SLA Monitoring and Management Framework for Telecommunication Services

Jacek Kosiński, Piotr Nawrocki, Dominik Radziszowski, Krzysztof Zieliński, Sławomir Zieliński
AGH University of Science and Technology
{jgk, piter, radzisz, slawek, kz} @ics.agh.edu.pl

Grzegorz Przybylski, Paweł Wnęk
Comarch S.A
pawel.wnek@comarch.com
grzegorz.przybylski@comarch.pl

Abstract

This paper presents SLA monitoring and management framework for telecommunication services. The basic requirements of this class of systems are specified and verified in context of existing SLA standards and tools. The proposed system architecture is very general and may interoperate with the existing performance monitoring systems and management tools. The key design effort is focused on the general mapping between observed activities of the underlying telecommunication infrastructure elements and corresponding SLA instance object state construction. The proposed framework application is illustrated by the simple case study.

1. Introduction

SLAM (Service Level Agreement Management) [1] is gaining recently increasing attention of telecom services providers and customers. Due to progress in QoS services provisioning and dynamic interactive control and monitoring of telecommunication resources the SLA (Service Level Agreement) management is the most demanding functionality. From the telecom system architecture point of view - SLAM presents another layer, operating over monitoring and resource control layers and providing more abstract and consistent view of services offered to end-users. It is a next logical step of the development of these systems, connecting together a technical perspective of system infrastructure operation with market and user-driven business objectives of the telecom company [2,3].

SLA in general specifies expectations about the provided service quality seen from the end-user viewpoint which are expressed as rather aggregated metrics defined for the selected time of the service provisioning. Difficulties in SLAM construction and implementation are generated by complexity of SLA contract metrics representation and on-line, bi-directional transformation of the business perspective to the technical infrastructure operational parameters – Fig.1.

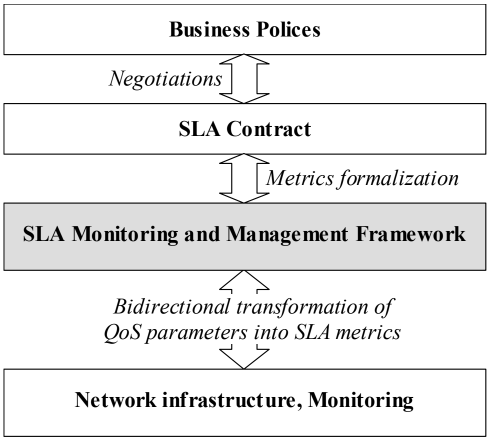

Business Polices
Negotiations
SLA Contract
Metrics formalization
SLA Monitoring and Management Framework
Bidirectional transformation of QoS parameters into SLA metrics
Network infrastructure, Monitoring

Figure 1. Information flow in SLAM framework

This paper describes SLA monitoring and management framework designed and implemented under “SLA Management Framework for Telco Service Providers” Eureka Project. Successful system construction was possible due to preexisting expertise and technology in area of QoS monitoring and control provided by ComARCH [9]. The most innovative part of this research concerns components for monitoring layer data transformation and their propagation to SLA management subsystem.

The structure of the paper is as follows. In Section 2 short overview of exiting SLA management tools is presented. Next in Section 3 SLA metrics are specified very briefly. Section 4 describes SLA monitoring and management framework requirements. The proposed framework architecture is presented in Section 5. Finally Section 6 contains SLAM application case study. The paper is ended with conclusions.

2. Existing SLA management standards and tools

The SLA [1] management tools let enterprise IT managers look at network performance the similar way the service provider does. SLA management tools such as CISCO TMS, ViewGate Networks' Inteligo, FireHunter (Agilent) and Oblicore Guarantee are typically deployed by the service providers. Often these tools are used by the enterprise (via a Web interface), allow IT personnel to monitor network performance and manage their network SLAs in real time. An important aspect of management tools is SLA specification language such as WSLA.

SLA tools should also monitor and manage SLA metrics in real time, rather than providing mere historical views of past performance. The historical information should be saved in the database. This section contains very brief overview of leading LSA management tools from the functional point of view.

2.1 Total Service Management (Cisco)

Cisco Total Service Management (TSM) [4] delivers on the demands of end-to-end service-level management with precise demarcation, per-tunnel measurement, and both business-level and detailed technical management reporting.

Cisco TSM mainly focuses on the networking layers and makes net-centric SLAM metric information, such as round-trip latency, jitter, and response time, available to vendor applications dealing with upper and lower layers. Thus, TSM makes it possible to create an end-to-end solution by integrating cooperative applications from best-of-breed vendors.

The ability to define a service level in a consistent and standard way is a key to successful deployment of service-level management tools. Cisco intends to provide leadership in creating open standards around the definition and measurement of SLAs. Cisco SLAM implements XML-based data exchange for configuration and control of network-layer service-metric data collection. Cooperating vendor applications are able to request SLAM to collect service metrics such as round-trip latency, jitter, and response time using a standard definition. TSM utilizes Cisco IOS embedded technology called Service Assurance Agent (SA Agent) to perform network-centric testing of services and provide service metric information to cooperating applications through a standards-based XML interface.

The Cisco TSM solution is being designed to provide both high-level management reports and detailed reports for network administrators. This is a significant improvement on traditional reporting features, which prepared data only for a focused set of individuals within an IT department.

2.2 Inteligo Advisor (ViewGate Networks)

Inteligo Advisor [5] is a service provider tool for monitoring service and verifying service-level agreements (SLAs). Utilizing artificial-intelligence techniques, Inteligo Advisor predicts usage needs for virtual private networks (VPNs) and recommends service adjustments needed to insure that the service definitions are application appropriate. ViewGate Network tool also provides real-time SLA compliance verification and network-usage anomalies. This combination of functionality enables service providers to meet SLA, improves their response to changes in customer needs, and build greater customer loyalty.

The Viewgate Inteligo Distributed Information Repository maintains Inteligo Advisor VPN-usage data. It performs statistical forecasting based on historical performance information and compares those findings with the customer's SLA.

If Inteligo Advisor system detects a network-performance metric (such as jitter) trending toward a violation of the SLA, it alerts the customer and service provider accordingly. Utilizing an artificial intelligence engine, Inteligo Advisor then recommends actions to address the predicted network situation. Inteligo Advisor's trend analysis enables continual rightsizing of the service. Inteligo Advisor also measures service performance against the SLA in real-time. It can alert service provisioning, traffic-optimization, billing, and other operations supported by the system, as well as the customer oriented SLA violations and situations where violations are imminent. The alert can trigger corrective actions, resulting in better service.

2.3 FireHunter (Agilent)

Agilent's QoS Manager (Firehunter) [6] allows the client to monitor service performance and availability in real-time for wireless, wireline, switched and packet data networks, for a complete picture of service quality from your customers' perspective and automated management of service level agreements (SLAs). QoS

Manager manages a variety of wireless and wireline technologies, including GSM, GPRS, UMTS, optical and IP.

QoS Manager's automated SLA management and reporting provides the tools for capturing SLA requirements from customer contracts and configuring the system. The client can automatically create and schedule graphical SLA report cards that demonstrate compliance with customer contracts and document QoS delivered across all services, distributing them in real time on customers' secure Web sites.

The client can use Firehunter to measure, monitor and report on Web, e-mail, news, DNS, RADIUS, file transfer protocol (FTP), LDAP, Web hosting, mobile data and network performance management services. The QoS Manager Service Model graphically portrays the end-to-end service delivery chain and reflects the status of each delivery element. The client can also customize the application to add new service models and create specialized views and reports.

2.4 Oblicore Guarantee

Oblicore Guarantee [7] is a complete software package that allows proactive, continuous, top-down management of Service Level Agreements (SLAs) and other important business obligations. This tool is a field-proven software suite that allows the client organization to meet contractual commitments, service level agreements, operational goals, more often and more the SLA business logic and compares performance objectives to real-time data aggregated across the enterprise.

Guarantee enables real-time, end-to-end service level management, supporting SLAs throughout their entire lifecycle: creating a service catalog, evaluating terms for prospective commitments, managing versions, reporting on performance, and calculating financial impact.

2.5 WSLA (IBM)

Web Service Level Agreement [8] language (WSLA language) defines assertions of a service provider to perform a service according to agreed guarantees for IT-level and business process-level service parameters such as response time and throughput, and measures to be taken in case of deviation and failure to meet the asserted service guarantees. The assertions of the service provider are based on a detailed definition of the service parameters including how basic metrics are to be measured in systems and how they are aggregated into composite metrics.

The WSLA language is based on XML; it is defined as an XML schema. WSLA can be used by both (provider and customer) services to configure their respective systems to provide and supervise their service. This process is called deployment. Each organization uses its own independent deployment function that interprets the WSLA and takes appropriate action. The deployment step includes creation and parameterization of the relevant service implementing systems and WSLA supervising services. After deployment, supervising services can enact the WSLA.

A very important aspect of WSLA is its capability to deal with specifics of particular domains and technologies. The language is extensible to include specific types of operation descriptions, measurement directive types for specific systems, special functions to compose aggregate metrics and predicates to evaluate specific metrics. The WSLA extension mechanism makes use of the ability to create derived types using XML schema. By design, the core of the WSLA language is very compact. To be of immediate use, the WSLA language includes a set of standard extensions. It allows WSLA authors to define complete agreements that relate to Web services and include guarantees for response time, throughput and other common metrics.

3. SLA metrics definitions

The purpose of service level agreements is to formally define the parameters of service the provider guarantees to deliver. After agreeing upon an SLA, the specified parameters are monitored in order to detect agreement breaches. One of main issues of SLA creation is to define methods of measuring certain service parameters that must be agreed upon by both sides of the contract. When a SLA breach is detected, an appropriate remedy procedure (also defined in the contract) is applied. Moreover, the client is eligible to get some compensation from the provider. Identifying the violations and calculating penalties might prove quite challenging.

In general, a SLA contract can be presented as a logical product of predicates built upon results of measuring certain service parameters. To make it more understandable, the SLA can be expressed as a hierarchy of sub-contracts that would eventually be expressed as rules defined on the measurement-based predicates.

The SLA Management System (SLAM) relies on a monitoring database, which feeds it with the results of parameter measurements. The tasks of the SLAM systems are to:

- interpret data provided by a monitoring system in context of agreements signed by a particular provider. This functionality includes also notifying the provider about unexpected events that could lead to an SLA breach before it occurs.

- provide regular reports for the customer regarding the SLA parameters [1].
- provide data for external systems for calculating compensations for SLA breaches.

The compensations are typically expressed in terms of rebates or payment-free periods of service provision the client gets in case of contract breach. Although the calculation details are covered by the contract, and the SLAM system itself is not intended to calculate the penalties, a few factors regarding the calculations were taken into consideration in the design phase in order to point out the data that should be delivered to the calculation subsystem. On one hand, to allow maximum flexibility of penalty calculations, fine-grained output seems to be required. On the other, the SLAM is not expected to duplicate the monitoring system functionality. An approach that was selected was to define an interface between SLAM and monitoring system(s) that would be used, if needed, to deliver the fine-grained data.

4. SLA monitoring and management system requirements

A key success factor for developing efficient SLAs is to ensure that focus is kept on translating business objectives to SLAs, where tangible service metrics can be measured, reported on, and validated. A key reason for failing to manage service levels is attributed to long, complex, and unrealistic agreements. As SLAs become more popular, the requirements for SLAs become more straightforward. The enterprise customer wants:

- The ability to confirm that the SPs SLAs are being met for both connectivity and hosted applications,
- The ability to demarcate for SLA problems with fine granularity,
- The ability to impose financial penalties for violated SLAs,
- Web-accessible business-level reports and detailed technical reports that can be enhanced and extended easily and incrementally,
- Validation that key business objectives are being met when deploying new world technologies.

SLA monitoring and management (SLAM) requirements were investigated in details basing on questionnaires and interviews with eight national and international telecommunication operators. This section summarizes this work and gives very fresh view of current, real technical and non technical requirement that must be meet by SLAM platform.

4.1 Technical Requirements

The challenges of providing end-to-end service-level management can be daunting. In fact, no single vendor today offers a comprehensive end-to-end service-level management solution. This section defines SLA management activity and summarizes the obstacles that must be overcome to monitor and manage service levels effectively.

Service-Level Agreement Management Defined

Delivering on service-level agreements requires service-level agreement management (SLAM) where a given service has several service-components. While different vendors have different definitions and capabilities for managing service-levels, only one definition and capability is meaningful to customers: end-to-end management of all aspects that relate to the connectivity, performance, and availability of the service or application.

Therefore, vendors focus has been on developing an end-to-end SLAM solution. The solution enables integration through open-standard interfaces and visibility of customers to all the various service-components that make up a service.

The Difficulties of Providing End-to-End SLAM

Delivering end-to-end SLAM is extremely challenging. The solution must leverage component management products from multiple vendors; it must work with clients and equipment the customer does not own or have control of; it must adapt to new SLAM metrics as new technologies are deployed; it must scale by orders of magnitude; and it must collect the right network and application SLAM metrics at the right times – Fig.2.

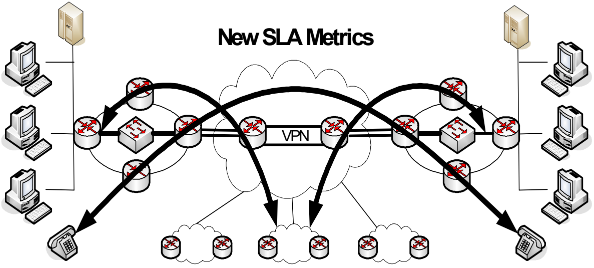


Figure 2. Effective SLAM

Collecting Data through the Stacks

SLA metrics vary widely, and therefore the technologies for measuring those metrics vary as well.

In managing service levels end to end, it is necessary to measure and collect data at every level of the network stack—from the Layer 5, 6, and 7 client-to-server layer to the Layer 3 and 4 network and network services layer to the Layer 1 and 2 WAN layers. The problem is that today, no product or technology does this. There are high-quality products from multiple vendors covering the client-server layers and the WAN layers, but there is little available in the Layer 3/Layer 4 space and virtually nothing that ties together the entire stack.

Importance of Device Instrumentation

Another complication of end-to-end SLAM is that client and application information must be acquired and related to the metrics acquired from the network infrastructure. The solution needs to minimize the increased complexity in the network infrastructure when acquiring the network SLAM metrics. This is best achieved when the network devices are instrumented appropriately.

Need for "Drill-Down" Capabilities

While it is important for an end-to-end SLAM solution to provide summary information to management, it is also critical that administrators be able to drill down and get details about specific problem areas. The ability to demarcate what is causing the SLA not to be met—whether it is a problem at the client end, at the server end, or in the network—is crucial.

4.2 Functional Requirements

Functional requirements for SLAM platform were worked out basing on ISPs questionnaires, literature study and analysis of actual competitive solutions, have a few dozen of pages. The most important ones are presented in this chapter. They are divided into two groups, SLA contract specification and platform functionality.

SLA contract requirements

From the business perspective SLA contract has to include not only technical but mainly organizational areas of service provisioning, all of them have to be mapped in SLA contract effectively deployed and managed by the platform. This area covers:

- Definition of service provided, parties involved, and effective dates of agreement.
- Specifications of hours and days that service will be offered, including testing, maintenance and upgrades.
- Specification of the number and locations of users and/or hardware/software for which the service will be offered.
- Explanation of problem-reporting procedures, including conditions of escalation to next higher level of support.
- Explanation of procedure for both parties to request minor changes to the way the service is provided (e.g., software upgrade to a server), which may include expected times for completing routine change requests.
- Specification of charges and billing and payment procedures associated with the service, which may be flat rate or tied to different levels of service.
- Specification of target levels of service quality and how associated metrics are calculated and how frequently they are reported, to include availability and response time.
- Description of procedures for resolving service-related disagreements.
- Process for amending SLAs to support expected growth, change, and demand for the service or system.
- Specifications of how to address off-hours support, unscheduled outages, or emergency back- up systems, how emergencies will be handled and an outline of escalation process.

Platform functionality requirements

Form SLAM operator perspective, the platform should:

- provide mechanisms to manage services in the following areas: SLA management, services quality monitoring, services repository,
- support SLA development through the SLA templates management and SLA offers management,
- provide inventory of services. Services in inventory are represented in hierarchical structure where one service may depend on other services.
- provide a set of standard predefined QoS objectives for standard QoS parameters, including: Max Time To Restore Service and Min Availability,
- support service independent types of SLA QoS parameters: Availability, Current Outage Time, Total Outage Time, Number of outages and Applied bonus(discount),
- support SLA QoS statistical parameters: MTBF (Mean time between failure), MTTR (Mean time to repair) and MTBO (Mean time between outages), calculated at the end of contract date or period,

- allow to monitor quality of managed services from the business perspective. The SLA calculates time left to restore failed service, current and future losses,
- Support SLA alarms which contain information such as: SLA contract, affected customer, violated QoS parameters (current value and violated constraints), applied discount (loses),
- allow to prioritize network and service faults according to possible financial loses, customer types, affected customers,
- generate at the end of assessment period summaries which contains all SLA violation information. This summary may be exported to external BSS system for bonuses calculation. System provides additional SLA reports for customer. The reports are available through web interface.

The above functional requirements could be changed and adapted depending on the stage of the solution development and the conclusions drawn in each stage the system deployment.

5. SLA framework architecture

Effective implementation of SLA management framework requires specific infrastructure services to be accessible. This includes data sources together with specific adapters to collect and process the necessary data, graphical user interface, and the communication bus. The framework requires also to be deployed over real network architecture. This process needs access to specific inventory database to retrieve information about physical description, parameters and localization of each item (network device, interface, host) and how it is interconnected.

Because the SLAM system has to process different types of information – the modular approach has been proposed as the most suitable for such implementation. Two types of information have been identified to be the most important. The first type it is the Fault Management information, either retrieved from existing Fault Management systems, or gathered directly from network elements. This information should be processed in the Service Monitoring module in the SLAM system. The second type of information it is performance related KPIs (key Performance Indicators), and heavily time dependant. This information should be processed by the Performance Management System in which the Performance Management engine is the most important.

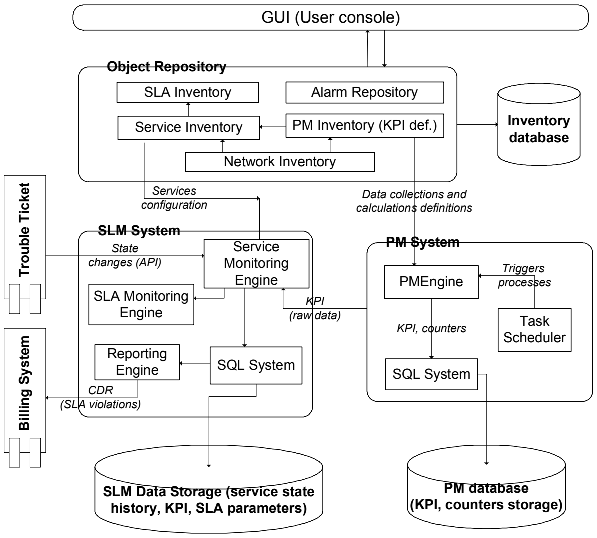


Figure 3. SLAM system architecture

Figure 3 describes the overall architecture schema of the SLAM framework. Modular approach allows, by the use of mediation agents, to collect data from different data sources such as: Trouble Ticketing system, Performance Management database or directly from the network elements. Such approach enables to use the SLAM framework, either as an umbrella system for other vendor specific, or client specific systems, or as a stand alone solution. The details of an example implementation will be provided later in this paper in the Case Study section.

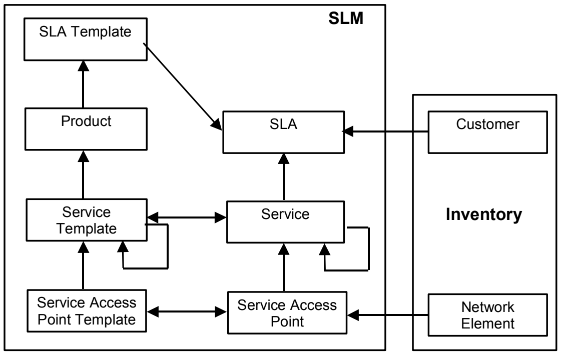


Figure 4. Logical architecture of the SLAM framework

The logical architecture of the system presented in Fig.4 is composed of different levels containing:

- Service Access Points (SAP) – related to network elements (physical devices or components of these devices),
- Services – composed of different service nodes or/and other services,

- Products – understood as groups of services,
- SLAs – linked directly to products.

Each SAP, Service, SLA or Product is defined as a template object first and then instantiated as a working instance (see Fig.5). SAP represents the relation between service and network infrastructure. SAP Template at the same time is a template object for SAP and in addition to it's attributes, SAP Template object consists of set of rules for easy assigning appropriate Network Element objects to SAP objects related to a given SAP Template.

Service Object is a single element of service-tree which is the representation of the real-life service structure and consists of the following attributes:

- current state of the element,
- event propagation formula for given element,
- KPI propagation formula for given element.

Each service object may have one of the two processing modes: Automatic - the state of the Service object is automatically calculated by the engine upon event propagation formula, Manual - the state of the Service is set by operator. Similarly to SAP Template - Service Template object is used to define service hierarchy and processing rules only. This object is not processed by SLAM Engine during Service and SLA monitoring process.

Product object is used to group service tree-like structures. The template of the SLA contract can only be created in context of Product.

Finally the SLA Object representing SLA contract consists of the set of common SLA parameters (Availability, Max Time to Restore, Time to Violate) and SLA KPIs, and it is instantiated according SLA Template. Only one Product Object can be attached to the SLA Template.

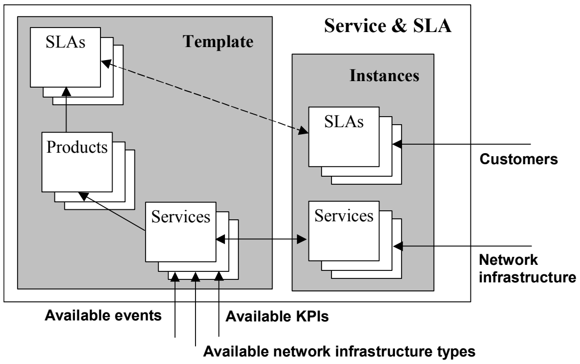


Figure 5. Instances and Templates in the SLAM framework

The "near real time" computation approach has been identified as the most suitable for updating parameters of the SLA objects. The required granularity of time is the parameter of the system. The performance issues and data delivery schedules determine the frequency of calculations of states for each of the objects.

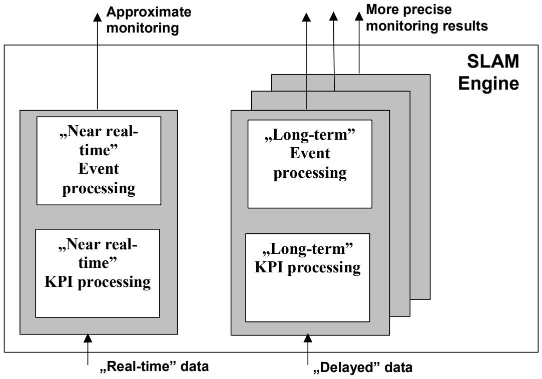

Approximate monitoring
More precise monitoring results
SLAM Engine
„Near real-time" Event processing
„Near real-time" KPI processing
„Long-term" Event processing
„Long-term" KPI processing
„Real-time" data
„Delayed" data

Figure 6. "Near real time" and "Long-term" information processing

The proposed concept is presented in Fig.6. The "near real time" mode processes data incoming from the network infrastructure (events, KPIs) in almost real-time. All parameters of services and SLAs are calculated continuously with configurable time period (depending on number of monitored services and SLAs and hardware specification the „real time" calculation period can vary between 1 minute to 60 minutes). Because not all network devices provide required data in short periods of time, it is possible that results of „real time" processing can be inaccurate (for example KPIs can be retrieved form some devices once an hour or even rarely).

In order to process historical or outdated data coming from network infrastructure (events, KPIs) the „Long-term" processing mode has been implemented. It recalculates all parameters of services and SLAs taking care that all the information which was not available during „near real time" processing will be processed tin the "long-term" cycle.

„Long-term" processing helps to improve the results of evaluation of parameters of services and SLAs. It is possible to define more than one „long-term" processes running with different delays and in different schedules, but at least one „long-term" processing must be running to generate reliable data for SLAs after accounting period.

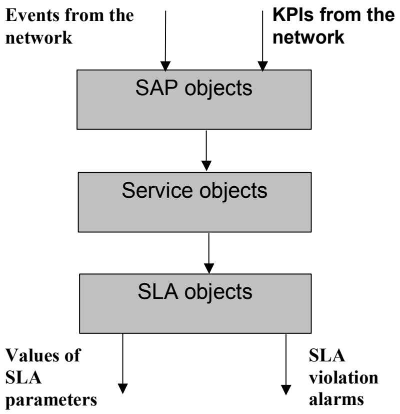


Figure 7. Events and KPIs processing

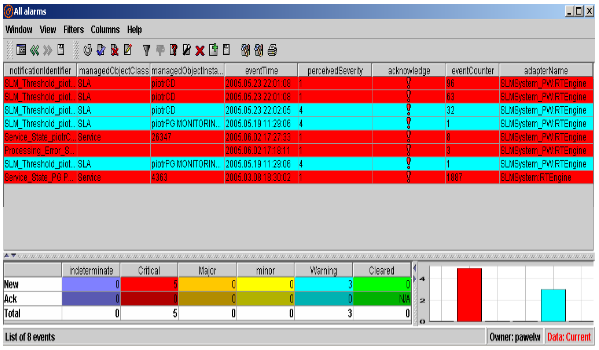
Events from the (see Fig.7) network are collected by the Fault Management system and processed through filters classifying them into three categories (CRITICAL, WARNING, CLEAR). They are used to define the state of the SAP which is then propagated upwards the service tree structure. For each service object the service propagation rules decide which events should be considered to put the service in a specific state. For SLA object the parameters are calculated and after checking for thresholds the values of SLA KPIs, SLA violation alarms are generated if necessary. The SLA KPIs are specific for SLA and are somewhat different form the performance indicators for services and SAPs.

The KPIs for Service Access Points are aggregated on each level of the structure. It is possible to define separate aggregation formulas for each type of SAP or Service. After aggregation the KPIs are used to define the state of the Service and then propagated upwards to define the state of the SLA.

Three main modes have been implemented to manage the SLA contracts:

- „Real time" reporting with Alarm List,
- „On demand" reporting with Reporting Module,
- Automatic, scheduled reporting with separate Reporting Module.

The Alarm List, shown in Fig.8 enables (with use of specific filters) to show only this information which is at the moment the most relevant and important for the operator. The operator can monitor the states and KPIs of the SLAs, and in case of an emergency can "drill-down" through the whole tree like structure of the services and SAPs exactly to the network element which causes problems. This speeds up the reaction times, and helps to cut down on penalty costs resulting from SLA violations.



notificationIdentifier	managedObjectClass	managedObjectInsta...	eventTime	perceivedSeverity	acknowledge	eventCounter	adapterName
SLM_Threshold_piot...	SLA	piotrCD	2005.05.23 22:01:08	1	!	96	SLMSystem_PW:RTEngine
SLM_Threshold_piot...	SLA	piotrCD	2005.05.23 22:01:08	1	!	63	SLMSystem_PW:RTEngine
SLM_Threshold_piot...	SLA	piotrCD	2005.05.23 22:02:05	4	!	32	SLMSystem_PW:RTEngine
SLM_Threshold_piot...	SLA	piotrPG MONITORIN...	2005.05.19 11:29:06	4	!	1	SLMSystem_PW:RTEngine
Service_State_piotrC...	Service	26347	2005.06.02 17:27:33	1	!	8	SLMSystem_PW:RTEngine
Processing_Error_S...			2005.06.02 17:18:11	1	!	3	SLMSystem_PW:RTEngine
SLM_Threshold_piot...	SLA	piotrPG MONITORIN...	2005.05.19 11:29:06	4	!	1	SLMSystem_PW:RTEngine
Service_State_PG P...	Service	4363	2005.03.08 18:30:02	1	!	1887	SLMSystem:RTEngine

	indeterminate	Critical	Major	minor	Warning	Cleared
New	0	5	0	0	3	0
Ack	0	0	0	0	0	N/A
Total	0	5	0	0	3	0

List of 8 events | Owner: pawelw | Data: Current

Figure 8. Alarm list service monitoring

The "on demand" reporting helps the operator to see all the aggregated KPIs of the selected service and SLA. It might also be useful before actually signing and implementing the SLA, in order to prepare feasibility study if the expected parameters can possibly be met.

Automatic reporting can be scheduled after the SLA has been signed in order to prepare detailed billing information and SLA report for the customer. Specific rules can be defined in order to calculate the penalties resulting from SLA violations.

SLA management framework can also be integrated with CRM system, and certain access rights to some information can be given to sales agents or account managers to help them manage specific SLAs of their clients retrieve information about their states.

6. SLA application case study

The prototype of SLAM framework system has been installed as a pilot implementation for a large national fixed telephony and internet service provider. The solution is now used to monitor state of the 4 VPN(s) (Virtual Private Networks) and to calculate the SLA parameters of the SLA contracts. Each node of the particular VPN is connected to the different interface of the core network routers and is monitored independently. The monitoring of the VPN node is performed by Service Assurance Agent which is embedded part of the OS of the Cisco routers. The agent periodically sends the configured amount of data packets to the Customer Premises Equipment (CPE) where Agent Responder is installed. Therefore the agents provide following metrics of the link:

- Availability (0 – link is available, 1 – link unavailable),
- Average Round Trip Time [ms],
- Round Trip Time Jitter [ms],

- Packet Loss [%],
- Incoming traffic (kb/s),
- Outgoing traffic (kb/s).

The measured values are accessible by means of SNMP protocol and gathered from the routers every five minutes and stored in flat files by an external application. Periodically (every hour) the files with performance data are uploaded into two databases: production one and developer one.

The requirements were defined such, that only the performance indicators were monitored, and no fault management data were taken into consideration. The performance data are stored every hour into dedicated database (Sybase ASE-15). Several thresholds for availability and performance parameters have been defined and penalty calculation formulas have been implemented according to the requirements of the ISP.

The mediation used to fetch the data from the dedicated mediation database running Sybase is composed of the two server processes:

- process querying the relational database,
- "adapter" which is a translator between standard SLAM system data request API calls to the process which queries the SQL database.

Once data are fetched the process creates the Performance Management (PM) Module data response and sends it back to the PM Module where the data is processed (KPI are calculated and stored into internal database and could be utilized by SLAM module).

The inventory data model has been configured in order to store specific objects required to create performance data collection processes, two new classes have been created in order to map the inventory model on it's representation in the Mediation Database. The first class (SAACollector) represents the connection between the interface on the ISP router and the customer site. For each VPN node one instance of this object has been created. The second class (VPNSite) represents a customer site as a part as the particular VPN.

Because the SLAM framework philosophy requires first to create templates and then instantiate them - six templates have been defined: one SAP template (defines template for the VPN node link and specifies which KPIs are propagated to the service level), one service template (using the previously defined SAP template with the specification which KPIs are available at the SLA level), and one product template (containing the previously defined service template) and three SLA templates.

Then the specific instances of the objects and SLA contracts have been created out of the previously defined templates. The correlation system has been used in this implementation to perform custom action when the PM Threshold alarms are generated. This kind of alarms notifies the user that bandwidth usage is above given threshold and the penalties should not be calculated for the KPI(s).

The reporting functionality allows the users to:

- display the current KPI values in the graphical way (chart),
- monitor the status of the services and SLA contract in so called "near real-time" by means of the Alarm list and dedicated views available in SLAM module,
- calculate the penalties for the SLA violations by means of predefined report templates.

Therefore the SLAM system has been implemented successfully in a real environment. All the predefined requirements of the client have been met, and presently the system is being evaluated from business perspective.

7. Conclusions

The proposed framework for SLA monitoring and management represents rather general solution that may be customized according the offered services provisioning, SLA definition requirements and underlying monitoring system infrastructure.

The constructed mechanisms of data flow and transformation between the monitoring layer and the SLA layer are rather general. They benefit from clear object oriented approach and representation of the services as a tree structure.

A very sensitive point of the proposed solution is the SLA evaluation procedure. Splitting this procedure into "near real time" and "long-term" phases seems to be the only solution which could satisfy rather general needs.

The proposed SLAM system follows an open loop control model without feedback between observed SLA values and activities that should be taken on network infrastructure layer to achieve SLA objectives. Closing this loop it another challenging task should be resolved.

Acknowledgement

This research was partially supported by Polish Ministry of Education and Science; grant no. E! 3152 SLAM.

8. References

[1] Carr Jim, *Service Level Agreements,* CMP, Inc., 2001.

[2] *Service-Level Management: Defining and Monitoring Service Levels in the Enterprise*, White Paper, Cisco Systems, Inc., 2001.

[3] John Lee, Ron Ben-Natan, *Integrating Service Level Agreements: Optimizing Your OSS for SLA Delivery*, Wiley, 2002.

[4] Rick Sturm, Wayne Morris, *Foundations of Service Level Management*, Sams, 2000.

[5] *Viewgate's Advisor Product Enables Service Providers to Deliver on VPN SLAs and Build Customer Loyalty* , Viewgate Networks, ISP Business, July 2001.

[6] Aglient, *Agilent's QoS Manager – Firehunter*, www.agilent.com.

[7] Oblicore, *Oblicore Guarantee*, www.oblicore.com.

[8] IBM, *WebService Level Agreements – WSLA*, www.research.ibm.com/wsla

[9] Comarch S.A, *InsightNet*, www.comarch.com.