accommodate a customer's unique data or business process management needs. When the number of such customer-specific SLA contracts grows, the difficulty of cost-effectively meeting *service level management* (SLM) objectives increases. One of the most pressing challenges facing an e-business service provider today, therefore, is to manage the execution of *all* its SLA contracts in business terms (e.g., minimizing financial penalties for service-level violations, maximizing service level measurement based customer satisfaction metrics).

SAM is an SLA contract execution manager under development at the T.J. Watson Research Center. *SAM*'s SLM technologies (see Figure 1) would, among other capabilities, (1) enable the provider to deploy an effective means of capturing and managing contractual SLA data as well as provider-facing non-contractual SLM data; (2) assist service personnel to prioritize the processing of action-demanding quality management alerts as per provider's SLM objectives; and (3) automate the prioritization and execution management of single- or multi-task SLM processes on behalf of the provider, including assigning SLM tasks to service personnel and/or software agents. As illustrated in Figure 1, by providing a generic framework for capturing and managing SLA/SLM data and management processes, *SAM* supports a wide variety of on-demand services (e.g. Web Hosting, Managed Storage, Backup and Restore) without modification. Notice that service fulfillment of a customer's contract may be geographically distributed with *SAM* providing service management support to each of the service delivery centers.

The remainder of this paper presents a *SAM*-enabled generic approach to economically managing the execution of SLM processes by furnishing the design rationale of *SAM*'s SLM technologies, especially *SAM*'s SLA semantic model and business-impact based SLM process execution management technology.

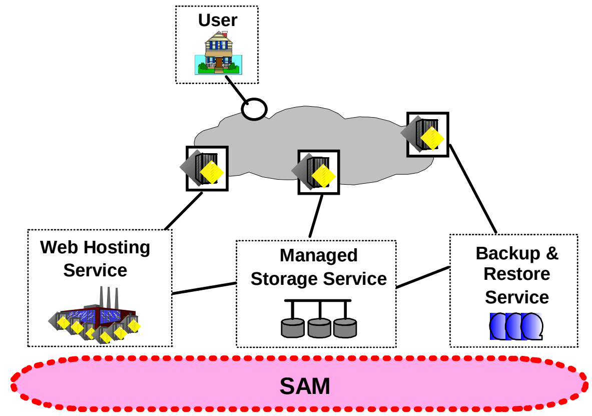

Figure 1. SAM Supporting Unified Execution Management of e-business on Demand Outsourcing Contracts in Business Terms.

Section 2 presents non-trivial data management needs for economically executing SLM processes, our comparative semantic analysis of nine real e-business SLA contracts/templates, and *SAM*'s SLA semantic model. Section 3 elaborates the design rationale of *SAM*'s distributed autonomous SLM technologies and illustrates quality alert prioritization algorithms with event-driven simulation based comparisons. Finally, Section 4 concludes the paper with a summary of the current status of the *SAM* project.

2: Data Management Needs for Economically Executing SLM Processes

Contrasting with the service contracts whose quality cannot be quantitatively measured, e-business SLA contracts include measurable production-time quality standards (a.k.a., SLTs) with penalty clauses for service-level violations. The quality standards can be specified in terms of service system availability, network latency, problem resolution response time, on-demand resource provisioning turnaround time, end-to-end transaction response time, etc. In order for the customer and the provider to objectively determine service-level violations (as well as to avoid related disputes), the methodologies used for generating valid quality measurement data must be specified as clearly as possible in the SLA contracts. Workload admission control mechanisms and policies, for example, could be included in system performance related service level specifications.

It is often unappreciated that service-level evaluation related data are a proper subset of the contractual data (e.g., SLA refund/reward computation algorithms) that are essential for the economical execution management of SLA contracts. We call the set of SLM related contractual data the *SLA Data*. Associated with this SLA Data is a corresponding set of provider-facing (i.e. the customer is unaware) *SLM Data*. For example, a managed storage services contract that offers virtual disk space to a customer with an availability SLT requires the provider to manage the mapping between the virtual disk space and the corresponding real storage resources. The SLA Data in this case comprises all data attributes associated with the availability of the virtual disk, including required capacity, availability SLT, pricing, etc. The SLM Data includes the data attributes associated with this mapping (the physical storage servers' identities, their allocated capacity, etc.). While the SLM Data and the SLA-SLM data relationships must be managed by the provider as well, such non-contractual implementation details need not be exposed to the customer.

In order to economically manage the execution of its e-business SLA contracts, a provider must find a generic means of capturing and managing the SLA Data, the SLM Data, and the relationships between them. To the best of our knowledge, a satisfactory means of doing this has yet to be identified. As a start we have completed an in-depth comparative semantic analysis of nine typical commercial e-business SLA contracts/templates [7], including, in total, 36 *service-level guarantees* (SLGs) and 79 *service-level intents* (SLIs), which are basically SLGs less penalty clauses for service-level violations. Based upon insights from this comparative semantic analysis, we identified a generic set of SLA semantic elements and relationships that can be used to capture the SLM-related semantics of all of the nine real SLA contracts/templates.

2.1: A Comparative Semantic Analysis of Real e-Business SLA Contracts/Templates

Table 1 summarizes the real, typical e-business SLA contracts/templates we reviewed in-depth while creating *SAM*'s SLA semantic model. The first column provides a reference name for the contract: UunetColo, UunetNS [6], EarthLink [8], WH1, WH2, WH3, WH4, STL and UKERNA [9]. The second column indicates the contract type: three network-oriented service contracts (UunetColo, UunetNS, EarthLink), four IBM internal e-business Web Hosting contracts (WH1-WH4) and two IT outsourcing contracts (STL - an IBM internal contract and UKERNA). The third (and forth) column indicates the number of SLGs identified in each contract, and the fifth (and sixth) column the number of SLIs. The SLGs and SLIs are further divided into availability/performance categories.

Contract Name	Contract Type	SLGs		SLIs		TOT-AL
		Avail.	Perf.	Avail	Perf.	
1. UUNET Colo	Network+Colocation	2	4			6
2. UUNET Network	Network	1	3			4
3. EarthLink	Network	1				1
4. Web Hosting 1	eBusiness Hosting	1	1			2
5. Web Hosting 2	eBusiness Hosting	1				1
6. Web Hosting 3	eBusiness Hosting	3	3		1	7
7. Web Hosting 4	eBusiness Hosting	9	7			16
8. STL	IT			10	4	14
9. UKERNA	IT			12	52	64
TOTAL		18	18	22	57	115

Table 1. SLGs and SLIs in Nine Real SLA Contracts/Templates.

We noticed significant differences between the degrees of clarity of the SLGs and SLIs in those SLAs. The SLGs were stated with greater specificity and covered only a few common service-levels (e.g., "availability"). The SLIs were described less precisely and often addressed more complex system performance management issues (e.g., "The volume of traffic itemized into the wrong category shall not exceed 0.5% of the total accounted traffic for each institution" [UKERNA]).

The 75 performance service-levels in the reviewed SLAs are classified into two groups:

36 *IT System* service-levels, further divided into: Network (e.g., Network Latency [UunetColo]), Application (e.g., SQL Response Time [UKERNA]) and Server (e.g., VM Response Time [UKERNA]).

39 *Business Process* service-levels, further divided into: HelpDesk (e.g., "Number of abandoned calls in one day" [WH4]), Outage Notification (e.g., "Provider will notify Customer of service outage within 15 minutes" [UunetColo]), Service Responses (e.g., "A response time within five working days to a request for delegation in the target domain of 99.5%" [UKERNA]), Report Timeliness (e.g., "Provider shall issue invoices as soon as practical after the start of the service period and not more than two months after the start of the service period" [UKERNA]) and Maintenance (e.g., "Total maintenance time shall not be more than 0.5% of service time, averaged over the year" [UKERNA]).

Our comparative analysis of the SLA contracts/templates yielded several non-trivial observations:

The evaluation of a service-level (especially performance service-levels) usually uses a hierarchy of quality measurement threshholds rooted at the SLT (e.g., "99.3% [of Severity 1 problems are resolved] within 30 minutes of the Start Time" [WH3]).

A single SLG refund/reward computation may use the evaluation results for several service-levels (e.g., "A Service Level Default occurs when Provider fails to meet a Minimum Service Level during any month of the Term, at any time, or fails to meet an Expected Service Level with a Performance Category on four (4) or more occasions during a calendar twelve (12) month period following the Acceptance Date Plus five (5) months" [WH4]).

Many SLAs include clauses that call for root cause analyses when qualifying availability/performance measurement data (e.g., "This SLA objective specifically does not include failures caused by Customer" [WH2]), often as a result of necessary legal phraseology. These clauses make it impossible to automatically generate correct service-level reports without implementing the necessary measurement data qualification process, which presently often requires human intervention.

A customer's reaction to abnormal service conditions may affect the outcome of official service-level reports (e.g., "Outages will be counted as Power Unavailability only if Customer opens a trouble ticket with UUNET Customer support within five days of the outage" [UunetColo]).

2.2: SAM's SLA Semantic Model

Our comparative semantic analysis of the nine real SLA contract/templates resulted in a generic SLA semantic model for *SAM*. The model has been subsequently validated against more than 60 contracts (over 200 SLI/SLG's) including those from the Information Technology Association of America (ITAA) Application Service Provider (ASP) SLA Library [10] and is believed to apply to a broad set of contracts. A UML [11] diagram showing the model is presented in Figure 2 (with additional details in [7]). The diagram depicts the primary relationships between the principal semantic elements from the viewpoint of SLA execution management. The relationships are labeled relationships such as "uses" and "includes". To allow easy top-to-bottom reading, some relationships are passive (e.g., "is generated by").

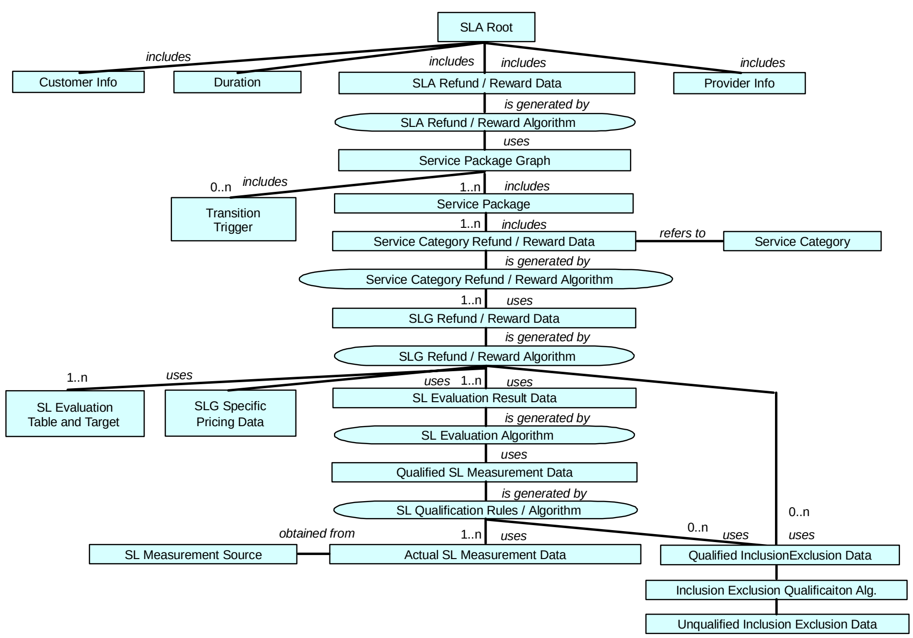

Figure 2. Semantic Model and Relationships for Exposed Business Impact Computation of SLAs.

Starting from the top, the diagram shows "SLA" is composed of customer information (customer ID, contact personnel, etc.), contract duration (start date, end date, termination rules and conditions, etc.), SLA-wide refund/reward data (amount of refund or reward that should apply over the settlement interval after considering all SLA-wide terms and conditions, e.g., a refund limit of 15% regardless of the severity of service-level violations) and provider information (provider ID, contact personnel, etc.).

The relationships, represented in the UML by solid lines, are an essential component in the semantic model and define specific interactions between the principal semantic elements. We highlight the more subtle ones:

Qualified SL Measurement Data is generated by application of SL Qualification Rules/Algorithm to the Actual SL Measurement Data, i.e., "raw" service-level measurement data. In computing service availability, for example, the SL Measurement Data includes "Service Up" and "Service Down" times. SL Qualification Rules are extracted from the contract, e.g., "Fault or negligence by the Provider" [WH1]. The qualification rules also take as input "Qualified Inclusion/Exclusion Data" (such as customer approved downtime and maintenance window start/stop times).[1] Our review of the data models of commercial service-level monitoring tools indicates that this crucial step is often overlooked, leading to the use of unqualified quality measures in the service-level evaluation process or to complex corrections being applied at the service-level evaluation phase. The output from the various qualification rules is qualified (i.e., validated) SL measurement data and qualified inclusion/exclusion data. This latter phase is also important because contracts frequently place restrictions on the excusable downtime per month (e.g., "compliance with scheduled maintenance windows" [WH3]).

A single SLG Refund/Reward Algorithm may receive several SL Evaluation Data sets corresponding to different service-level evaluations when computing its output (SLG Refund/Reward Data). An example of this was provided in Section 2.1 ("A Service Level Default occurs when Provider fails to meet a Minimum Service Level during any month of the Term..." [WH4]). In this example, there are actually three separate service-levels in the guarantee (*Monthly Minimum*, *Monthly Expected* and *Yearly Average*) which collectively dictate the annual credit due. It is also important to notice that the Refund/Reward is computed from the *number of occasions the expected service-level is not met*, rather than the actual value of the measured service-level.

[1] In some contracts, it is necessary for the measurement data qualification rules to be aware not only of the local site's maintenance window, but also the maintenance windows of others. E.g., "Provider shall perform maintenance in a manner that ensures that one Web Hosting Environment with Geographically Diverse Web Hosting Environments will be fully operational and accessible at all times" [WH3] requires tracking several environments simultaneously.

The overall SLA Refund/Reward Algorithm (or SLA business impact evaluation algorithm) uses an accumulation of individual SLG Refund/Reward data sets, organized into Service Categories (e.g., network service, server farm, help desk, etc.) along with SLA-wide policy statements to compute the actual refund/reward due the Customer. E.g., "A Maximum of 25% of the Customer's monthly fee will be credited each month (limit one credit per [network] line per month)." [EarthLink]. Referring to Table 1, we note that a simple network access contract may have only a single SLG [EarthLink], whereas some contracts reviewed have several Service Categories and many separate service-levels. E.g., UKERNA has 12 Service Categories and 64 different SLIs. Many of these require tracking of multiple service-levels (e.g., service-levels for "Total Time" and "Prime Time").

The SLA semantic model includes a notion of *Service Package Graphs*, composed of Service Packages and Transition Triggers, which reside between the Service Category Refund/Reward Data and the SLA Refund/Reward Algorithm (see Figure 2). They enable the model to capture contractual agreements on how quality standards and SLA refunds/rewards details can change. E.g., "Customer credit for the *first* month of a new order, which meets the [refund] requirements, is 25% of the prorated monthly fee" [EarthLink] or "Customer may add or delete Performance Categories by sending written notice to Provider at least ninety (90) days prior to the date that such new Performance Categories are to be effective" [WH3]. In both of these instances, the contract defines rules and procedures for adjusting the algorithms/data and/or service-level measurements used in computing the overall refund/reward. A Service Package can be considered as a data container for a set of Service Categories under contract-defined circumstances. The transition from one Service Package to another is accomplished by some contract-defined event (e.g., a date, as in five months after the start of the contract, or a new service-level). It is possible that an intermediate Service Package is needed to "tidy" up the service-level computations required during the transition.

3: Business-Impact Based SLM Process Execution Management in *SAM*

An *SLM event* conveys a point-in-time situation (e.g., "server X goes down", "server X goes up", "new resource provisioning request X arrives at time Y", etc.) that deserves the provider's attention as per contract-specified quality measurement criteria and

SLTs. Common operations management practices do not distinguish SLM events from other service management events that have little to do with contractual service-level evaluation rules. Service personnel are generally unaware of the business impact (e.g., revenue loss) a specific service-level violation could create, though they can use availability/performance monitoring and trend analysis products to trigger the execution of service management processes. They normally use a best-effort approach to handle action-demanding service management alerts/problems, though customer-reported problems are usually contractually classified into a set of "severity levels" based upon problem response/resolution time requirements. Such best-effort based labor-intensive approaches to managing the execution of e-business SLA contracts are not appealing to the providers when cost, efficiency and shortage of IT skills are pressing issues.

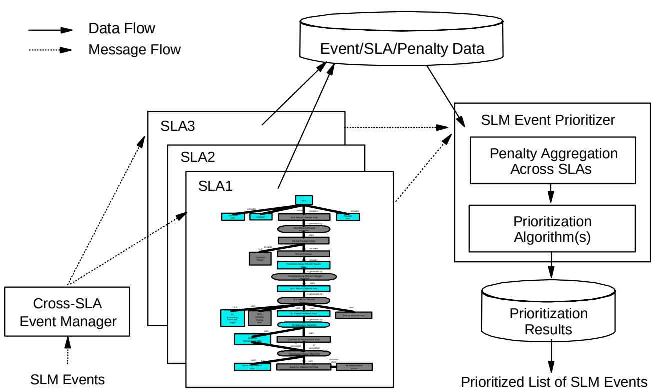

Figure 3. Business-Impact Based SLM Event Prioritization in SAM.

SAM helps an e-business service provider to deploy a business-impact based SLM process management solution by two means:

- Assist service personnel to prioritize the processing of action-demanding SLM events as per provider's SLM objectives; and
- Automate the prioritization and execution management of approved SLM processes on behalf of the provider, including assigning SLM tasks to service personnel.

Figure 3 illustrates the working principle of *SAM*'s (cross-SLA) SLM event prioritization scheme. *SAM* is a distributed management environment in which each established SLA contract is associated with one and only one *SLA management object* (SMO), which, in essence, transforms the associated SLA contract into an active, first-class autonomous computing entity in *SAM* and facilitates the life cycle management of the contract. When a new SLA-neutral SLM event (e.g., "Server X down" event for a shared server X) arrives, *SAM*'s *Cross-SLA Event Manager* distributes it to the set of relevant SMOs. Each of the target SMOs then processes the event as per its own SLA/SLM data (e.g., quality measurement statistics, service-level evaluation related software modules, refund/reward computation modules, etc.), updates action-completion-time based SLA-specific penalty functions in the system, and informs *SAM*'s *SLM Event Prioritizer* of the changes. The *SLM Event Prioritizer* aggregates all of the penalty functions for each SLA-neutral action-demanding SLM events, prioritizes the events based upon their respective aggregated penalty functions, and makes the prioritization results available to its client applications (e.g., Web browsers). Figure 4 illustrates the exposed[2] business impact across all contracts of two outstanding events (servers S1 and S2 down) based on the contract refund/reward functions. In this instance each of the penalty plots (S1 down, S2 down) corresponds to availability guarantees with the business impact computed from a table-based refund/reward. The overall impact of these two outstanding events is the aggregation of their individual impacts.

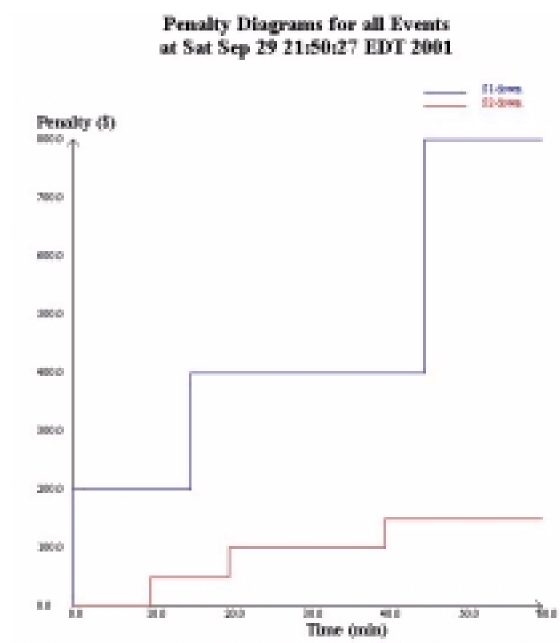

Figure 4. Exposed Business-Impact of Outstanding SL Management Events.

Figure 5 exemplifies three prioritization algorithms we have implemented in our *SAM* prototype, assuming

[2] The financial impact over time if the provider fails to resolve this issue, per the signed contract.

the provider wants to minimize the financial risk of service-level violations.

The penalty diagram in the figure shows there were two action-demanding SLM events (“S1 down” and “S2 down”) when the diagram was generated. It also shows the provider was exposed to $200.00 revenue loss if the needed SLM actions for “S1 down” could not be completed within 15 minutes (relative to time the diagram was generated) and $600.00 within 45 minutes. Similarly, the provider was exposed to $50.00 revenue loss if it failed to complete the SLM process for “S2 down” within 10 minutes, $100.00 within 20 minutes, and $150.00 within 40 minutes.

Earliest Higher Penalty Time - Complete Order at Sat Sep 29 21:50:27 EDT 2001

Rank	Event	Time to Next Penalty	Additional Penalty
1	S2 Down	10	$50.0
2	S1 Down	15	$200.0
3	S2 Down	20	$100.0
4	S2 Down	40	$150.0
5	S1 Down	45	$600.0

Earliest Higher Penalty Time - Partial Order at Sat Sep 29 21:50:27 EDT 2001

Rank	Event	Time to Next Penalty	Additional Penalty
1	S2 Down	10	$50.0
2	S1 Down	15	$200.0

Higher Penalty Rate First - Partial Order at Sat Sep 29 21:50:27 EDT 2001

Rank	Event	Time to Next Penalty	Additional Penalty
1	S1 Down	15	$200.0
2	S2 Down	10	$50.0

Figure 5. Examples of Business-Impact Based SLM Event Prioritization Algorithms.

The “Earliest Higher Penalty Time - Complete Order (EHPT-CO)” algorithm sorts all of the exposed (or predicted) penalty points in chronological order based upon their respective penalty time. The “Earliest Higher Penalty Time – Partial Order (EHPT-PO)” algorithm takes into account only the earliest penalty point of each SLM event. Instead of sorting the penalty points by their respective penalty times, the “Higher Penalty Rate First – Partial Order (HPRF-PO)” algorithm sorts the earliest penalty point of each SLM event based upon the ratio of extra penalty amount and time to the penalty point. Notice that in this ordering the service personnel are advised to resolve S1 before S2. Although the output generated by EHPT-PO or HPRF-PO appears more concise and user friendly than that of EHPT-CO, EHPT-CO output facilitates exploiting estimated resolution times for events. For example, if it would take more than 10 minutes to complete the needed SLM process for “S2 down” and less than 15 minutes for “S1 down”, the provider should execute the SLM process for “S1 down” first to optimize its SLM objective. EHPT-PO would not allow this capability. Yet another approach might be “Higher Penalty Rate First – Complete Order (HPRF-CO)”. In this approach we would compare all rates, not just those for the first penalty points. By exploiting this additional information the algorithm will exhibit more robust behavior. We note that the penalty amount data is shown the figure to facilitate the presentation, and need not be shown to service personnel in practice.

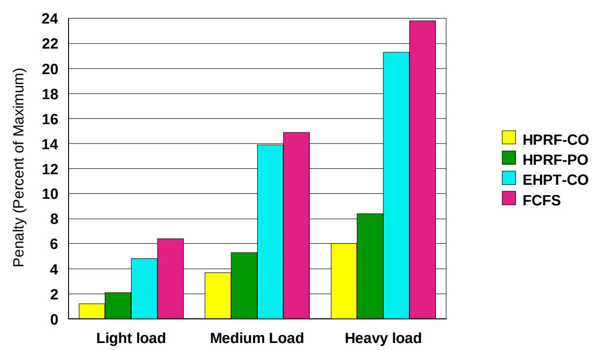

Figure 6. Relative Merit of Selected Prioritization Algorithms Based on Simulation.

Figure 6 shows the relative merits of these prioritization algorithms as well as that of a simple “First Come First Serve (FCFS)” policy. This somewhat ad-hoc algorithm simply orders the SLM events in order of arrival. EHPT-PO is not illustrated because it will perform worse than EHPT-CO. The figure is based on a set of event-driven simulation experiments. The model assumes: a Poisson arrival of jobs (low:med:high = 1:2:3) with low arrival rate chosen to achieve the threshold penalties, five job types (normal distribution), 10 employees, and an average of three penalty points per job. Additional details are provided in [12]. The HPRF algorithms do the best here, essentially because they capture the most information. Specifically, they employ both columns of data in Figure 5. Not surprisingly, HPRF-CO performs better than HPRF-PO. The EHPT-CO algorithm employs only the “Time to Next Penalty” column, not the “Additional Penalty” column. Its performance is not significantly better than that of FCFS, which uses neither column.

SAM's SLM event prioritization capability can be packaged and deployed in a service management center as an SLM decision support tool for service personnel. It can also be integrated with an SLM process execution manager (or an SLM action manager) to automate the task of prioritizing and controlling the executions of workflow-engine-driven SLM processes (including assigning SLM tasks to service personnel) on behalf of the provider. Although exploitation of this advanced SLM action management capability requires the provider to implement better SLM disciplines and to gather more SLM data (e.g., relationships between SLM events and SLM processes, SLM process definitions and related resource requirements, etc.), it makes SAM more effective in helping the provider execute its e-business SLA contracts in business terms [12].

4: Summary and Work in Progress

It is imperative for a competitive e-business service provider to be able to manage the execution of its SLM processes in business terms. Our SLA contract execution manager *SAM* would, among other benefits, (1) enable the provider to deploy an effective means of capturing and managing contractual SLA data as well as provider-facing non-contractual SLM data; (2) assist service personnel to prioritize the processing of action-demanding quality management alerts as per provider's SLM objectives; and (3) automate the prioritization and execution management of approved SLM processes on behalf of the provider, including assigning SLM tasks to service personnel.

We have created a generic SLA semantic model for *SAM*, and are developing an XML-based extensible SLA Specification Language to facilitate integrated management of SLA/SLM data and related programming modules. We have created an implementation of the *SAM* distributed system (with business-impact based SLM event prioritization capability) using XML, Java, SOAP, DB2, and MQSeries workflow technologies. Preliminary event-driven, simulation-based evaluations of our SLM task/resource optimizing scheduler show *SAM* can significantly reduce the financial risk of service-level violations relative to other common SLM job/workforce scheduling algorithms [12], such as FCFS, earliest deadline, etc. Our development and internal pilot deployment experience with *SAM* suggests our SLA execution management approach is both practical and useful.

References

1. Hiles, A., "The Complete IT Guide to Service Level Agreements - Matching Service Quality to Business Needs," *Rothstein Associates Inc.*, Brookfield, Connecticut, USA, 1999/2000 Ed.
2. Verma, D., "Supporting Service Level Agreements on IP Networks," McMillan, 1999.
3. Sturm, R., Morris, W. and Jander, M., "Foundations of Service Level Management," *SAMS*, 2000.
4. ASPIC Best Practices Committee, "A White Paper on Service Level Agreement," *Application Service Provider Industry Consortium*, November 2000.
5. Frey, N., Matlus, R. and Maurer, W., "A Guide to Successful SLA Development and Management," *Gartner Group Research, Strategic Analysis Report*, October 16, 2000.
6. Worldcom, "Service Level Agreements", http://www1.worldcom.com/us/legal/sla
7. Ward, C., Buco, M., Chang, R., and Luan, L, "A Generic SLA Semantic Model for the Execution Management of e-Business Outsourcing Contracts," in *Proc. of 3rd International Conference on e-Commerce (EC-Web 2002)*, September 2002.
8. EarthLink, "EarthLink Service Agreements," http://www.earthlink.net/biz/broadband/dedicated/agreement/
9. UKERNA, "Service Level Agreement Between JISC and UKERNA", *UKERNA*, April 2001, http://www.jisc-tau.ac.uk/sla/ukerna
10. ITAA, "ITAA's ASP SLA Resources," http://www.itaa.org/asp/slalibrary.htm
11. OMG, "Unified Modeling Language (UML)," *Object Management Group*.
12. Wolf, J., Buco, M., Chang, R., Luan, L., Ward, C., Yu, P. "On Workforce Scheduling to Minimize SLA Contract Penalties," IBM Technical Report, 2002.