

UNIVERSITY OF OREGON
APPLIED INFORMATION MANAGEMENT

Presented to the Interdisciplinary
Studies Program:
Applied Information Management
and the Graduate School of the
University of Oregon
in partial fulfillment of the
requirement for the degree of
Master of Science

**Toward Better Understanding
of Service Level Agreement
(SLA) Expectations, for
Improved Conformance Among
Stakeholders in a Distributed,
Dynamic, Composite Software
Solution**

CAPSTONE REPORT

**Cameron J. Ough
Senior Engineer
Intel Corporation**

University of Oregon
Applied Information
Management
Program

July 2008

722 SW Second Avenue
Suite 230
Portland, OR 97204

(800) 824-2714

Service level Agreements in Distributed Composite Software Solutions

Approved by

Dr. Linda F. Ettinger

Service level Agreements in Distributed Composite Software Solutions

**Toward Better Understanding of Service level Agreement (SLA) Expectations,
for Improved Conformance Among Stakeholders in a
Distributed, Dynamic, Composite Software Solution**

Cameron J. Ough

Senior Engineer

Intel Corporation

Abstract

Software development has evolved towards increased reuse of pre-existing components and delivery of functional capabilities through service-oriented architectures (SOA). The situation drives service owners to rely on multiple autonomous partners in the service delivery, requiring design of contractual agreements termed service level agreements (SLAs), which are established to provide business obligation guarantees. This study reviews challenges for stakeholders of SOA applications related to specification, implementation, validation, and audit of SLAs in today's distributed, composite software systems.

Table of Contents

Table of Contents5

List of Figures and Tables7

Introduction9

Description.....9

Research Problem.....10

Audience and Significance12

Limitations.....15

Writing Plan Preview18

Definitions19

Research Parameters24

Search Strategy Report.....24

Quality of Service (QoS) and SOA24

Documentation approach.....26

Literature Evaluation Criteria.....27

Writing Plan29

Annotated Bibliography32

Review of the Literature52

Part 1: Traditional application of SLA to service delivery52

Part 2: Introduction of SOA-based software services54

Part 3: Towards convergence of SLA tools/standards for SOA solutions.....61

Conclusions	66
Gap Analysis.....	66
Suggestions for Specific Solution Proposals That Could Be Revisited.....	68
Recommendations for Stakeholders of Composite Software Solutions.....	70
References	73

List of Figures and Tables

Figure 1. Service provider delivery via Partner entity 15

Figure 2. Traditional service delivery model, single communication provider (ISP)57

Figure 3. Modern service delivery model, multiple communication providers (ISPs).....57

Table 1. Search Grading.....25

Table 2. Initial Search Results.....25

Introduction

Description

As noted by Gu & Lago (2007), over recent years, a major shift in software development has been to move away from traditional, standalone software applications towards more component-based solutions. These solutions, described within discussions of Service Oriented Architecture (SOA) (O'Brien, Merson, & Bass, 2007), are focused on increased agility and reuse through the composition of discrete, pre-existing, functional software components into new, novel implementations (Gu & Lago, 2007).

As this reuse of pre-existing components grows, so too do the odds that the services that make up the aggregate solution will be developed and supported by third-party organizations (Gu & Lago, 2004). In such a scenario, with multiple components that make up the solution not under the direct control of a single business or organizational entity, the ability to commit to end users on adherence to pre-established service level agreements (SLAs) across all the component parts becomes orders of magnitudes more complex (O'Brien, Merson, & Bass, 2007). This complexity arises as a direct result of both the dynamic/adaptive nature that is inherent in such solutions, as well as the high probability that expectations between the distributed stakeholders for support, infrastructure, and release processes will become more divergent from one another as the solution continues to evolve (Gu & Lago, 2004).

The external ownership of dependent components also places limitations on the ability of the primary service owner to definitively measure the ability for component-based services to meet established levels of service availability and performance for its consumers, as external groups

may limit access to more direct introspection of the physical makeup of dependent services and/or the ability to directly affect or influence change in roadmap or schedule for possible enhancements or modifications (Di Penta, Canfora, Esposito, Mazza, & Bruno, 2007). According to Bertolino, De Angelis, Sabetta & Elbaum (2007), this problem becomes further compounded as the level of functionality grows for the core solution, and (assumedly) the breadth of dependent service components increases. Units of measurement to validate the conformance of all parties also take on different meanings depending on the service being offered and the client expectations for that particular composite service (Bertolino et al., 2007).

Research Problem

While efforts to introduce more measurement tools into the information flow across such systems may seem the immediate solution for better measurement, Grundy, Hosking, Li, & Liu (2006) note that the addition of components intended for improving service measurement may introduce new problems related to latency from the added auditing overhead to the transaction flow. In addition to being potentially expensive from the perspective of processing, Skene, Lammana, and Emmerich (2007) also note in a number of such situations (which they title ‘Inter-service composition’) the ability to validate all plausible scenarios which may validate compliance can be unworkable as the dimension of capabilities grows past some critical point. Additionally, maturity of tools to quantitatively measure the ability for component services to meet established levels of service availability measurements in modern SOA environments tend to be based on long-established measurement criteria, which do not comprehend metrics that reflect the state of systems based on more nascent, dynamic software service models (Bertolino, De Angelis, Sabetta & Elbaum, 2007).

More fundamental than the question of *how* to measure in order to evaluate SLA conformance is the questions of *what* to measure. As noted by Peng, Sun, Rose, & Li (2007), defining a set of Key Performance Indicators (KPI) to provide de-facto indicators on the health, success, and general ability for such a service to meet its contractual obligations can prove extremely challenging. Because development of key performance indicators with a degree of accuracy takes some trial and error to prove and also correlate cause and effect behaviors in a system (Peng et al., 2007), they can present a continually-moving target for SOA-based software systems which are intended to rapidly evolve and reap value rapidly from increased reuse on external services. According to Skene, Skene, Crampton, & Emmerich (2007), many of the more established organizations participating in such composite solutions may themselves perpetuate the application of more traditional models of SLA definition, which are generally based on relatively static/simplistic levels of service availability. Evolving past this model will be difficult, as beyond simple consensus on precise contractual KPIs between two or more organizations, it is also being discovered that new language contexts are required to describe new models of measure so as to be more easily understood by common communities involved in audit or monitoring of SOA solution environments (Di Penta, et al., 2007).

The purpose of this literature review (Cooper, 1998) is to select materials that describe methods and criteria necessary to measure levels of service quality in today's distributed, composite software systems, generally referred by the generic term 'Quality of Service' (QoS) (Di Penta, Canfora, Esposito, Mazza, & Bruno, 2007). More specifically, the inquiry is focused on selection of pertinent literature that addresses three areas: (a) identification of core problem domains for SLA definition for SOA-based software solutions, (b) how the application of measurement

indicators to such dynamic environments can evaluate conformance against SLAs, and (c) where tools can be applied to both validate and monitor expected indicators. The driving assumption for this inquiry is that with improved methods for quantifying specific key aspects of such distributed, composite software solutions (the ‘how’), the ability to more accurately set contractual conformance commitments (the ‘what’) with stakeholders responsible for utilizing the output of the solution will be greatly improved.

Audience and Significance

As stated, the general intent of this inquiry is to provide a basis for evaluation and application of metrics to the complex solution of measuring delivery of SOA-based software solutions, which, as those described by Cardellini et al. (2007), tend to be very dynamic in nature. The goal is not to analyze existing experiences to provide a definitive source of concrete criteria to approach qualification of such solutions, but rather to document and describe the areas of potential complexity when attempting to measure the service delivery expectations from the solution, and add to the available community knowledge. Complexity is examined in relation to three aspects of SLA conformance in SOA solutions, specifically as relates to (a) common areas of challenge in implementation, (b) definition and application of measurement indicators, and (c) the appropriate use of audit systems for runtime evaluation

The inquiry is designed for product management or solution owners of composite software solutions who are associated with the definition and commitment to provide services at levels defined with contractual penalties for non-conformance. As growth in the software industry continues, and more service providers seek to provide competitively-priced services through increased reuse, understanding the potential limitations and implications of SLA criteria and measurement in SOA-based environments will be a key concern to all stakeholders in the

delivery of such solutions (O'Brien, Merson, & Bass, 2007). Of specific mention, key individuals who should find relevance in this review are stakeholders in any aspect of in the lifecycle of such as service (Gu & Lago, 2007), from concept to deployment, and eventual support. Potential areas of responsibility for interested parties would most likely fall into one or more of the following software lifecycle periods:

- Definition of the service -- including the market evaluation activities, discovery of market synergies for the service, and potential opportunities for service reuse through integrating with other providers to develop composite solutions. This is referred to as the 'Market Scan' phase by Gu & Lago (2007), and is the point at which, in addition to seeking out reuse opportunities, stakeholders should apply a critical evaluation against the proposed service to determine if the intended functionality is redundant, or offers no significant improvement or novel capabilities over other established services. This review of market opportunities or gaps should also provide insight into areas of potential complexity introduced by a commercial SOA business strategy the intention of conforming to contractual SLAs with consumers of the intended service.
- Design and implementation planning – this phase requires a deeper evaluation of the potential implementation and integration of proposed components. Items of specific relevance in this phase include both problem areas for SLA, as well as initial planning and analysis of measurement criteria and KPIs -- what Peng et al.(2007) also term the period of "leading indicator discovery". It is during this phase that the determination on intended use of patterns and methods of integration/reuse will be drafted, with specific focus on the non-functional requirements of the system (security, performance, stability, etc.) that are generally directly related to SLA measurement (Kontogiannis et al., 2007).

- Development and validation – where SOA-based software services are initially “brought to life” and initial assumptions of their ability to scale and meet contractual obligations in service SLAs can be confirmed to some minimal extent. Stakeholders at this phase should have specific interest in the incorporation of capabilities to expose needed measurements at runtime, as well as finalizing criteria for tools used in such certification (and eventual audit) of QoS measurements (Kontongiannis et al., 2007). Gu and Lago (2007) have also noted that it is during this phase that service providers have the first critical opportunity to evaluate and validate their service implementation against the QoS metrics and reaffirm the competitive value of the solution against parallel solutions in the marketplace, perhaps through testing with a limited group of early adopters.
- Deployment and support – the final step to initial introduction of a SOA-based software solution is the initial launch into production usage, where the focus topics for this inquiry converge: the point at which SLA definitions should be finalized, and the system put under the auspices of audit systems for the purpose of measuring SLA conformance of services in the runtime environment (Gu & Lago, 2007).

The lifecycle phases described in this inquiry are not a one-time set of events, but represent a rapid, continuously repeating cycle, which allows the solution to stay in synchronization with the needs of a target market -- one of the hallmark characteristics of a SOA solution (Papazoglou & Heuvel, 2007).

The outcome of this inquiry is intended to provide this target audience with a review of the available relevant literature for the topic, some evaluation of the relative merits of the various perspectives, but with an ultimate goal of identifying areas of both strong and weak consensus in the field. This should serve to highlight areas in the evolving development of SLA measurement

models for SOA systems that have yet to be stabilized or resolved in terms of both technical capability and evolving community consensus.

Limitations

Inquiry focus

Skene, Skene, Crampton, & Emmerich (2007) have written extensively on the complexity of various approaches to the measurement of distributed, component-based software systems, as well as the ability to apply those measures to an expected set of mutually-understood & agreed-upon baseline criteria against which to evaluate. One of the potential scenarios where such a complexity arises is when there is an inability (or inequality) of one or more of the parties involved in a specific intra-service partnership to monitor SLA conformance to the same level. This may be perhaps due to relatively simple constraints on these participants, such as one party lacking equivalent capabilities for monitoring or a lack of infrastructure or in-house expertise. As outlined by Skene, Skene, Crampton, & Emmerich (2007), in some cases this may be simply due to the target of measurement being unobservable to one of the dependent parties -- such as in the following diagram (Figure 1), where the party initiating the activity in step 1, does not have direct visibility of when its partner service has initiated step 2 (fulfillment of the service).

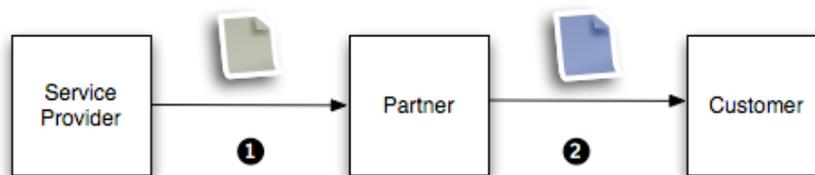


Figure 1. Service provider delivery via Partner entity

Skene, Lamanna, & Emmerich (2004) have also outlined the further complexity of moving past initial consensus between services partners on SLA units of measure to be used, and into the area of recording these to some standardized format that has some critical level of support and understanding across the service lifecycle ecosystem. The introduction by Skene, Lamanna, & Emmerich (2004) of their SLAng specification for documentation of such SLA measures belies the perceived industry infancy with regards to common expectations for such systems. However, as in the SLAng proposal, a minority of the authors included in this inquiry speak in terms of defining and applying absolute measures to the target domain, but rather for the ability for schemas used to define SLAs to provide a starting point, with a focus on extensibility in the definitions. Skene, Lamanna, & Emmerich (2004) make the following comment in their work on SLALang, a schema intended to document SLAs for SOA-type scenarios, “... we are interested in supporting third party monitoring schemes ... how such schemes will work will require careful modeling ...” (Sec. 6, Para. 9). Initial review of the available literature suggests this common perception in the relative immaturity of tools or utilities intended to resolve these problems. In line with this implied consensus, the focus of this review is on identifying core areas of concern related to approaching the specification of SLA conformance measurements for such solutions, as opposed to prescribing measurement methodologies that are expected to be global in their application. Areas that could be insinuated as obvious derivatives from this review, but which are excluded include:

1. Architectural strategies appropriate for distributed component software solutions that may minimize potential for non-conformance against SLAs.
2. Applications of SLAs against systems that have a unique level of QoS compliance, due to inordinately critical levels of service delivery.

3. Assumptions on service level scenarios that do not incorporate the potential factor for dependent components that are under the auspices or control of external organizations.

Of specific exclusion from this review are works that, while describing problem statements related to SOA/component software solutions, address solutions that relate to a very specific or narrow commercial or open-source product or specification as the assumed solution.

Timeframe – Material collected for review includes literature that has been published in the past 4 years, comprising the years 2004 through to present. The driving reason is that search results have shown general literature starting to approximate challenges related to this review topic appear early in this period, with the majority of relevant literature being much more recent (from the year 2007 forward). As noted by Gu & Lago (2007), this shift in software development to distributed component-based solutions has been a relatively recent trend, with ever-increasing focus as technology and standards arrive to the market to overcome limitations of earlier solutions.

Literature Selection Criteria

Preference is given to authors with a professional affiliation to a reputable organization and/or educational institution, as well as those with a reasonable set of known, cited sources within the domain of work (Bell, 2008). Literature selection is made from all available topic sources, including technical conference papers, published books, and papers of professional, non-affiliated industry bodies (such as the Association for Computing Machinery, which figured prominently in preliminary searches).

Audience

The intended audience for this literature review includes professionals who are involved (directly or indirectly) with either the specification or negotiation of service level agreements, or responsible for some aspect of the creation or deployment of services that could be bound by such agreements. This researcher expects that the audience has some implicit understanding of the problem and relevance to their field of work and/or study, most likely gained through practical experience, i.e., initial exposure to the problem domain through industry or organization endeavors.

Writing Plan Preview

The intent of this researcher is to collect information that describes the problem domain and proposed solutions from various perspectives, and derive some commonality in areas for further review or investigation, not to prescribe solutions. The approach to writing the Review of the Literature section is based on a couple of rhetorical patterns, each described by Obenzinger (2005). The first is defined as the ‘Swiss Cheese’ approach, which evaluates known, relevant information and identifies areas (or, more precisely, ‘gaps’) in the available research which the review intends to address. The second pattern is termed ‘Deja-vu all over again’, which outlines a method of research that reviews existing domain information on the target topic, and advocates for a re-validation of the original precepts, or a re-application of the original context against new target criteria. The second pattern seems very appropriate for the current topic, as the initial review of the selected literary sources reveals that while most are aligned on common problem areas, varying degrees of difference exist in their domains of application. Additionally, as a result of rapid evolution of scenarios, capabilities, and standards in the target problem domain, conclusions from as little as a few years ago might now benefit from new perspectives and a larger set of information.

Definitions

The following definitions are deemed to have strong relevance or affinity to the problem domain outlined in the research topic. Most are commonly used industry terms generally abbreviated to represent common patterns or implementation paradigms. The set of Definitions is organized into four general categories of: *Software Architecture Models*, *Application Delivery Models*, *Validation/Monitoring Terms*, and *Support Systems*. These four categories are intended to directly support the major sections of the review of literature, and are listed here in rough approximation to their relevance in the development of the review. The bulk of the terminology is categorized under *Validation/Monitoring Terms*.

Software Architecture Models

Component-based Software – Also referred to as composite software, Gill (2006) has described this category of software as being based on the assembly of multiple discrete components into a defined set of functionality.

Distributed Real-time and Embedded Systems (DRE) – Systems which, as defined by Liu, Raje, & Auguston (2007), are component-based solutions where components are dynamically selected for utilization in a process flow at the point of request, and may comprise one or more child components distributed across multiple discrete subsystems.

Demand-driven software – Software solutions which have the ability to adjust utilization of resources based on prioritization towards functionality which has more requests or perceived demand from service consumers (Atkinson & Griswold, 1996).

Loosely-coupled services – Services that are used to provide integration between two (or more) distinct functional domains which are designed to avoid possible inter-dependence between those services, allowing independent development (Papazoglou & Heuvel, 2007).

Service Component Architecture (SCA) – An emerging standard that defines the composition of multiple services into a functional component, with a goal of shortening the time to develop such a service by simplifying the development complexities (Barber, 2007).

Service-oriented Architecture (SOA) – Muthusamy, Jacobsen, Coulthard, Chan, Waterhouse, & Litani (2007) defined this as primarily an architectural approach for building systems where functional services are composed through reuse of multiple component parts, generally distributed across disparate systems boundaries.

Application Delivery Models

Application Service Provider (ASP) – Describes a model of service delivery whereby applications are delivered on demand, generally across a network, to end customers. These services are usually delivery by the ASP in conjunction with multiple delivery partners, potentially including the organization hosting the application, as well as the vendor providing the delivery transport channel (Skene, Skene, Crampton, & Emmerich, 2007).

Dynamic Service Selection – This describes a model in SOA solutions, whereby the most appropriate service to deliver specific functionality is selected at the time of the request (runtime) (Muthusamy et al., 2007).

Service Provider (SP) – Term to describe an organization that is responsible for providing some functional services (generally fee-based) to individuals or other organizations – examples

may include Application Service Providers (ASP) or Internet Service Providers (ISP) (Skene, Lamanna, & Emmerich, 2004). Also defined by Gu and Lago (2007) as the ultimate owners of a service, with the responsibility for the implementation and maintenance of said service.

Validation/Monitoring Terms

Key performance indicators (KPI) – Peng et al. (2007) define these as indicators, generally represented as a scalar value and associated unit of measurement, that provide a critical set of measurement criteria from which the overall health or level of performance of a given system.

Inter-domain SLA – Service level agreements that exist between two or more distinct controlling organizations, such as in the case of separate internet service providers (Boschi et al., 2005).

Measurement criteria – Definition of a mutually-established set of metrics to measure conformance against a service level agreement (SLA) and the means by which they will be derived (Boschi et al., 2005)

Monitoring framework – A set of software components that provide capabilities for measuring target areas for data metrics that provide some indication of the target health, state, or ability to meet performance of service level goals (Skene, Skene, Crampton, & Emmerich, 2007)

Observer components – Components that exist outside the normal flow of execution within a software domain, yet provide a capability to observe the state of activities within the

flow, generally for the purpose of auditing or monitoring (Zitouni, Seinturier, & Boufaïda, 2007)

Quality of Service (QoS) – This concept, as defined by Sun (2007) relates to the application of objective, repeatable measurements to quantify the overall health of service delivery in terms of characteristics that generally fall under the categories of time (commitment to process within a given period), capacity (ability to support a specified capability level), integrity (the consistent and accurate execution of duties), and fault-tolerance (commitment to maintain a guaranteed level of service irrespective of faults).

Service level Agreement (SLA) – Skene, Lamanna, & Emmerich (2004) have defined this as a mutually-established contractual obligation between two or more parties concerning the quality of services being delivered between the parties, including expected levels of Quality of Service.

Runtime – Defined by Gu and Lago (2007) as the period in which services are implemented and available to be executed against by other consuming services, as well as the point at which services may be subject to SLA conformance monitoring.

Support Systems

Enterprise Service Bus (ESB) – Common functional component found in a majority of SOA solution environments, which provides mediation and translation between multiple distinct services (Papazoglou & Heuvel, 2007).

Message-oriented middleware (MOM) – A solution environment that routes information between distinct components based on the metadata describing the message (or envelope), without

the need to seek reference with the submitter or intended recipient in the message flow
(to XML-based vocabularies to)

Service level monitoring tool (SLM) – Bertolino, De Angelis, Sabetta, & Elbaum (2007) define this as a solution used specifically to audit the runtime conformance of a service-based software solution against specified SLA measures.

Research Parameters

This section provides an overview of how the inquiry is designed. In addition to conveying the search methodology, it is also important to understand how this researcher ascertained the determination of ‘relevance’ to the topic. For that reason, a set of evaluation criteria is outlined to confirm credibility, authority and quality of the chosen sources. The approach used to both manage and document the disparate sources of evaluated literature is detailed. A plan this researcher intends to pursue in the goal of further exploring the topic domain is included.

Search Strategy Report

The following Boolean searches, comprising keywords in contextual sets, are used for literature searches, and relate to the previously-defined terms.

Service-oriented architecture (SOA) *and* service level agreements (SLA)

((Component-based *or* composite) *and* software) *and* service level agreements (SLA)

Quality of Service (QoS) *and* SOA

Search Grading:

Quality of results from searches is appraised on the following criteria:

Grading	Criteria
Excellent	Greater than 80% of initial top 20 results returned were of direct relevance to the topic description
Very Good	Greater than 60% of initial top 20 results were of direct relevance to the search criteria
Good	Greater than 40% of initial top 20 results were of direct

	relevance to the search criteria
Poor	Less than 40% of search results were of relevance to the search criteria.

Table 1. Search Grading

Initial Search Results

Search engine	Terms	# of Results	Result quality
ACM Digital Library	SOA <i>and</i> SLA	96	Excellent
	Composite software <i>and</i> SLA	73	Very Good
	QoS <i>and</i> SOA	90	Very Good
IEEE Computer Society	SOA <i>and</i> SLA	100	Good
	Composite software <i>and</i> SLA	100	Good
	QoS <i>and</i> SOA	100	Good
Google Scholar	SOA <i>and</i> SLA	3,930	Poor
	Composite software <i>and</i> SLA	1,060	Poor
	QoS <i>and</i> SOA	323	Poor

Table 2. Initial Search Results

Search Outcome

The most efficient searches, after continuous tuning on the key strategy vocabulary, have been discovered through the ACM Digital Library, and Google Scholar. Items retrieved from these engine have been the most relevant and, more importantly, more accessible. The ongoing inspection of results returned from the searches with other engines required cross-referencing through various public search engines, such as Google Scholar to gain complete access to the source material. As such, the ACM Library and Google Scholar have become the sole sources of information references being used in this literature search.

Documentation approach

In researching content appropriate for the literature review, a number of methods were used to identify and categorize information with regards to relevance to the research problem, as follows:

1. Key words are documented using mind-mapping software (FreeMind) creating a hub and spoke taxonomical tree from initial concept.
2. Initial results from various literature searches are documented to a software application used for hierarchal note storage (OmniOutliner by the Omni Group). Along with basic metadata (location, title, authors), key words highlighted for the specific work are also recorded.
3. The after an initial review of abstracts, additional impressions are captured, and sources which are available in electronic format are archived to an initial electronic file system.
4. Documents are printed to hard copy and reviewed, after which the initial mind map is updated with key words derived from the review. The notes application is updated with notes/impressions from the review. For electronic copies in Adobe PDF format, the

annotations are added directly to the document to highlight or comment on specific areas of interest within the document.

Literature Evaluation Criteria

In the course of evaluating the literature sources to be incorporated into this review, relevant content was measured on its' appropriateness in regards to the authority of the author(s), the quality of the source information, and the lack of obvious subjective conclusions or biases in the content (Standler, 2004). In addition, the literature resources are continuously held to qualify against the general problem statement of this review to re-confirm relevance to the evolving focus.

Credibility

In reviewing the works, one of the attributes that provides a simple expectation of initial credibility includes the number of authors involved in the publication, as this pre-supposes a certain level of peer review before publication, as opposed to an unchallenged, and potentially more subjective, single voice (Standler, 2004). Through successive searches in the same topic area, a number of names, both lone individuals and sets, repeated on occasion across various literature collaborations, providing further credibility. Search results on individual authors were performed, and highlighted other positive attributes, including:

- Whether the author has participated in publication or collaboration on a significant number of works in a very similar technical domain (Bell, 2008). Common indicators include similar key words in published abstracts, or similar technical forums (industry conferences) for publication of the works.

- The number of works that have been published in association with professional industry associations (such as the IEEE Computer Society) or retail technical publishing organizations (Standler, 2004).
- If publication of the source is sponsored, what is the professional stature of the sponsoring entity (Standler, 2004).

Quality

As a means to evaluate the overall quality of the source, a review of to the general writing style can be very helpful, especially as pertains to obvious misuses of grammar, misspellings, or terms applied in incorrect contexts (Standler, 2004). Further items for consideration can include the consistency used throughout the work with regards to tense, precision in language and avoiding ambiguous or colloquial terminology (Ciolek, 1996). For the works initially selected for this review, an cursory scan was completed for any immediately obvious predilections towards a specific solution or application against a very limited problem domain, as well as review of the author's professional employment or alignment (such as to a reputable university or known corporation) with preference towards those not intimating a preference to a problem solution close to obvious financial interests of their organization (Bell, 2008).

Relevance

As content sources are appraised for references to this literature review, they are evaluated to determine appropriateness and 'best fit' to the overarching research question for the literature review (Bell, 2008). Additionally, as searches yielded more relevant works, the approach outlined in the documentation approach section for creating taxonomical trees (referred to as a 'spider diagram' by Hewitt, 2008) is used to identify clusters of content with highest relevance to the core concept.

Writing Plan

The Review of Literature intends to bring together disparate sources of perspective on the challenges presented in the delivery of SOA-based software solutions as pertains to the implications for setting inter-organization SLAs on such solutions, and the ability to both monitor and validate the conformance at runtime. The plan for developing the content is based on a combination of rhetorical patterns outlined by Obenzinger (2005), which he titles the ‘Swiss Cheese’ and ‘Deja-vu all over again’ forms. The writing goal in ‘Swiss Cheese’ is to review available work and indentify gaps. The writing goal in ‘Deja-vu’ is to revisit or re-apply previous work to new implementation domains. The combination of these patterns is a very appropriate choice for the focus of this literature review, as component-based software is both very broad in its application, as well as rapidly evolving in terms of standards, patterns, and technology to support the domain (Cuadrado et al., 2008). These factors, when coupled with the results of initial literature searches, suggest that there is ample evidence of the need to examine the expected ‘gaps’ in the application of solutions in the available literature, and to revisit models that may have been applied to a narrow set of scenarios, or to domains that are no longer relevant (either due to evolutions in standard or technology). The four facets below are applied to the content development in the Review of the Literature section of this inquiry in the following outline:

1. The application of SLA to software solutions and the traditional delivery of services, including the common patterns of implementation of service level agreements, and the prevalent means to measure and monitor for contractual conformance. As an example, Mitchell & McKee (2005) suggest that more mature industries tend to think in terms of

static measures as a primary focus for SLAs, not the business goals of the service, which tend to be more dynamic in nature.

2. A review of SOA-based, component software solutions, as pertains to SLA, specifically with relation to the benefits and challenges of component reuse. This facet also addresses the details of SOA solutions that are of strongest relevance to QoS monitoring which Skene, Lamanna, & Emmerich (2004) have titled ‘intra-service composition’ -- the use of distributed components, owned by distinct service owners. This section details the monitoring and auditing limitations for SOA-based solutions, as pertains to a number of implementation limitations in the observance or monitoring of processes. A number of common challenges are presented, which include scenarios such as monitor of distributed processes (Skene, Skene, Crampton, & Emmerich, 2007), and the general complexity of managing more flexible, dynamic service delivery models of a SOA software (Mitchell & McKee, 2005).
3. A review of the evolution and promise for improved SLA conformance solutions targeted to SOA software domains. This will include a review for proposals on standards for the definition of SLAs (Masche, Mckee, & Mitchell, 2006), component design models for SOA environments (Papazoglou & Heuvel, 2007), and suggested modifications to solution development lifecycle (Gu & Lago, 2007).
4. Conclusions and recommendations, designed for use by product management or solution owners of composite software solutions, noting gaps in the selected literature related to the focus areas, and suggesting specific solution proposals that could be revisited or re-applied to different problem domains of held to scrutiny with updated target environments.

Annotated Bibliography

The key literature used to support this review are selected based on evaluation against the search criteria already defined, and is intended to provide viewpoints on the subject from one of four facets on the problem of applying SLAs to SOA-based software solutions. These four facets are integrated into the text, and serve to introduce each literature area. In all, 20 distinct literature references have been selected, and vetted against the evaluation criteria detailed previously. All abstracts are retrieved from introductory sections in the individual works.

Facet #1 (six references): *Perspectives on the complexity of the challenges for SOA solutions through review of various models or patterns applied to the service lifecycle, from concept through implementation and support.*

Castagna, G., Gesbert, N., Padovani, L. (2008), A theory of contracts for web services, *Annual Symposium on Principles of Programming Languages archive*, Retrieved April 22, 2008 from ACM Digital Library:
<http://delivery.acm.org/10.1145/1330000/1328471/p261-castagna.pdf?key1=1328471&key2=0516109021&coll=ACM&dl=ACM&CFID=65186722>

Abstract: Contracts are behavioral descriptions of Web services. We devise a theory of contracts that formalizes the compatibility of a client to a service, and the safe replacement of a service with another service. The use of contracts statically ensures the successful completion of every possible interaction between compatible clients and services. The technical device that underlies the theory is the definition of filters, which

are explicit coercions that prevent some possible behaviors of services and, in doing so, they make services compatible with different usage scenarios. We show that filters can be seen as proofs of a sound and complete subcontracting deduction system which simultaneously refines and extends Hennessy's classical axiomatisation of the must testing preorder. The relation is decidable and the decision algorithm is obtained via a cut-elimination process that proves the coherence of subcontracting as a logical system. Despite the richness of the technical development, the resulting approach is based on simple ideas and basic intuitions. Remarkably, its application is mostly independent of the language used to program the services or the clients. We also outline the possible practical impact of such a work and the perspectives of future research it opens.

Results of Evaluation: Work is cited by multiple other literature sources in relevant technical domain, and is hosted on ACM and IEEE catalogs. Authors have multiple relevant works and are affiliated with reputable universities. The authors all have multiple papers published in the past 10 years related to web technologies and service-based solution technologies (XML, web services, etc.).

Gu, Q. & Lago, P. (2007) A stakeholder-driven service life cycle model for SOA, Foundations of Software Engineering archive, Retrieved April 13, 2008 from ACM Digital Library: http://portal.acm.org/ft_gateway.cfm?id=1294930&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: Service-orientation is a relatively new paradigm aiming at developing software systems that are adaptive and dynamic. Service-oriented systems are developed by composing services that are shared across organizations. Because new roles and new

development tasks are introduced in service-oriented development as opposed to traditional software engineering, a new approach to service life cycle management is required. In this paper, based on the observations of the state of the art in the field, we propose a stakeholder-driven service life cycle model for service oriented architecture (SOA). Horizontally, the model shows the activities that associated with the stakeholders in SOA. While vertically, the model shows the interactions and cooperation between the stakeholders. This model facilitates the researchers to gain further insight into service-oriented development and governance.

Results of Evaluation: Work is hosted on ACM Digital Library and sponsored by ACM. Authors have multiple relevant works and are affiliated a reputable universities. The authors all have multiple papers published in the past 10 years related to software development and software architecture. P. Lago has numerous collaboration credits of works published by the IEEE computer society, predominantly related to software architecture.

Kajko-Mattsson, M., Lewis, G.A., Smith, D.B (2007) A Framework for Roles for Development, Evolution and Maintenance of SOA-Based Systems, Retrieved May 13, 2008 from:
<http://delivery.acm.org/10.1145/1280000/1270304/29600007.pdf?key1=1270304&key2=3915280121&coll=Portal&dl=ACM&CFID=27982995>

Abstract: Development, evolution and maintenance of SOA-based systems demands rethinking of the traditional roles for performing these activities. The key objective of this paper is to present preliminary ideas on the roles required for developing, evolving and

maintaining SOA-based systems and to suggest a framework for areas of needed research.

Results of Evaluation: Work is hosted on ACM Digital Library and sponsored by an ACM software interest group. Authors have multiple relevant works and are affiliated a reputable organizations. The authors all have multiple papers published in the past 10 years related to SOA systems, as pertain to software architecture, risk analysis, and engineering practices.

Kontogiannis, K., Lewis, G.A., Smith, D.B., Litoiu, M., Muller, H., Schuster, S., Stroulia, E.

(2007) The Landscape of Service-Oriented Systems: A Research Perspective, Retrieved May 14, 2008 from:

<http://delivery.acm.org/10.1145/1280000/1270298/29600001.pdf?key1=1270298&key2=1915280121&coll=Portal&dl=ACM&CFID=27982995>

Abstract: Service orientation has been touted as one of the most important technologies for designing, implementing and deploying large scale service provision software systems. In this position paper we attempt to investigate an initial classification of challenge areas related to service orientation and service-oriented systems. We start by organizing the research issues related to service orientation in three general categories-- Business, Engineering and Operations, plus a set of cross-cutting concerns across domain. We further propose the notion of Service Strategy as a binding model for these three categories. Finally, concluding this position paper, we outline a set of emerging opportunities to be used for further discussion.

Results of Evaluation: Work is hosted on ACM Digital Library, sponsored by an ACM software interest group, and published by the IEEE computer society. Authors have multiple relevant works and are affiliated with reputable organizations. The authors all have multiple papers published in the past 10 years related to SOA systems, as pertain to software architecture, risk analysis, and engineering practices. Two of the authors are collaborators in another work used as a reference in this review.

Muthusamy, V., Jacobsen, H., Coulthard, P., Chan, A., Waterhouse, J., & Litani, E. (2007) SLA-Driven Process Management in SOA, *Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, Retrieved April 16, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1321243&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: In a Service-Oriented Architecture (SOA), distributed applications are built by orchestrating reusable services using high-level workflows or business processes. The complexity of developing and maintaining these processes is addressed by SOA development cycles that identify the roles of participants at each stage. To assist development, sophisticated tools have been developed, such as the IBM® WebSphere® suite of SOA products. However, the development, administration and maintenance of a business process still requires much manual effort that can be automated. In particular, the non-functional goals of a business process, often expressed as Service level Agreements (SLA) need to be manually considered at each stage of the development process.

Results of Evaluation: Work is hosted on ACM Digital Library and published by ACM.

Authors have multiple relevant works and are affiliated with reputable organizations. The authors all have multiple papers published in the past 10 years related to SOA systems, as pertain to software engineering, SOA messaging patterns, and network quality.

Papazoglou, M. P., Heuvel, W. (2007) Service oriented architectures: approaches, technologies and research issues, The VLDB Journal archive, Volume 16 , Issue 3, July 2007,

Retrieved May 12, 2008 from:

http://delivery.acm.org/10.1145/1270000/1265298/778_2007_Article_44.pdf?key1=1265298&key2=2815280121&coll=Portal&dl=ACM&CFID=27982995

Abstract: Service-oriented architectures (SOA) is an emerging approach that addresses the requirements of loosely coupled, standards-based, and protocol- independent distributed computing. Typically business operations running in an SOA comprise a number of invocations of these different components, often in an event-driven or asynchronous fashion that reflects the underlying business process needs. To build an SOA a highly distributable communications and integration backbone is required. This functionality is provided by the Enterprise Service Bus (ESB) that is an integration platform that utilizes Web services standards to support a wide variety of communications patterns over multiple transport protocols and deliver value-added capabilities for SOA applications. This paper reviews technologies and approaches that unify the principles and concepts of SOA with those of event-based programming. The paper also focuses on the ESB and describes a range of functions that are designed to offer a manageable, standards-based SOA backbone that extends middleware

functionality throughout by connecting heterogeneous components and systems and offers integration services. Finally, the paper proposes an approach to extend the conventional SOA to cater for essential ESB requirements that include capabilities such as service orchestration, "intelligent" routing, provisioning, integrity and security of message as well as service management. The layers in this extended SOA, in short xSOA, are used to classify research issues and current research activities.

Results of Evaluation: Work is hosted on ACM Digital Library and published by Springer-Verlag, a global academic publisher. Authors have multiple relevant works and are affiliated with a reputable university. The authors all have multiple papers published in the past 10 years related to SOA-related technologies, including web services security, support and quality.

Facet #2 (four references): *Outlines of proposed models for improved definition of SLAs to account for the nature of SOA solution environments.*

Cardellini, V., Casalicchio, E., Grassi, V., & Lo Presti, F. (2007) Efficient Provisioning of Service level Agreements for Service Oriented Applications, *2nd international workshop on Service oriented software engineering*, Retrieved April 17, 2008 from ACM Digital Library:
http://portal.acm.org/ft_gateway.cfm?id=1294936&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: In this paper, we consider a provider that offers an application implemented as a composite service to several users with (possibly) different Quality of Service (QoS)

requirements. To this end, the provider negotiates with both the clients and the service providers Service level Agreements (SLAs), which define the respective QoS-related obligations along with the interval of time over which such obligations are to be met.

Results of Evaluation: Work is hosted on ACM Digital Library and published by ACM. Authors have multiple relevant works and are affiliated with a reputable university. The authors all have multiple papers published in the past 10 years related to service-oriented systems or predecessor web service technologies and architecture patterns.

Masche, P., Mckee, P., & Mitchell, B. (2006) The increasing role of service level agreements in B2B systems, Retrieved May 14, 2008 from:

http://www.nextgrid.org/download/publications/The_increasing_role_of_service_level_agreements_in_B2B_systems.pdf

Abstract: Service orientated architectures (SOA) will support a dynamic market in commodity services and enable business to drive down costs and respond faster and flexibly to changing markets. Virtualization delivers similar benefits for the management of resources. If viable business models for the combination of these two technologies can be found, a true commodity grid can become reality. An essential principle of any viable business model will be to secure the flexibility for service providers to manage and provision services. The provider consumer relationship is encapsulated within a service level agreement (SLA). We propose that this SLA contains terms that only relate to business level objectives (BLO). Deployment and management details of a service are hidden by virtualization in the provider's domain and therefore should not be expressed in the SLA. The SLA will become key to build confidence in the business relationship

between provider and consumer and a differentiating factor between providers in a market place.

Results of Evaluation: Work is cited by several other literature sources, and authors have a number of relevant works and are affiliated with a reputable organization (British Telecom). The authors all have multiple papers published in the past 10 years related to related to service delivery quality over networks.

O'Brien, L., Merson, P., & Bass, L. (2007) Quality Attributes for Service-Oriented Architectures, *International Conference on Software Engineering archive*, Retrieved April 11, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1270300&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: The SOA approach is a very popular choice today for the implementation of distributed systems. The use of SOA or more specifically the Web services technology is an important architecture decision. An architect should understand how different quality attributes for a system are impacted by that decision. While there are significant benefits with respect to interoperability and modifiability, other qualities such as performance, security and testability are concerns. This paper discusses how the different quality attributes of a system can be positively or negatively affected by the use of such technology. It describes the factors related to each attribute, as well as possible tradeoffs and existing efforts to achieve that quality. The paper also discusses open issues in service level agreements that are used to contract the level of service quality between service providers and users.

Results of Evaluation: Work is hosted on ACM Digital Library and published by IEEE Computer Society. One of the authors (Len Bass) has an extensive history of publishing multiple relevant works, and all are affiliated with reputable organizations/universities (University of Limerick, Ireland, & Software Engineering Institute, USA).

Skene, J., Lamanna, D., & Emmerich, W. (2004) Precise Service level Agreements, *Proceedings of the 26th International Conference on Software Engineering*, Retrieved April 14, 2008 from ACM Digital Library:
http://portal.acm.org/ft_gateway.cfm?id=999422&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: SLAng is an XML language for defining service level agreements, the part of a contract between the client and provider of an Internet service that describes the quality attributes that the service is required to possess. We define the semantics of SLAng precisely by modeling the syntax of the language in UML, then relating the language model to a model that describes the structure and behavior of services. The presence of SLAng elements imposes behavioral constraints on service elements, and the precise definition of these constraints using OCL constitutes the semantic description of the language. We use the semantics to define a notion of SLA compatibility, and an extension to UML that enables the modeling of service situations as a precursor to analysis, implementation and provisioning activities.

Results of Evaluation: Work is hosted on ACM Digital Library and published by IEEE Computer Society. One of the authors has an extensive history of publishing multiple

relevant works, and all are affiliated with a reputable university (University College London). This work is also cited by a number of other literature sources.

Facet #3 (six references): *Discussions on the application of monitoring criteria and new/novel approaches to the application of measures to quantitatively determine run-time conformance for such SOA-based solutions against SLAs.*

Bertolino, A., De Angelis, G., Sabetta, A., & Elbaum, S. (2007) *Scaling up SLA Monitoring in Pervasive Environments, International workshop on Engineering of software services for pervasive environments*. Retrieved April 15, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1294914&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: Service level agreements (SLA) are contracts defining the obligations and rights between the provider of a web service and a client with respect to the service's quality. An essential component of SLA management is the run-time checking of relevant quality parameters. SLA checking must be precise and continuous to timely detect any possible SLA violations. Available techniques and mechanisms to check SLAs, however, are unlikely to scale to more complex families of SLAs tailored to fit diverse and very large groups of clients whose devices have limited storage and bandwidth capabilities. In this paper we focus on the elaboration of this challenge and on identifying opportunities that arise from the inherent diversity present in scenarios rendering web services in pervasive environments.

Results of Evaluation: Work is hosted on ACM Digital Library and published by ACM. Collectively, the authors have an extensive history of publishing multiple relevant works, and all are affiliated with reputable organizations/universities. The article is relevant to the focus of this literature topic, specifically as related to monitoring and testing of service-based software.

Liu, S., Bryant, B., Gray, J., Raje, R., Olson, A., Auguston, M. (2005) Two-level assurance of QoS requirements for distributed real-time and embedded systems, *Symposium on Applied Computing archive*, Retrieved April 23, 2008 from ACM Digital Library: <http://delivery.acm.org/10.1145/1070000/1066884/p903-liu.pdf?key1=1066884&key2=5987109021&coll=ACM&dl=ACM&CFID=65186722>

Abstract: Assuring quality of service (QoS) requirements is critical when assembling a distributed real-time and embedded (DRE) system from a repository of existing software and hardware components. This paper presents a two-level approach for assuring satisfaction of QoS requirements in the context of a reduced design space for DRE systems. Techniques from artificial intelligence and statistics are used to fulfill these collective objectives at system assembly time. The result not only lessens the overhead of validation of QoS requirements at run-time, but also reduces the development and integration cost of DRE systems.

Results of Evaluation: Work is hosted on ACM Digital Library and published by ACM. The authors have an extensive history of publishing multiple relevant works, mostly related to modeling of complex software systems for runtime performance estimation, and validation. All authors are affiliated with reputable organizations/universities

(University of Alabama at Birmingham, USA, Indiana Purdue University, USA, and the Naval Postgraduate School in California, USA).

Mitchell, B. & Mckee, P. (2005) SLAs A Key Commercial Tool, Retrieved May 15, 2008 from:

<http://www.nextgrid.org/download/publications/echallenges.pdf>

Abstract: SLAs are likely to become a key tool for the service provider in offering commercial services in Service Oriented Architectures. A well chosen SLA helps in the establishment of credibility for the service provider; it is a means to attract customers in a competitive environment and establishes the customer-provider business relationship. It may help to retain customers and will be used as a mechanism for service differentiation. SLAs have many complex issues associated with them and are difficult to use and understand, particularly in a value chain, so a common approach to the representation of SLAs is desirable. Most of the current SLA representations focus almost exclusively on the functional aspects of the service being offered such as performance levels and ignore the non-functional factors that have a considerable influence on the successful establishment of a business relationship. Automation of this process is further inhibited by the confidentiality requirements that often surround commercial SLAs.

Results of Evaluation: Work is cited by at least two other relevant works related to SLAs and their relation to business strategy. The authors have a modest history of publishing relevant works, mostly related to analysis of software systems comprising web services. Both authors are affiliated with a reputable organization (British Telecom).

Molina-Jimenez, C., Shrivastava, S., Crowcroft, J., & Gevros, P. (2004) On the Monitoring of

Contractual Service level Agreements, Retrieved May 14, 2008 from:

<http://ieeexplore.ieee.org/iel5/9222/29240/01319502.pdf?tp=&isnumber=&arnumber=1319502>

Abstract: The goal of monitoring contractual Service level Agreements (SLAs) is to measure the performance of a service, to evaluate whether its provider complies with the level of Quality of the Service (QoS) that the consumer expects. The aim of this paper is to bring to the system designer's attention the fundamental issues that monitoring of contractual SLAs involves: SLA specification, separation of the computation and communication infrastructure of the provider, service points of presence, metric collection approaches, measurement service and evaluation and violation detection service. The paper develops an architecture and give reasons why currently it is practicable to offer guaranteed QoS only to consumers sharing Internet Service Providers (ISPs) directly with the service provider.

Results of Evaluation: Work is published by IEEE Computer Society, and is cited by a number of other relevant works. The authors have an extensive history of publishing multiple relevant works related to contractual agreements on service solutions. All authors are affiliated with reputable organizations/universities.

Morgan, G., Parkin, S., Molina-Jimenez, C., Skene, J. (2007) Monitoring Middleware for Service Level Agreements in Heterogeneous Environments, Retrieved May 14, 2008 from: <http://www.cs.newcastle.ac.uk/publications/inproceedings/papers/898.pdf>

Abstract: Monitoring of Service Level Agreements (SLAs) is required to determine if the Quality of Service (QoS) provided by a service provider satisfies the expectations of a service consumer. Although tools exist that can generate the software required to evaluate

SLAs from the SLA specifications themselves, the code required to gather metric data is still predominantly coded by hand: a time consuming task. In this paper we describe an SLA monitoring implementation that can generate metric data gathering software directly from machine readable SLAs. Assuming that an organization specializing in SLA monitoring and evaluation may not wish to be tied to any one particular middleware platform and/or SLA language, we aim to provide generic monitoring services that may be suitable for use in heterogeneous environments. We demonstrate the flexibility of our approach by providing monitoring solutions for observed systems implemented using Web Services and Enterprise Java Bean (EJB) middleware using a third party SLA language.

Results of Evaluation: Work is published by IEEE Computer Society, and is cited by a number of other relevant works. The authors have an extensive history of publishing multiple relevant works related to contractual agreements on service solutions. All authors are affiliated with reputable organizations/universities. Two of the authors are collaborators on other works being referenced this literature review.

Skene, J., Skene, A., Crampton, J., & Emmerich, W. (2007) The monitorability of service level agreements for application-service provision, *Run-time performance and resource-awareness -- Proceedings of the 6th international workshop on Software and performance*, Retrieved April 21, 2008 from ACM Digital Library:

<http://delivery.acm.org/10.1145/1220000/1216997/p3-skene.pdf?key1=1216997&key2=4361109021&coll=ACM&dl=ACM&CFID=65186722>

Abstract: Service level Agreements (SLAs) mitigate the risks of a service-provision scenario by associating financial penalties with aberrant service behavior. SLAs are useless if their provisions can be unilaterally ignored by a party without incurring any liability. To avoid this, it is necessary to ensure that each party's conformance to its obligations can be monitored by the other parties. We introduce a technique for analyzing systems of SLAs to determine the degree of monitorability possible. We apply this technique to identify the most monitorable system of SLAs including timeliness constraints for a three-role Application-Service Provision (ASP) scenario. The system contains SLAs that are at best mutually monitorable, implying the requirement for reconciliation of monitoring data between the parties, and hence the need to constrain the parties to report honestly while accommodating unavoidable measurement error. We describe the design of a fair constraint on the precision and accuracy of reported measurements, and its approximate monitorability using a statistical hypothesis test.

Results of Evaluation: Work is hosted on ACM Digital Library and published by ACM. The authors collectively have an extensive history of publishing multiple relevant works, and all are affiliated with reputable universities. This work is also cited by at least eight other literature sources, the majority of which follow a common theme of validation and monitoring of software service environments.

Facet #4 (four references): *Proposed solutions for approaching the validation or certification of such solutions to meet SLAs prior to the solution being launched into full production (live) usage.*

Di Penta, M., Canfora, G., Esposito, G., Mazza, V., & Bruno, M. (2007) Search-based Testing of Service Level Agreements, *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, Retrieved April 16, 2008 from ACM Digital Library:
http://portal.acm.org/ft_gateway.cfm?id=1277174&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: The diffusion of service oriented architectures introduces the need for novel testing approaches. On the one side, testing must be able to identify failures in the functionality provided by service. On the other side, it needs to identify cases in which the Service Level Agreement (SLA) negotiated between the service provider and the service consumer is not met. This would allow the developer to improve service performances, where needed, and the provider to avoid promising Quality of Service (QoS) levels that cannot be guaranteed. This paper proposes the use of Genetic Algorithms to generate inputs and configurations for service-oriented systems that cause SLA violations. The approach has been implemented in a tool and applied to an audio processing workflow and to a service for chart generation. In both cases, the approach was able to produce test data able to violate some QoS constraints.

Results of Evaluation: Work is hosted on ACM Digital Library and published by ACM. The authors have history of publishing multiple relevant works, specifically related to software development, integration of disparate system through web services, and other common SOA system technologies or precursor technologies. All authors are affiliated with a reputable university (University of Sannio, Italy).

Gill, N. (2007) Importance of Software Component Characterization For Better Software

Reusability, *ACM SIGSOFT Software Engineering Notes, Volume 31, Issue 1*, Retrieved April 16, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1108771&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: Component-based software development (CBSD) is the process of assembling existing software components in an application such that they interact to satisfy a predefined functionality. This approach can potentially be used to reduce software development costs, assemble systems rapidly, and reduce the maintenance overhead. One of the key challenges faced by software developers is to make component-based development (CBD) an efficient and effective approach. Since components are to be reused across various products and product-families, components must be characterized and tested properly. The present paper is a survey paper and firstly, it discusses CBD and related issues that help improving software reuse. Testing of third party components is a very difficult task in the absence a properly characterized software component. Besides improving software reusability, component characterization also provides better understanding of architecture, better retrieval, better usage and better cataloguing. This paper mainly discusses the essence of proper component characterization that ultimately helps the developers in software reuse, which is highly desirable in component-based software development. Further, paper also discusses other benefits of component characterization that are most essential in component-based development.

Results of Evaluation: Work is hosted on ACM Digital Library and published by ACM. The author has multiple published relevant works within the past 5 years, specifically related to software quality and measurement for component-based software solutions.

Grundy, J., Hosking, J., Li, L., & Liu, N. (2006) *Performance Engineering of Service Compositions, Proceedings of the 2006 international workshop on Service-oriented software engineering*. Retrieved April 16, 2008 from ACM Digital Library:
http://portal.acm.org/ft_gateway.cfm?id=1138493&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Abstract: While a service-oriented approach to software engineering has become popular in recent times, the actual performance of systems composed from many distributed parts is still largely unpredictable. We describe our recent research applying performance test-bed generation techniques to service-oriented architectural models as an advance on the state of the art in performance engineering of service-oriented software. We outline our related research on tools for business process composition, performance engineering and dynamic system architectures.

Results of Evaluation: Work is hosted on ACM Digital Library, published by ACM, and sponsored by a Software interest group. The authors have history of publishing multiple relevant works, specifically related to software development and modeling of performance test cases. All authors are affiliated with the University of Auckland, New Zealand.

Zitouni, A., Seinturier, L., & Boufaïda, M. (2007) Contract-based approach to analyze software components, Retrieved April 29, 2008 from:

<http://hal.inria.fr/docs/00/27/15/33/PDF/zitbousei2-uml%5B1%5D.pdf>

Abstract: Component-based software development focuses on building large software systems by integrating existing software components to reduce cost, risk and time.

However, behavioral and compositional conflicts among components constitute a crucial barrier to successful software composition. In this paper, we present a contract-based approach to analyze and model the properties of components and their composition in order to detect and correct composition errors. With this approach we characterize the structural, interface and behavioral aspects, and a specific form of evolution of these components.

Results of Evaluation: This work is published by IEEE Computer Library. The authors have history of related publications and collaborations, all of which are closely related to the topic of dynamic software and service models and approaches for validation. Lionel Seinturier, in particular, has credits on dozens of papers related to topics of software development, distributed component systems, and flexible software environments. All authors are affiliated with reputable universities (University of Lille, France, and the Mentouri University of Constantine, Algeria).

Review of the Literature

The Review of Literature brings together various perspectives related to the challenges of specifying contractual service level agreements (SLAs) between one or more participant organizations involved in a distributed, component software solution. The goal is to ensure runtime conformance through prior validation and audit at point of usage. The Review focuses on four facets of the research problem, beginning with the application of SLAs to software solutions and the traditional delivery of services, common patterns of implementation of service level agreements, and the prevalent means to measure or monitor conformance. This is followed by a review of SOA-based, component software solutions, specifically where they pertain to SLA and the relation to the benefits and challenges of inter-organization component reuse. Next, a review of the evolution and promise for improved SLA conformance solutions targeted to SOA software domains is provided. This includes a review for proposals on standards for the definition of SLAs, component design models for SOA environments, and suggested modifications to solution development lifecycle.

Part 1: Traditional application of SLA to service delivery, including the common patterns of implementation of service level agreements, and the prevalent means to measure or monitor conformance.

Mitchell & McKee (2005) report that certain industries have a long history of using SLAs to contractually define expectations for delivery of services, and more specifically in the delivery of electronic services. They note this is most evident in the telecommunication sector, and more recently in application data provider networks used in industries such as banking, retail, and

aviation. Muthusamy et al. (2007) comment that for these more established industries, ensuring conformance with SLAs generally involved the application of more resources than were needed, supported by manually tuning the solution to meet relatively static expectations of expected runtime constraints. Additionally, the responses to QoS violations of production services were generally also met through relatively manual means, such as the re-engineering of the services that did not conform to the obligations under the SLA, or addition of more computational resources to the processing of the service, all of which involved significant resource allocation (Muthusamy et al., 2007).

Universal Service Level Agreement Precepts

The following section describes two commonly agreed upon precepts that Service Level Agreements are intended to address and support.

Precept #1: Precise description of expectations. One of the most fundamental expectations for the application of an SLA to the delivery of services is to precisely define the predominantly non-functional characteristics in the delivery of the target service, which generally fall under the categories of availability, timeliness, and consistency (Cardellini et al., 2007). This precision is also needed to guarantee the ‘immutability’ of an SLA and the ability to concisely capture the specific intent of QoS, and not be subject to successful challenge to the relevance/correctness of the content defined therein – what Skene, Skene, Crampton, & Emmerich (2007) refer to as the ‘protectability’ of the SLA. As the monitoring and enforcement of SLAs are generally linked to business goals for both consumers and providers of services (O’Brien, Merson, & Bass, 2007), the contractual obligations have a real potential impact to the financial bottom line of all parties involved in runtime transactions.

Precept #2: Third party oversight. Because SLA definitions also serve multiple stakeholders throughout the design, development and validation phases in the lifecycle of a service and not just strictly at runtime (Muthusamy et al., 2007), it is exceedingly important that these expectations are not subject to significant interpretation on their evolution towards, or implementation in, a production environment. This need for clear, consistent definition of expectations also extends beyond stakeholders directly involved in the implementation or consumption of the service, but can involve the incorporation of external parties, such as third-party monitoring service organizations or auditors, who are involved to provide objective or neutral enforcement for the mutual assuredness of SLA conformance (Molina-Jimenez, Shrivastava, Crowcroft, & Gevros, 2004).

Mitchell & McKee (2005) assert that it is a basic expectation that in order for SLAs to function as motivation between parties, they must impose implications for non-conformance that have real meaning to the parties involved. This means that penalties should have gravity to them in order to provide a real incentive towards correct performance.

Part 2: Introduction of SOA-based software services, and implications to SLA, specifically with relation to the benefits and challenges of component reuse.

With the evolution of new patterns of software development, the focus for a majority of service development organizations has turned away from monolithic, self-contained solutions and towards opportunities for functionality reuse from existing assets, quite often distributed in nature (Papazoglou & Heuvel, 2007). As noted by O'Brien, Merson, & Bass (2007), this new

distributed architecture requires a thoughtful examination with respect to the new realities of QoS attributes for such solutions and the delivery of SLA-bound services.

New Models Of Service Delivery

O'Brien, Merson, & Bass (2007) note that service-based solutions have opened the potential for organizations to rapidly respond to demands for new capabilities and levels of functionality, which provides the level of agility needed to respond to the increasing rate of change in commercial opportunities. When coupled with the evolution and prolific adoption of service communication protocols and vocabularies, this fosters opportunities for organizations to develop functional capabilities that are based on highly distributed, highly dynamic dependent components (Papazoglou & Heuvel, 2007). According to Cardellini, Casalicchio, Grassi, & Presti (2007), it is this dynamic quality of service selection that has also become a multi-faceted concern: modern service consumers are now able to dynamically select appropriate services at the point of request, either based on availability or cost, requiring the service provider to be cognizant of performance against QoS expectations. Whereas the majority of traditional service consumers look for reliable, consistent service delivery, the modern service consumer may prefer to subscribe to a lower-cost service, which delivers equivalent functionality to another, but with different quality of service related to properties like availability or response time (Cardellini, et al., 2007). It is also becoming more common for distinct consumers of a specific service to subscribe based on unique QoS requirements which, as noted by Bertolino et al. (2007), puts numerous dependencies on the service provider to manage SLAs, not as static service guarantees prescribed to all consumers of a service, but as multi-dimensional objects with an indeterminate number of potential forms that may not be anticipated at time of design.

Of equal concern, is the growing focus on reuse, which follows a pattern of building composite solutions from distributed components, pre-established components, or what Di Penta et al. (2007) refer to more accurately as the new paradigm of software being ‘used instead of being owned’ (p.1090). Facilitated by the growth of the new interoperability capabilities, Skene, Lamanna, & Emmerich (2004) note that this growing preference by service development organizations for pre-existing components increases the likelihood of external or autonomous ownership of dependent components, as well as the reality of extension into the domain of business-critical applications, not strictly low cost/low risk implementation environments. With the increasing probability that the services will not be located within the contingent environment of the organization that is developing the service, the complexities for monitoring and affecting change, such as the pattern of reimplementation that is common for legacy services, are many times more complex (Muthusamy et al., 2007).

The Failure to Apply Legacy SLA Models to SOA

New paradigms in service delivery have arrived rapidly, and the application of existing models to this nascent ecosystem does not work. Bertolino et al. (2007) describe, rather bluntly, that in their estimation the overlay of traditional SLA tools to such composite service systems, fails “naively” (p.65). Whereas SLA traditionally was applied to inter-organization service consumption, the growth in service delivery to end consumers has imposed restrictions in the ability to apply mutual monitoring of services (Skene, Skene, Crampton, & Emmerich, 2007). Traditional measurement for conformance generally involved a smaller set of actors, and implementation of monitoring impacted a smaller domain (see *Figure 2*).

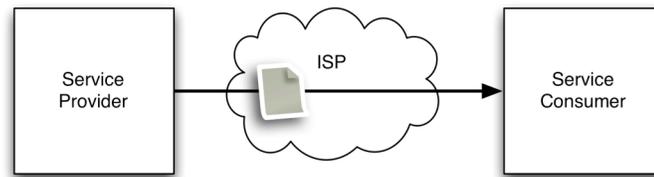


Figure 2. Traditional service delivery model, single communication provider (ISP)

Proliferation of services to models of direct delivery to end users complicates this earlier model, due to numerous aspects, including the potential negative response from private citizens on imposing service measurement instrumentation on their end systems (privacy concerns) (Morgan, Parkin, Molina-Jimenez, Skene, 2007). Additionally, migration to business models that sell directly to end users may introduce a new set of unexpected partners, specifically Internet Service Providers (ISPs) that maintain the medium across which the services are delivered. Whereas a last-generation model for service delivery, assumedly between organizations, could generally rely on a very small, possibly one, connectivity provider, Molina-Jimenez, Shrivastava, Crowcroft, & Gevros (2004) note that common contemporary delivery of services to end users can now easily span dozens of different participants in the communication channel (see *Figure 3*).

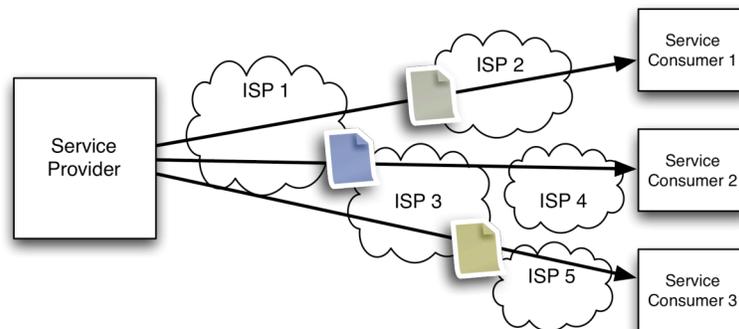


Figure 3. Modern service delivery model, multiple communication providers (ISPs)

Potential Jeopardy in Pre-Release Validation of SOA Solutions

There are some basic constraints of SOA environments that make validation during the development phase difficult, which can include: a) lack of access to the full scope of detail of a dependent service due to external ownership, b) the ability to qualify operation scenarios which may only be known at runtime, c) the potential for dependent services to evolve and/or change on their own schedule, and d) the potential to incur additional costs by consuming external services for pre-release validation (simulating runtime experiences) (Di Penta, Canfora, Esposito, Mazza, & Bruno, 2007). Also, due to the likely scenario that composite software solutions maintain dependencies with multiple distinct child components derived from other service owners/developers, it can be exceedingly difficult for an implementation team to accurately qualify the complete runtime QoS characteristics outside of a production deployment environment (Sun, 2004).

As noted previously, the nature of SOA-based composite services implies fairly rapid evolution of service capabilities, which can result in changes to previously-established expectations for functionality and expected levels of service quality, forcing additional validation cycles to reconfirm SLA indicator baselines (Di Penta et al., 2007). When such a scenario is coupled with the inability of a service owner to control the timing of the change of a dependent service, forcing a re-validation activity at some critical time (for example on the eve of a feature launch) could spell disaster for an organization.

Next-Generation SLA Precepts

With the conclusion that legacy paradigms for the application of SLA to the realities of QoS expectations for next-generation composite software solutions need to evolve, the following section describes four additional precepts that Service Level Agreements must now address and support.

Precept #3: Audit of solutions based on distributed components. As in traditional systems, there is a need in next-generation systems to ensure conformance through audit. However, as illustrated in Figure 1, accurately monitoring critical activities in a distributed, composite solution can be hindered by simple inability to access or have direct line of sight to all components involved (Skene, Skene, Crampton, & Emmerich, 2007).

Precept #4: Multiplicity of service delivery models. The ability to set clear expectations on a complete SLA for a composite solution requires comprehension of what Sun (2004) terms ‘compositional reasoning’ – that is the need to develop quality of service expectations for the solution, keeping in mind both the unique behavior and QoS guarantees made by the distinct component parts within the solution, as well as the patterns of compositions. Skene, Lamanna, & Emmerich (2004) break this into two distinct models of composition of services against SLAs, namely the ‘inter-service’ and ‘intra-service’ models; the inter-service model describes scenarios where service guarantees for one autonomic service are evaluated against the SLA expectations by a consumer, and the intra-service model, where QoS expectations for the solution are a factor of the sum of the SLA guarantees of the component parts. While both models expose added complexity, it is the intra-service model which would seem to imply the added dimension in complexity and a fundamental change from single dependency of solution developer, to the various component developers (Sun, 2004).

Precept #5: Consideration for dynamic behavior. There is also a need for modern SLAs to comprehend the dynamic nature of services, and the ability to make determinations in method and quality of delivery at runtime or at the point of request to optimize resources as relates to cost (Bertolino et al., 2007). The integration of SLA audit tools directly into the service execution layer provides for immediate measurement and associate response on violations or dynamic re-allocation of compute resources, or what Muthusamy et al. (2007) call dynamic resource allocation. Additionally, consumers of the service have options to perform dynamic service selection (Muthusamy et al., 2007) perhaps based on previous experiences, or feedback from a community of peer consumers, which provides further incentive to the service provider to ensure conformance.

Precept #6: The tendency towards service multi-tenancy. The capability for multi-tenancy (referring to service delivery to multiple distinct sets of users) fosters the creation of services that can support differentiated QoS levels based on per-consumer expectations (Cardellini et al., 2007). This model of service delivery, as noted by Skene, Lamanna, & Emmerich (2004), creates a need for the SLA definition to comprehend contracts as the interaction between a service and a specific consumer, not strictly as a static measurement of the expected non-functional qualities of the service (availability, reliability, etc.). Differentiated services levels between consumers also have the ancillary effect of creating new competitive models, where services can be dynamically selected at runtime, which, as Di Penta et al. (2007) mention, must be comprehended for the assumed complications for accurate prediction of runtime behaviors of a planned service against QoS expectations.

This support for multiple users and unique QoS expectations has the secondary effect of motivating service providers to implement adaptive delivery capabilities that can assign compute resources dynamically at runtime to maximize cost efficiency (Muthusamy et al, 2007). As noted by Mitchell & Mckee (2005), this push towards adaptive hosting environments for the services being delivered requires a much finer grained perspective on the underlying compute hardware, which presents challenges for deeper integration between the services, the monitoring components, and the resource control points. Additionally, as the flexibility of the system evolves, so too do the potentially unanticipated side effects, such as the ability for monitoring systems to support the increased requirement for audit on smaller intervals which creates further computation performance impact (Cardellini et al., 2007).

Part 3: Towards convergence of SLA tools/standards for SOA solutions, including examination of the promise for improved SLA conformance solutions targeted to SOA software domains.

Monitoring Patterns/Systems

Betrolini, De Angelis, Sabetta, and Elbaum (2007) describe the need for SLA audit mechanisms that apply a more adaptive approach than traditional solutions and which can not only observe, but direct the allocation or de-allocation of computational resources as a reaction to service demand by consumers. In order to achieve this, it is noted throughout the selected literature that SLA definitions are put into machine-readable format, to facilitate what Muthusamy et al. (2007) describe as “runtime adaption ... without human intervention” (p.2), which is deemed critical for such a solution to be effective. This ability for self-adaptation from automated means is seen as

the best opportunity to meet the challenge of differentiated service monitoring, with the specific goal of optimizing resources to meet the best balance of cost against service SLA obligations (Bertolino et al., 2007).

The nearly ubiquitous migration of composite software solutions data interchange methods that use XML-based vocabularies and a common set of transport protocols (JMS & HTTP, most notably) has simplified SLA measurement to a certain extent. Of specific mention are the emergence of the enterprise service bus (ESB) and message-oriented middleware systems (Morgan et al., 2007). The introduction of the enterprise service bus, while predominantly targeting the mediation and translation between distinct message vocabularies, has also provided a centralized location from which to effectively measure service delivery (Muthusamy et al., 2007).

In their work on SLAng, Skene, Lamanna, & Emmerich (2004) note that in certain cases, the specifics on what is being monitored may present challenges to regional privacy regulations, or could even expose proprietary intellectual property of one of the participants in the service transaction, potentially necessitating the introduction of trusted third-party organizations into the audit process. Molina-Jimenez et al., (2004) note that third-party auditors can provide needed competencies for monitoring, not only for privacy concerns but also due to the growing inequality between common service providers and the consumers for that service with regards to access to monitoring infrastructure or technical expertise. Molina-Jimenez et al. (2004) also characterize this need for a third party as twofold in terms of function – the measurement service, which collects data on the service delivery with a mandate to balance inspection with the

potential impact to performance, and the evaluation service, which is responsible for auditing the collected data against stipulated SLAs and levying judgment and notifying involved parties on possible violations.

SLA Specifications

A number of means to detail SLA contractual specifications have arisen over the past years as a response to the shortcomings of legacy models to accurately reflect the intent of SLAs and ensure their mutual understanding between parties involved in a composite solution. Also, as mentioned in a number of pieces of the literature selected for this review, there is a strong need for SLAs to be captured into machine-readable formats, to provide better integration. The most prevalent specifications for meeting some of the needs outlined in this review are listed below, with brief description of the core qualities and relative merits.

WS-Agreement. Final specification for WS-Agreement (<http://www.ogf.org/documents/GFD.107.pdf>) was published in March of 2007 by the Open Grid Forum (<http://www.ogf.org/>), the result of collaboration between multiple commercial partners including IBM, NEC, HP, and others. The introductory description accurately mimics one of the common problem domains for dynamic, composite software solutions related to SLAs, that “actual resource usage cannot simply be advertised as an invariant property of a service”. This specification has been cited by numerous authors in the literature selected for this review as complementary or foundational to their work, including Bertolino, De Angelis, Sabetta, & Elbaum (2007), Mitchell & McKee (2005), and Di Penta, Canfora, Esposito, Mazza, & Bruno (2007).

WSLA. Developed by IBM research (<http://www.research.ibm.com/wsla/>), as a specification, the Web Service Level Agreements Project (WSLA) was initiated in 2001, and updated in January of 2003. As stated in the abstract of the specification, “However, a WSLA only covers the agreed common view of a service between the parties involved. To actually act as a participant in a WSLA, parties have various degrees of freedom to define an implementation policy for a service and its supervision. Typically, the obligations of a WSLA must be translated into system-level configuration information, which can be proprietary to each party involved.”

Molina-Jimenez et al. (2004) mention that the WSLA specification does not provide guidance to a common scenario in consumer SOA solutions, which is how to address the issue of delivery across multiple ISP communication channel that form the path between service provider and consumer. Skene, Lamanna, & Emmerich (2004) comment that the syntactic separation in WSLA of monitoring specifics away from the contractual obligations allows more flexible distribution of the SLA to multiple parties, while at the same time avoiding potential exposure of sensitive information.

SLAng. This specification was originally proposed by Skene, Lamanna, & Emmerich in a 2003 paper, with focus on framing negotiations for non-functional quality of service properties in an unambiguous manner. Concepts include the application of adaptive periods to which agreement measures apply, including complex overlapping periods of SLA applicability, such as recurrence periods, and periods based on offset from another point in time, as well as stipulation of obligations for reporting of collected measurements. The authors make the comparison that

SLAng defines obligations in terms of behavior expected between the service provider and a specific consumer of that service, as opposed to more static system measurements which they suggest open to a greater degree of interpretation. The modeling of behaviors uses a fairly common means of expressing relationships (UML) and uses an XML vocabulary as the persisted state of these relationships.

Conclusions

The literature review concludes with a set of analyses, designed for use by service management or solution owners of composite software solutions. The conclusions and recommendations are intended for reference by stakeholders in the lifecycle of a composite software solution, specifically for service environments that are expected to meet SLA obligations. The goal in this section is to identify gaps in the selected literature related to the focus areas, and propose revisiting certain proposals from the review against updated paradigms. The intent of this researcher is to address the relative merits of the various perspectives in the selected literature, and to highlight specific areas in the maturing field of SLA measurement models for SOA systems, specifically with relation to specifications or patterns that offer promise to alleviate the current complexities related to QoS certification the solutions typical in distributed, composite solution development and support.

Gap Analysis

While most of the literature selected for use in this inquiry discussed potential solutions to the problem, the majority of the use cases appeared to explore a relatively restricted domain and usage model. For example, Bertolino et al. (2007) explored a model of run-time adaption that was predominantly based on distance of SLA violations from expected baseline, but did not expound on the interrelation of secondary QoS and how adapting to meet the needs of one metric could adversely impact another (for example, scaling up on memory allocation to a process can result in slower CPU due to the need for the process to address a larger memory space).

Of a similar vein, while there was a significant amount of discussion in terms of component composition patterns, there would seem to be opportunities to address the overlap of multiple patterns into more complex orchestrations, which are becoming more common in composite enterprise applications. As a corollary, there appeared to be a gap in terms of the differentiation in composite solutions not by service consumers, but by different communication requirements, such as a consumer requiring guaranteed delivery across an asynchronous transport, versus guaranteed delivery across a synchronous channel. As an example, Di Penta et al. (2007), described the application of genetic algorithms to a number of potential paths, but the compositional QoS is determined to be more a factor of the latency in the individual components, as opposed to consumer with different delivery models.

Little discussion in the selected literature related to evaluation of the impact of more granular measurement and analysis on the system being observed, including expectations for overhead and guidance on impact to capital expenditures (additional hardware, software, etc.) and recurring costs (support, licensing, etc.). As mentioned by Bertolini et al. (2007), the application of SLAs monitoring against all potential performance indicators of a commercial service can be extremely expensive, increasing in rough alignment with the growth of users on the system. Nowhere in the surveyed literature, however, was this potential cost more concisely quantified in any specific manner. A review of various categories of QoS indicators, related expectations of tooling support required for successful audit of those indicators, and the expected comparative costs against growth of consumers for the service, would appear a beneficial area for deeper review, due to the very potential impact to underlying cost of service delivery.

No model was presented to describe set-based approaches to capturing points of common KPI analysis, in a manner similar to Venn diagramming. It would be helpful to look at the multiplicity of potential SLA conformance metrics in terms of zones of commonality for different key aspects, such as measurement dependencies on infrastructure, instrumentation complexity for the monitored system environments, or the expected QoS priorities by different segments of service consumers. An analogous method can be found in the Hertzsprung-Russell diagram method (Wikipedia, 2008), used in astronomy for analyzing common grouping and standard deviation from the mean for stars, based on key physical attributes.

Suggestions for Specific Solution Proposals That Could Be Revisited

This researcher believes that a number of areas in the surveyed literature would benefit from the application of the proposed solution or investigation to different, or possibly new, implementation environments for two reasons. The first concerns new service delivery models in current SOA ecosystems. The second concerns areas that while not directly considered in the original source literature, still seem to pose questions that could be of interest to this audience.

Different Client Platforms

The monitoring infrastructure methodology outlined by Molina-Jimenez et al. (2004) reflected a focus on proximity to the provider and consumer of a service, specifically as related to communication latency. There is an opportunity to revisit this model with a broader range of usage models and QoS metrics to collect and correlate. A more current potential example would

include the delivery of equivalent services to a set of distinct device platforms, specifically with consumer services to traditional desktop platforms, against mobile or small form factor internet devices, which Church, Smyth, Cotter, and Bradley (2007) have noted will shortly rival these more traditional platforms for access to digital information. The medium over which these devices depend on as a primary means of communication does not have the bandwidth and QoS guarantees as would be expected of more established network communication channels (Perera, Sivaraman, & Seneviratne, 2004), and would appear to present a significant additional dimension to the application of SLA for service delivery.

Another theme from the review, the application of genetic algorithms, such as those described by Di Penta et al., could be applied to a broader range of applications, and would suggest the application against a broader scope of variability in service delivery models, such as those across mobile communication channels. These will assumedly be of great interest to service owners that are considering opportunities for service delivery across a variety of distinct delivery mediums, due to potential for increased opportunities for growth.

Different Categories of Service Partners:

Large enterprises are common in the modern work, and while there are some relatively similar models of contractual obligations with external and internal parties, there is an assumption is that there are some differentiating aspects when working with service providers within what Mitchell & McKee (2005) term the 'resource layer of enterprises'. As opposed to a services exchange model with a completely external entity, there are assumed benefits associated with a service provider-consumer relationship within the boundaries of the organization that would imply a

include lower cost for subscription to services, as well as a potential for better chance of alignment on capabilities for those services and the associated schedules for release and upgrade, due to shared financial goals. However, the potential problems associated with service usage in a closed environment can also imply a narrow scope of options for services, or potentially a less competitive drive to provide services to a limited audience, as opposed to public sector consumers. Also, related to a topic that is very contemporary in the software industry (Carmel & Abbott, 2006), this reviewer suggests a comparison between onshore and offshore service organizations in terms of quality of service, cost models, and expectations for audit, versus the expected additional latency and potential security/privacy challenges for certain types of data.

Recommendations for Stakeholders of Composite Software Solutions

In particular, Molina-Jimenez et al. (2004) discuss the emergence of third party audit organizations which provide to parties in a service transaction arrangement the role of trusted third party auditor, by inserting an analysis and collection system at critical points in the public network. Such organizations can bring capabilities to bear on the problem of both QoS indicator definition and contractual definition before deployment of services, as well as the instrumentation of the services ecosystem for access to KPIs, runtime monitoring and alerting to relevant parties on non-conformance incidents, and the important task of assigning ownership for non-conformance events. Service owners who intend to provide services that involve SLAs that involve multiple public consumers, and may not wish to make a significant direct investment of capital into runtime monitoring infrastructure and support resources, would be well served to consider the engagement with such an organization.

Also of interest to stakeholders involved in the pre-deployment modeling or validation of services is the introduction of nascent/evolving techniques for modeling services, such as with the statistical analysis and SLA scenario generation through the use of artificial intelligence systems such as those outlined by Liu, Bryant, Gray, Raje, Olson, & Auguston (2005). Another example of such prescriptive analysis through recent techniques includes the application of Genetic Algorithms in the analysis of SOA solution environments, which can generate test data along multiple potential evolutionary paths in a sequence of composite calls, and shows promise for highlighting SLA non-conformance scenarios before release to the general public (Di Penta et al., 2007). Through the use of such methodologies, a service implementation team can identify what Di Penta et al. (2007) term “QoS-risky paths” as early as possible, which may sound immediately intuitive to the reader, but can be difficult in most cases to divine in the early stages of service software development.

A further suggestion for SOA solution owners is to not underestimate the human factor in applying SLA models, as it is highly likely that various stakeholders in the service delivery lifecycle most likely also have legacy expectations on the means and methods to develop and support such solutions. In order to drive stakeholders to a common frame of reference, it is essential to resolve and describe QoS metrics in terms of standard grammars and specifications, of which WS-Agreement seems to provide a well-regarded starting point (Sun, 2004). As noted by Skene, Lamanna, and Emmerich (2004), the more precision that can be applied to describing the expected behavior for the service between parties, the greater the potential to reduce variations of interpretation of the obligations.

Finally, it also deemed critical that service developers and owners have a complete understanding of the full scope of operations and interdependencies between all of the components services within a software solution. Two key facets are involved. The first of these facets is the medium and method for communication between services. As noted by Molina-Jimenez et al. (2004), the communication network used for delivery of services, most specifically over the internet, can present one of the least controlled mediums for SLA conformance, because communication can involve many potential partners in the delivery, some of the them unknown until the point of service delivery. The second facet is the potential paths and models for activities in a functional transaction, or what Sun (2004) describes as the compositional patterns for component software solutions, because understanding the dependencies and potential points of resource contention can assist in early avoidance of service bottlenecks.

References

- Atkinson, D.C., & Griswold, W.G. (1996) The Design of Whole-Program Analysis Tools, Retrieved May 4, 2008 from: <http://ieeexplore.ieee.org/iel2/3540/10631/00493398.pdf>
- Barber, G. (2007) The Business Value Proposition of SCA, Retrieved June 4, 2008 from: <http://www.osoa.org/display/Main/The+Business+Value+Proposition+of+SCA>
- Bell, C. (2008) Critical information of Information Sources, Retrieved April 30, 2008 from: <http://libweb.uoregon.edu/guides/findarticles/credibility.htm>
- Bertolino, A., De Angelis, G., Sabetta, A., & Elbaum, S. (2007) Scaling up SLA Monitoring in Pervasive Environments, *International workshop on Engineering of software services for pervasive environments*. Retrieved April 15, 2008 from ACM Digital Library: http://portal.acm.org/ft_gateway.cfm?id=1294914&type=pdf&coll=ACM&dl=ACM&CFID=64337530
- Boschia, E., Denazisa, S., & Zsebyb, T. (2005) A measurement framework for inter-domain SLA validation, Retrieved May 6, 2008 from ScienceDirect: http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6TYP-4H16KXH-2-Y&_cdi=5624&_user=852645&_orig=search&_coverDate=03%2F31%2F2006&_sk=999709993&view=c&wchp=dGLbVtz-zSkzS
- Cardellini, V., Casalicchio, E., Grassi, V., & Lo Presti, F. (2007) Efficient Provisioning of Service Level Agreements for Service Oriented Applications, *2nd international workshop on Service oriented software engineering*, Retrieved April 17, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1294936&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Carmel, E., & Abbott, P. (2006) Configurations of global software development: offshore versus nearshore, Retrieved June 17, 2008 from:

http://portal.acm.org/ft_gateway.cfm?id=1138509&type=pdf&coll=ACM&dl=ACM&CFID=74065775&CFTOKEN=45532706

Castagna, G., Gesbert, N., Padovani, L. (2008), A theory of contracts for web services, *Annual Symposium on Principles of Programming Languages archive*, Retrieved April 22, 2008

from ACM Digital Library: <http://delivery.acm.org/10.1145/1330000/1328471/p261-castagna.pdf?key1=1328471&key2=0516109021&coll=ACM&dl=ACM&CFID=65186722>

Church, K., Smyth, B., Cotter, P., & Bradley, K. (2007) Mobile information access: A study of emerging search behavior on the mobile Internet, Retrieved June 16, 2008 from:

<http://delivery.acm.org/10.1145/1240000/1232726/a4-church.pdf?key1=1232726&key2=3735393121&coll=ACM&dl=ACM&CFID=74065775&CFTOKEN=45532706>

Ciolek, T.M. (1996), The Six Quests for the Electronic Grail: Current Approaches to Information Quality in WWW Resources, Retrieved April 30 2008 from

<http://www.ciolek.com/PAPERS/six-quests1996.html>

Cooper, H. (1998) *Synthesizing Research: A guide for Literature Reviews* (3rd ed.). Thousand Oaks, CA: Sage.

Cuadrado, F., García, B., Dueñas, J. C., & Parada, H. A. (2008) A Case Study on Software Evolution towards Service-Oriented Architecture, Retrieved May 9, 2008 from:

<http://ieeexplore.ieee.org/iel5/4482830/4482831/04483115.pdf?tp=&isnumber=&arnumber=4483115>

Di Penta, M., Canfora, G., Esposito, G., Mazza, V., & Bruno, M. (2007) Search-based Testing of Service Level Agreements, *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, Retrieved April 16, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1277174&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Fitzgerald, B., Olsson, C.M., Achatz, R., Bosch, J., Rombach, D., Beauvais, T., Fuggetta, A., Banâtre, J., Bancilhon, F., De Panfilis, S., Bomarius, F., Saikkonen, H., Kuilder, H., & Boeckle, G. (2006) The Software and Services Challenge. Contribution to the preparation of the Technology Pillar on “Software, Grids, Security and Dependability” of the 7th Framework Programme, Retrieved May 14, 2008 from:

ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/st-ds/fp7-report_en.pdf

Gill, N. (2007) Importance of Software Component Characterization For Better Software Reusability, *ACM SIGSOFT Software Engineering Notes, Volume 31, Issue 1*, Retrieved April 16, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1108771&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Grundy, J., Hosking, J., Li, L., & Liu, N. (2006) Performance Engineering of Service Compositions, *Proceedings of the 2006 international workshop on Service-oriented software engineering*. Retrieved April 16, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1138493&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Gu, Q. & Lago, P. (2007) A stakeholder-driven service life cycle model for SOA, Foundations of Software Engineering archive, Retrieved April 13, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1294930&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Hewitt, M. (1998) Carrying out a literature review, Trent Institute of Health Services Research, Retrieved April 30, 2008 from:

<http://128.223.179.107/aim/Capstone07/HewittLitReview.pdf>

Kajko-Mattsson, M., Lewis, G.A., Smith, D.B (2007) A Framework for Roles for Development, Evolution and Maintenance of SOA-Based Systems, Retrieved May 13, 2008 from:

<http://delivery.acm.org/10.1145/1280000/1270304/29600007.pdf?key1=1270304&key2=3915280121&coll=Portal&dl=ACM&CFID=27982995>

Kontogiannis, K., Lewis, G.A., Smith, D.B., Litoiu, M., Muller, H., Schuster, S., Stroulia, E.

(2007) The Landscape of Service-Oriented Systems: A Research Perspective, Retrieved May 14, 2008 from:

<http://delivery.acm.org/10.1145/1280000/1270298/29600001.pdf?key1=1270298&key2=1915280121&coll=Portal&dl=ACM&CFID=27982995>

Leedy, P. & Omrod, J. (2005) Practical research: Planning and Design (8th ed.). Saddle River, NJ: Pearson Merrill Prentice Hall.

- Leung, C. M. & Schormans, J. A. (2005) Measurement-based end to end latency performance prediction for SLA verification, *Journal of Systems and Software archive*, Retrieved May 6, 2008 from: http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6V0N-4BMCFHJ-1-41&_cdi=5651&_user=852645&_orig=search&_coverDate=02%2F01%2F2005&_sk=99259996&view=c&wchp=dGLbVlb-zSkWW
- Liu, S., Bryant, B., Gray, J., Rajee, R., Olson, A., Auguston, M. (2005) Two-level assurance of QoS requirements for distributed real-time and embedded systems, *Symposium on Applied Computing archive*, Retrieved April 23, 2008 from ACM Digital Library: <http://delivery.acm.org/10.1145/1070000/1066884/p903-liu.pdf?key1=1066884&key2=5987109021&coll=ACM&dl=ACM&CFID=65186722>
- Masche, P., Mckee, P., & Mitchell, B. (2006) The increasing role of service level agreements in B2B systems, Retrieved May 14, 2008 from: http://www.nextgrid.org/download/publications/The_increasing_role_of_service_level_agreements_in_B2B_systems.pdf
- Mitchell, B. & Mckee, P. (2005) SLAs A Key Commercial Tool, Retrieved May 15, 2008 from: <http://www.nextgrid.org/download/publications/echallenges.pdf>
- Molina-Jimenez, C., Shrivastava, S., Crowcroft, J., & Gevros, P. (2004) On the Monitoring of Contractual Service Level Agreements, Retrieved May 14, 2008 from: <http://ieeexplore.ieee.org/iel5/9222/29240/01319502.pdf?tp=&isnumber=&arnumber=1319502>

- Morgan, G., Parkin, S., Molina-Jimenez, C., Skene, J. (2007) Monitoring Middleware for Service Level Agreements in Heterogeneous Environments, Retrieved May 14, 2008 from: <http://www.cs.newcastle.ac.uk/publications/inproceedings/papers/898.pdf>
- Muthusamy, V., Jacobsen, H., Coulthard, P., Chan, A., Waterhouse, J., & Litani, E. (2007) SLA-Driven Process Management in SOA, *Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, Retrieved April 16, 2008 from ACM Digital Library: http://portal.acm.org/ft_gateway.cfm?id=1321243&type=pdf&coll=ACM&dl=ACM&CFID=64337530
- Niles, K. (2002) Characterizing the Measurement Process, SixSigma Insights Newsletter. Vol. 3, #42. ISSN: 1530-7603. Retrieved May 6, 2008 from: <http://www.iSixSigma.com/library/content/c020527a.asp>
- Obenzinger, H. (2005) What can a literature review do for me? How to research, write, and survive a literature review. Retrieved April 20, 2008 from <http://ual.stanford.edu/pdf/LiteratureReviewHandout.pdf>
- O'Brien, L., Merson, P., & Bass, L. (2007) Quality Attributes for Service-Oriented Architectures, *International Conference on Software Engineering archive*, Retrieved April 11, 2008 from ACM Digital Library: http://portal.acm.org/ft_gateway.cfm?id=1270300&type=pdf&coll=ACM&dl=ACM&CFID=64337530
- Papazoglou, M. P., Heuvel, W. (2007) Service oriented architectures: approaches, technologies and research issues, *The VLDB Journal archive*, Volume 16 , Issue 3, July 2007, Retrieved May 12, 2008 from:

http://delivery.acm.org/10.1145/1270000/1265298/778_2007_Article_44.pdf?key1=1265298&key2=2815280121&coll=Portal&dl=ACM&CFID=27982995

Peng, W., Sun, T., Rose, P., & Li, T. (2007) A Semi-automatic System with an Iterative Learning Method for Discovering the Leading Indicators in Business Processes, Retrieved May 4, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=1288557&type=pdf&coll=GUIDE&dl=GUIDE&CFID=27092420

Perera, E., Sivaraman, V., & Seneviratne, A. (2004) Survey on network mobility support, Retrieved June 17, 2008 from

http://portal.acm.org/ft_gateway.cfm?id=997127&type=pdf&coll=ACM&dl=ACM&CFID=74065775&CFTOKEN=45532706

Skene, J., Lamanna, D., & Emmerich, W. (2004) Precise Service Level Agreements, *Proceedings of the 26th International Conference on Software Engineering*, Retrieved April 14, 2008 from ACM Digital Library:

http://portal.acm.org/ft_gateway.cfm?id=999422&type=pdf&coll=ACM&dl=ACM&CFID=64337530

Skene, J., Skene, A., Crampton, J., & Emmerich, W. (2007) The monitorability of service level agreements for application-service provision, *Run-time performance and resource-awareness -- Proceedings of the 6th international workshop on Software and performance*, Retrieved April 21, 2008 from ACM Digital Library:

<http://delivery.acm.org/10.1145/1220000/1216997/p3-skene.pdf?key1=1216997&key2=4361109021&coll=ACM&dl=ACM&CFID=65186722>

Wikipedia (2008) Hertzprung-Russell Diagram, Retrieved June 19, 2008 from:

http://en.wikipedia.org/wiki/Hertzprung-Russell_diagram

Standler, R.B. (2004), Evaluating Credibility of Information on the Internet, Retrieved April 29,

2008 from <http://www.rbs0.com/credible.pdf>

Sun, C. (2004) Empirical reasoning about quality of service of component-based distributed systems , *ACM Southeast Regional Conference archive*, Retrieved April 23, 2008 from

ACM Digital Library: <http://delivery.acm.org/10.1145/990000/986620/p341-sun.pdf?key1=986620&key2=9677109021&coll=ACM&dl=ACM&CFID=65186722>

Zitouni, A., Seinturier, L., & Boufaida, M. (2007) Contract-based approach to analyze software components, Retrieved April 29, 2008 from:

<http://hal.inria.fr/docs/00/27/15/33/PDF/zitbousei2-uml%5B1%5D.pdf>