Management of Enterprise Quality of Service

● Koji Ishibashi ● Mikhail Tsykin

(Manuscript received November 28, 2003)

Service level agreements (SLAs) are commonly used to ensure the quality of service (QoS). With the advent of Grid and on-demand computing, QoS management promises to become business-critical. Fujitsu is at the forefront of the industry's QoS management development and standardization efforts. This paper presents some new QoS concepts and techniques, introduces Fujitsu's latest QoS management product–Systemwalker Service Quality Coordinator (SSQC)–and discusses the Application Quality/Resource Management (AQRM) Initiative of The Open Group.

1. Introduction

There can be many problems in managing the performance and capacity of large-scale systems consisting of thousands of nodes. However, there is a general agreement among IT professionals that, given adequate tools and skilled people, problems in the measuring and reporting of performance metrics can be overcome. Unfortunately, this is not the case for the prediction of enterprise quality of service (QoS) and service levels. One of the reasons for this is that there are no established definitions, vocabulary, or standards in these fields. In their absence, it is usually assumed that:

- QoS is often represented by the end-to-end response time (ETE RT), and
- very short-term prediction is required.

Both assumptions are natural. ETE RT is indeed the prime metric for determining service levels.[1),2)] As for its prediction, the common perception is that transaction-based ETE RT is operationally useful only if it can warn system and network managers to expect immediate problems.

However, enterprise QoS should be considered within the framework of formal service level agreements (SLAs). Measure of contractual compliance with these is the only suitable, acceptable, and useable QoS metric at the enterprise level. Such compliance is assessed over a period of time (frequently a month), and the metrics used are statistical in nature (averages and percentiles are common). Therefore, it is useful to assess how an organization is "tracking" against an SLA. Even more useful is to know what level of service is required for future compliance.

This approach is useful not only on an enterprise level. Recent developments in telecommunications and Web Services promote two new types of SLAs: dynamic and auction (see Section 2.1). These are concluded automatically between programs and must also be managed automatically. They are expected to be widely used in Grid and on-demand computing, for example.

Given the diversity of the Open Systems, it is clear that industry-wide standardization of QoS management is required if it is to realize its full potential. At present, this effort is centered at The Open Group (see Section 4). Other relevant industry bodies such as DMTF, OASIS, CMG, ITSMF, SNIA, and GGF mostly maintain "watching briefs" at this stage.

Thus, the objective of this paper is three-fold:

1) Present new, modern concepts for QoS management
2) Introduce Systemwalker Service Quality Coordinator (SSQC), which is a new product designed to support QoS management within the framework of TRIOLE
3) Briefly mention the Application Quality/Resource Management (AQRM) Initiative for standardization of QoS management

2. Concepts

2.1 Definitions

Until recently, basic QoS definitions did not exist (although the authors have offered some in a previous publication[3]):

1) Availability

This is a metric that shows whether an entity operates normally (i.e., is available for business) during a given interval. There are two main types:

- Measured

An entity is available if there is evidence of normal completion of standard activities. For instance, an application is available if transactions (user or robotic) are completed normally.

- Derived

An entity is available if there is no evidence to the contrary. For example, an application is available if there are no user complaints that it is not available.

2) Customer or client

This means the service recipient. The service recipient can be an internal or external recipient.

3) End-to-end

This is a description of a metric, process, or entity that characterizes a complete route or path. For instance, the ETE RT describes the response time for a transaction across its entire path.

4) Measurement

- Real-time

Conducted at least at the time resolution of the process being measured or managed. For instance, both monthly and sub-second measurements are real-time providing they are appropriate for a given process. Good examples are management of static SLAs and mainframe performance management, respectively.

- Near-real-time

Conducted at the timescale of the process being measured, but at a coarser resolution than the process itself. One way to describe the timescale of this measurement is that it is sufficient to detect the desired phenomenon and affect the process as required.

- Non-real-time

Conducted at a significantly lower resolution than the timescale of the process being measured.

- Stochastic

Determines various characteristics of a measured process (its distribution, standard deviation, variance, etc.). Stochastic measures may be derived by:

– Sampling or statistically probing the process being measured
– Performing statistical operations over the sample population available a-priori

In both cases, care must be taken to ensure that the sample adequately represents the process as a whole.

- Deterministic

Measures every instance of a process (every individual transaction, IP packet, etc.)

5) Quality of experience (QoE)

Any metric that shows whether the experience of a customer with regard to service, either documented in an SLA or implied (but measurable), was met over a service period.

6) QoS

Any metric that shows whether the requirements of an SLA or implied (but measurable) customer expectations were met over a service period.

7) Real-time enterprise (or E-enterprise)

An enterprise that conducts its IT operations on its business timescale.

8) Service

Measurable result or outcome of a business or technical process involving a supplier or server and a customer or client.

9) SLA

A formal agreement (contract) between a supplier and customer that formalizes the details of a service (contents, price, delivery process, acceptance and quality criteria, penalties, etc.). Informal SLAs exist also and may be as important and binding as formal ones. However, informal SLAs cause misunderstandings, misinterpretations, and disputes to a greater extent than formal ones and are not recommended. There are three major types of SLAs:

- Static

An SLA that generally remains unchanged for multiple service periods. These SLAs are common in outsourcing.

- Dynamic

An SLA that is generally changed from service period to service period in order to accommodate changes in the provision of the service. These SLAs are increasingly common in telecommunications.

- Auction

A new type of SLA associated mostly with Web Services. It is used for automated selection of the most suitable supplier.

10) Service period

The period to be assessed for QoS. For a business process subject to an SLA, the service period is typically a calendar month. For other processes, it may be a transaction or any other measurable and relevant period of time.

11) Suppliers and server service providers.

These may be internal (e.g., an IT department) or external (e.g., an outsourcer).

2.2 Measurement of QoS

By definition, this occurs after a service has been delivered by the supplier and used by the customer. Some examples are:

1) The SLA report is delivered after the end of the service period
2) The end-to-end response time is calculated once a transaction is completed
3) The packet loss and similar telecommunication metrics are calculated over agreed periods of time

2.3 QoS management: reactive vs. predictive

2.3.1 Common (reactive) approach

The conventional way to manage a service is to measure the QoS and then determine whether the requirements have been met. This means that problems are detected and then corrective action is taken.

This approach presents problems for everybody, for instance:

1) For customers: Deterioration of service cannot be prevented. Problems must happen before corrective action can be taken. In worst cases, on-going operation of business may suffer.
2) For suppliers: Penalties (frequently specified in modern SLAs) cannot be avoided. In the worst cases, business continuation becomes problematic.

2.3.2 Predictive approach

By contrast, the authors rely on predicting the result of QoS compliance. Therefore, it is frequently possible to take corrective action before a problem occurs, thus eliminating it or, at least, minimizing its impact.

This approach avoids the problems associated with the reactive approach:

1) For customers: Deterioration of the service can be prevented and business conduct optimized.
2) For suppliers: SLA penalties can be avoided and prospects for business continuation improved.

2.4 Basic technical principle: near-real-time, asynchronous operation

Any information acquisition and delivery process can be reduced to four basic tasks:

1) Capture: Acquisition of raw data
2) Transformation: Conversion of data into information
3) Delivery: Information delivery to the recipient
4) Consumption: Usage of information

A process incorporating these tasks is usually considered to be a synchronous process, because the next task is started when the current task is completed. This is the principle of the so-called real-time approach, in which data is captured and delivered to users as is or with minimal transformation. This synchronicity represents a major problem in system management, because it leads to either overloading of networks due to high-volume data movements (sub-second cycle) or poor quality data due to long sampling cycles.

The authors use the so-called near-real-time approach, in which the timing of information delivery is dictated by the users' needs. For instance, if a user manages a typical IT environment, sub-second data availability is pointless, because network delays will ensure that such data is outdated when it arrives. Once-a-minute data delivery is fine. However, data can be captured frequently, buffered, processed, and made available as needed. This keeps data movements low while preserving the quality of data. This is accomplished by keeping the four basic tasks asynchronous (**Figure 1**).

3. Fujitsu's solution

3.1 Process

QoS management, similarly to capacity and performance management (or any other management), involves four activities:

1) Monitoring the QoS
2) Reporting the QoS
3) Predicting the QoS
4) Maintaining the QoS

In real terms, QoS management equates to the conventional discipline of capacity planning, but with an overriding emphasis on the achievement of QoS objectives (see the definitions given in Section 2.1). Thus, it may be safely said that QoS management incorporates the following conventional disciplines:

1) Capacity planning
2) Performance management
3) Service level reporting

With this in mind, the authors define the objectives of QoS management as follows:

1) Continuous collection and remote storage of performance and user data in order to support the data needs and automation of subsequent processes
2) Short-term monitoring of system and application resources to affect the following:

- Problem detection
- Identification of potential problems
- Prediction of problems
- Initiation of problem alerts

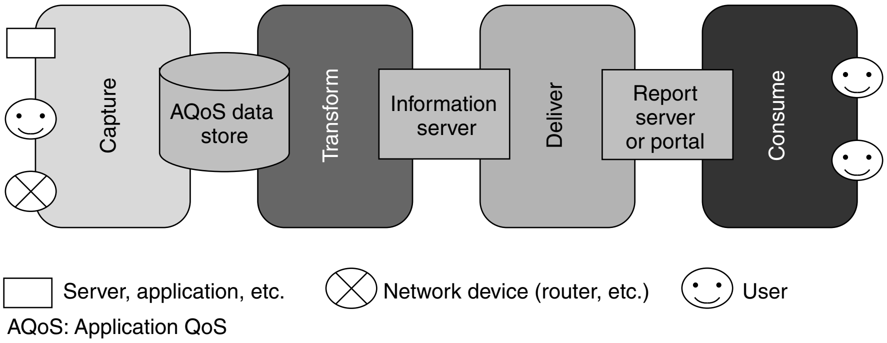

Figure 1
Four asynchronous loops for QoS information.

3) Regular service level reporting, including the prediction of results for the next service level interval, in accordance with an SLA
4) Prediction of potential problems in the medium and long-term time frames
5) On-demand capacity planning and sourcing of pre-collected, fit-for-purpose data from remote storage
6) On-demand Web-enabled reporting

Figure 2 shows the process diagram. Note the continuous nature of the collection, prediction, and reporting activities. In contrast, alert generation and the resulting tuning and capacity planning activities are irregular (on-demand). This is in keeping with the asynchronous process described above (Figure 1). Detailed descriptions of the process are available in References 3) and 4).

3.2 SSQC

3.2.1 Description

SSQC is a product designed to collect, store, and provide on demand a variety of data relating to computer and network performance, management, and administration.

SSQC is intended to operate as part of the TRIOLE framework to provide support for availability monitoring, service level management, and performance management activities.

SSQC operates in a three-tier architecture (**Figure 3**) by doing the following:

1) Collecting data from application, Web, and DB servers at a local or departmental level
2) Collecting end-user response times and service response times
3) Storing detailed collected data on a local department server (DS)
4) Storing summary data on an enterprise server (ES)

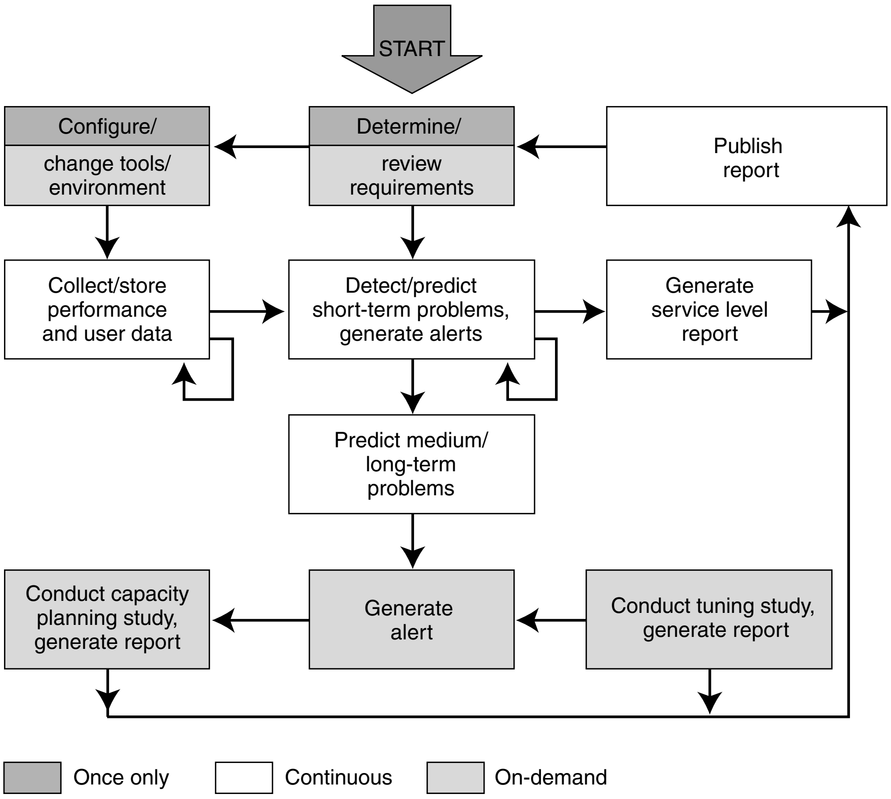
START
Configure/
change tools/ environment
Determine/
review requirements
Publish report
Collect/store performance and user data
Detect/predict short-term problems, generate alerts
Generate service level report
Predict medium/ long-term problems
Conduct capacity planning study, generate report
Generate alert
Conduct tuning study, generate report
Once only
Continuous
On-demand

Figure 2
QoS management process.

5) Providing data for extraction, analysis, and presentation

SSQC is intended to run 24 hours a day 365 days a year with minimal management and will run on a variety of extant Windows and Unix platforms.

3.2.2 Function

SSQC will directly collect Windows and Unix system, network, Oracle, Symfoware, and Interstage performance data using the standard system or RDBMS commands or other published interfaces such as the Windows registry or Oracle management tables. SSQC will collect end-user response times on Windows clients and measure service responses and availability. SSQC will also accept and process data collected by external tools and processes.

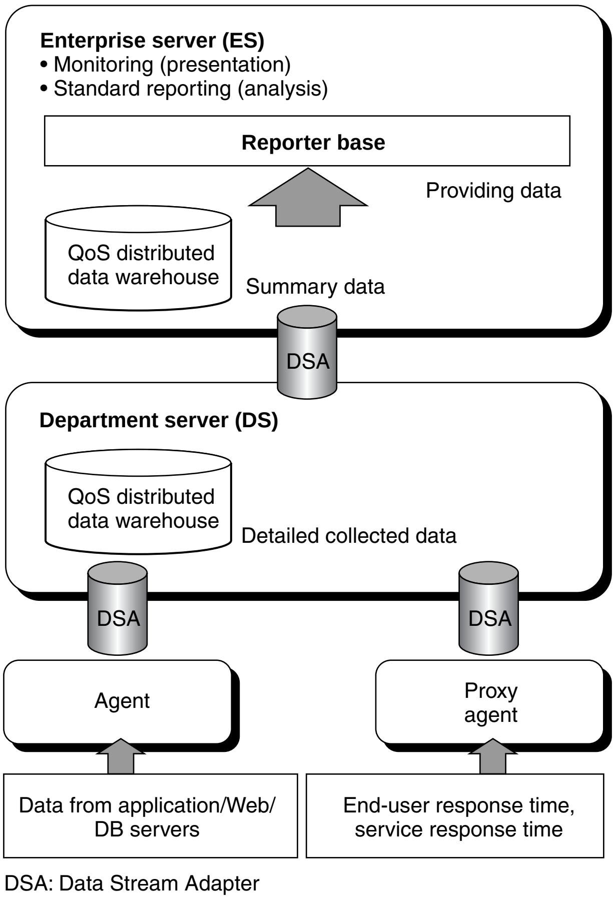
Enterprise server (ES)
• Monitoring (presentation)
• Standard reporting (analysis)
Reporter base
Providing data
QoS distributed data warehouse
Summary data
DSA
Department server (DS)
QoS distributed data warehouse
Detailed collected data
DSA
DSA
Agent
Proxy agent
Data from application/Web/ DB servers
End-user response time, service response time
DSA: Data Stream Adapter

Figure 3
Architecture of SSQC.

The objective of the current version of SSQC is to enable proactive management of the QoS. To meet this objective, it delivers the service-level visualization functionality (from the end user's perspective) and the relevant performance data.

Future versions will support, amongst other things, an advanced inference analysis function for identifying transaction processing bottlenecks and performing other tasks.

Data will be processed and formatted on the machine where collection is done and then forwarded to a DS. Then, some of the data will be stored at the DS and some will be forwarded to an enterprise server for storage. Collected data will be managed according to the scheme presented in **Table 1**.

The four different types of data will be generated on the agent system on which they are collected. Data will be automatically deleted once it exceeds the applicable age limit.

Data will be stored in a distributed data warehouse. Data management and data dictionary facilities are provided.

Display, analysis, and reporting of SSQC data and cross-correlation with data from other sources will be performed on an enterprise server using a powerful data analysis and reporting tool called OCMM. A detailed description of OCMM's functionality is beyond the scope of this paper.

3.3 QoS management in TRIOLE

SSQC is one of the basic components of TRIOLE.[5] A detailed description of TRIOLE is

Table 1
Data management scheme.

Data type	Storage scheme
1-minute resolution	Stored at the collection system. Not managed by SSQC
10-minute resolution	Stored on the department server; kept for 14 days
1-hour resolution	Stored on the department server; kept for 92 days
1-day resolution	Stored on the department server; kept for 731 days
Summary	Stored on the enterprise server; kept for 1 day (renewed daily)

beyond the scope of this paper, but a brief discussion is in order. For more information, the reader is referred to Reference 5).

Two of the major objectives of TRIOLE are stability and reliability throughout the entire system and reduced total cost of ownership (TCO). This will be accomplished via a pervasive, system-wide optimization of resource consumption and automated compliance with SLAs: in other words, QoS management as defined in Section 2.1. QoS management will increase in importance with the increasing adoption of Grid and on-demand computing, because it will enable scheduling, reconfiguration, and billing in "blade" grids.

These objectives of stability and reliability will be accomplished by Systemwalker Resource Coordinator, based on information about resource utilization and SLA compliance delivered to it by SSQC.

4. AQRM

4.1 Introduction

SSQC provides a QoS management platform for large-scale, heterogynous open systems. For that purpose, it captures relevant information from a variety of platforms. However, currently it cannot capture all the necessary data.

The requirement is to capture attribution, resource consumption, state, and QoS information for units of work (transactions) as they traverse different system and application domains, mutate, split, and recombine. The problem is that a transaction identity is not preserved across domains or even within a given domain. The only answer is an industry standard or another type of standard, and the task of establishing such a standard has fallen on the AQRM Forum of The Open Group.

AQRM emerged in February 2003 as a joint activity of the QoS Taskforce and Enterprise Management Forum. A summary of its "Call to Action," which describes the reasons for its formation and major objectives, is given below. Up-to-date information is available at The Open Group Web site.[6] The authors believe that active participation in AQRM is necessary for the continued evolution of the industry and they personally participate on behalf of Fujitsu. They have contributed a functional description of the Application QoS (AQoS), which formed the backbone of AQRM's standardization activities.

4.2 Call To Action: a summary

The IT industry is starting a major technology shift towards modular computing and component-based Web Services application architectures. In addition, businesses are shifting towards the real-time enterprise model.

Taken together, these driving forces represent a fundamental change in the way systems will be implemented and, even more importantly, managed. As a result, a new management paradigm is required.

If the IT industry is to respond to these changes, we will need an appropriate set of AQRM standards to enable the integration of applications and management systems. However, at present there is no single, cohesive industry-standards effort focused on addressing these two issues.

This represents a strategic opportunity to form an influential standards initiative that will:

1) Establish industry acceptance of an open architecture for managing such environments
2) Accelerate the availability of application and infrastructure instrumentation
3) Provide a vehicle that enables all relevant constituencies of the IT industry to cooperate with each other
4) Solve the current and future applications management problems

The main tasks that must be accomplished to form such an initiative are 1) recruit a critical mass of key players in a timely fashion and 2) commit sufficient resources to lead the group and accomplish the technical work.

5. Conclusion

In this paper, the authors accomplished the following:

1) Presented new concepts for QoS management
2) Introduced SSQC, which is a new product for supporting QoS management within the framework of Fujitsu's TRIOLE concept
3) Briefly mentioned the AQRM Initiative for the standardization of QoS management

The authors plan to ensure that Fujitsu's customers continue to benefit from progressive enhancements of SSQC. These enhancements will comply with the principles described above and will reflect the progress of AQRM's standardization activities.

References

1) M. Tsykin: Service Level Measurement: Checkpoint 2000. The Guide To IT Service Management, J. Van Bon, A. Wesley, 2002, p.324-334.
2) M. Tsykin: On Web Quality of Service: Approaches to Measurement of End-to-End Response Time. Lecture notes in Computer Science, Volume 2094/ 2001, Pascal Lorenz, Springer-Verlag Heidelberg, p.291-301.
3) M. Tsykin et al.: Automated Monitoring and Reporting of Enterprise Quality of Service. Proceedings of The 7th World Multi-conference on Systemics, Cybernetics and Informatics (SCI 2003), Orlando, July 2003.
4) M. Tsykin et al.: On Automated Monitoring of SLAs. CMG Journal of Capacity Management, Summer 2002, CMG, p.27-36.
5) TRIOLE.
http://www.fujitsu.com/services/solutions/triole/vision/
6) The Open Group AQRM Web Site.
http://www.opengroup.org/aquarium/

Koji Ishibashi received the B.S. degree in Communication Engineering from Osaka University, Osaka, Japan in 1981. He joined Fujitsu Ltd., Kawasaki, Japan in 1981, where he has been engaged in research and development of system management software since 1990. Currently, he is responsible for developing Systemwalker Service Quality Coordinator.

Mikhail Tsykin received the Diploma in Hydrology from Moscow State University in 1973. He joined Fujitsu Australia Limited in 1980, where he founded the Systems Engineering Research Centre (SERC) in 1988. He is currently the Senior Business Development Manager at SERC. He is a member of the Computer Measurement Group and a Steering Committee member of the AQRM Forum of The Open Group. He is a regular contributor to national and international conferences, with over 30 papers to his credit, and a recipient of the Best Paper Award at the SCI'2003 Conference.